ENERGY EFFICIENT ROUTING STRUCTURES AND WAKEUP SCHEMES FOR WIRELESS SENSOR NETWORKS

by

BYOUNGYONG LEE

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2009

To my wife Young Ok and my parents

ACKNOWLEDGEMENTS

I would first like to acknowledge my supervising professor Dr. Ramez Elmasri, for the priceless advice he offered during my Ph.D course. His encouragement has helped me develop the central ideas for this dissertation. Dr Ramez Elmasri is the best professor I have had in my life. I would also like to express my appreciation to my advisors: Dr. Hao Che, Dr. Leonidas Fegaras, and Dr. Yonghe Lie, for my academic development.

I would like to thank Dr. Kyoungseo Park, Dr. Jung Hun Oh and Dr. Young Bun Kim and Dr. Jeongkyu Lee for all the time they spent advising me. Furthermore, I wish to thank Chulho Ahn, Hyun Lee and Jaesung Choi for the discussions held about my area of research. I would like to extend my appreciation to my Wizwa family.

I should not forget to thank my wife Young Ok Choi, who has supported and suffered with me throughout our life together. As for my parents and brother Byounghun Lee, I would like to also thank them for their support. In addition, I would like to extend my appreciation to my father-in-law and mother-in-law for the support they offered, as I pursued my doctorial degree.

Finally, I would like to thank God for giving me a lot of blessings and love.

May 4, 2009

ABSTRACT

ENERGY EFFICIENT ROUTING STRUCTURES AND WAKEUP SCHEMES FOR WIRELESS SENSOR NETWORKS

Byoungyong Lee, Ph.D.

The University of Texas at Arlington, 2009

Supervising Professor: Ramez Elmasri

Wireless sensor networks, which consist of a large number of sensor nodes and a base station, are used for many applications aimed at collecting information. Each sensor node is equipped with a small amount of battery, limited memory, finite radio range and small CPU. It gathers required information and it sends the information to the base station. The large number of sensors can cover a large area by cooperating with each other to build a multi-hop wireless network. However, the small amount of battery is one of the critical concerns because sensor network life time depends on battery longevity. It is hard to replace or recharge the battery in each sensor node. Generally a sensor node consumes its energy during processing, receiving, transmitting and overhearing of messages. Among those, we focus on reducing the data communication and overhearing energy consumption. In order to accomplish these tasks, we propose novel energy efficient routing structures and wakeup schemes in this dissertation.

First we propose an energy balanced technique for in-network aggregation with multiple tree structure (MULT). We try to reduce concentrating network traffic on a few special nodes. For building the multiple tree structure, we first create node clusters, and then connect the nodes in each cluster. Finally with cluster head nodes, we construct a multiple tree structure. In the

second technique, we propose a sensor network subtree merge algorithm (SNSM), which uses the union of disjoint set forest algorithm to avoid unnecessary energy consumption in ancestor nodes for routing. We reduce the energy consumption for routing in sensor network for spatial range query through the SNSM algorithm. We apply SNSM algorithm to a minimum spanning tree structure. For our third contribution, we make a wakeup scheme to reduce the overhearing energy consumption using different wakeup time scheduling on children nodes. Our wakeup scheme includes two wakeup schedules. One is odd and even wakeup scheduling (OEWS) and another is individual wakeup scheduling (IWS). There is a trade off between reducing overhearing energy consumption and delay time. Therefore we propose double tree structure called DTS to reduce the delay time. Simulation results illustrate that our energy efficient routing structures and wakeup schemes extend the sensor network lifetime and make a small trade-off between energy consumption and delay time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Wireless networks is the fastest growing field of communication technologies. Various new technologies are emerging such as wireless sensor networks, smart homes, personal portable devices, pervasive computing and wireless health care systems. Because of these developments, many technical issues have emerged, which have to be solved to allow full realization of these technologies. One of these issues is battery technology. With the increase in capacity of memory, processor and radio range, the energy consumption of wireless devices have increased. However the development of battery capacity has been much slower than energy consumption of wireless devices. We need to develop wireless devices running on low power. Another issue is the mobility of wireless networks. Continuous data service is very important for wireless networks. When a wireless device goes beyond the radio boundary, it changes the available radio channel to support mobility. Mobility for continuous data communication and quality of service are related. A third issue is the security of wireless networks [12]. Because of the characteristics of wireless network, information is exposed to the open environment.

Wireless sensor networks is a new technology that emerged from the development of wireless networks. It has potential to develop industrial, military, health care, home automation, scurity, and transportation and other applications. Wireless sensor networks have some different characteristics from traditional wireless networks [12], therefore requiring new technologies.

1.1 Wireless Sensor Networks

Wireless networks are expected to play an increasing role in our life to improve convenience, health, and safety. Advances in MEMS (Micro-Electro Mechanical System) have led to the emergence of wireless sensor networks because it is possible to make a low power processor, powerful radio unit and cheap memory. Wireless sensor networks are increasingly applied to various physical worlds for surveillance and other applications such as military, fire monitoring, health care and so on [13, 14, 15]. A wireless sensor network is typically deployed with numerous small sensor nodes equipped with small CPU, low power battery, small memory, short radio transmitter, receiver and various sensors. Because large numbers of sensors are typically deployed, the trend has been to decrease the cost of each sensor node. As a result, a sensor node has smaller size than before. Each sensor node scattered in the sensor field is part of the network.

When receiving a query from a user, the base station sends the query to nodes in the target area for collecting the information through the formed network. Because there is limited wireless network infrastructure, each sensor node plays a role as either a routing or sensing node or both. Each sensor node measures environmental variables such as a pollution, radiation, humidity with sensor device. Fig. 1.1 illustrate the example of wireless sensor network. After a sensor node gathers information through the sensor device, it sends the data to neighbor nodes or directly to the base station along with information about the sensor network topology. A typical sensor node is composed of sensing unit, processing unit, transmission unit and power unit.

Recent sensor network applications typically form a wireless network. The main problem with wired sensor network is cost and delays in deployment [11]. Also it has environmental constraints. For example it is hard to deploy wired sensor nodes to a dangerous area. The wireless sensor network overcomes many restrictions in wired sensor networks. Therefore we can enlarge sensor network applications with wireless networks.

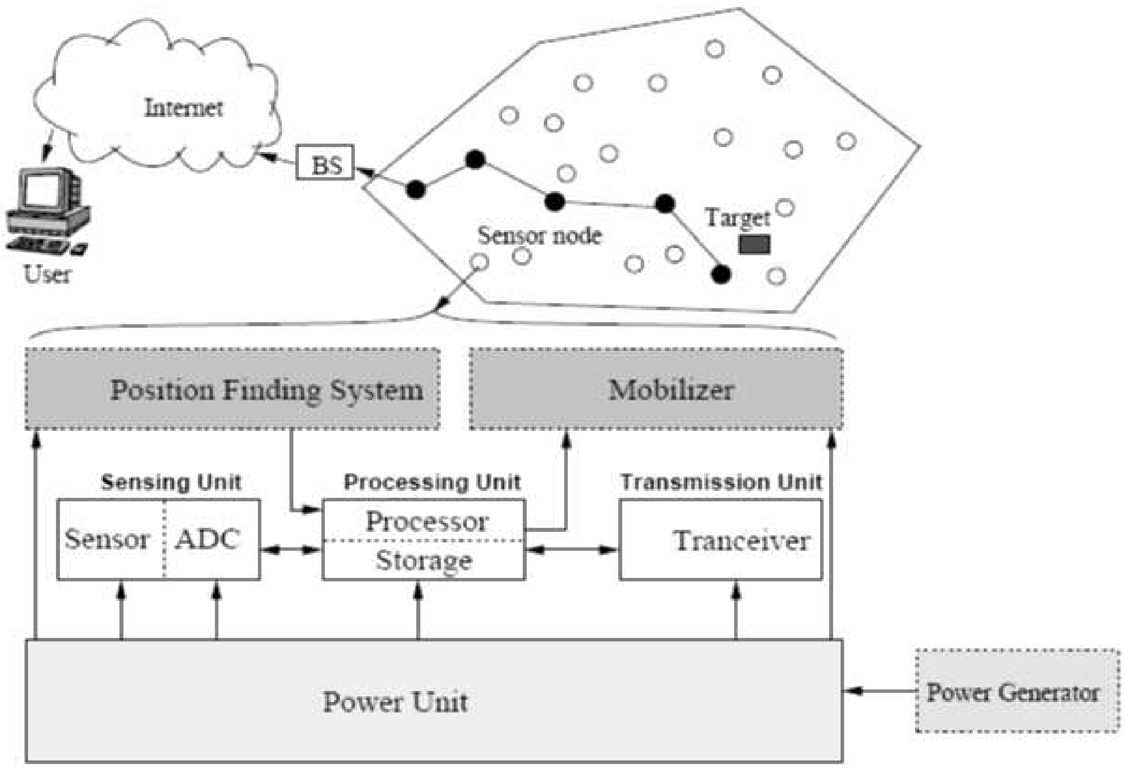


Figure 1.1: Wireless sensor networks [1]

In [2], wireless sensor network applications can be classified into two categories. Fig. 1.2 shows the two categories which are *monitoring* and *tracking*. The applications for *monitoring* are security detection [16, 17], inventory monitoring, patient monitoring [18, 19, 20, 21], environmental monitoring [22, 23, 24] and so on. For example, sensor networks are useful for measuring air and water quality in an environmentally polluted area. Also sensors can provide early warnings for chemical exposure and ozone. Forest fire [25] and Earthquake monitoring systems are useful applications for wireless sensor networks to warn of big disaster in advance. For health care systems, we can monitor vital signs of patients and elderly persons. It is very important to send the notice as soon as possible. There are several applications area for *tracking* such as enemy tracking, animal tracking, human tracking and traffic tracking [26, 27, 28]. For example, a sensor network is suitable on the military battlefields as an intelligence tracking

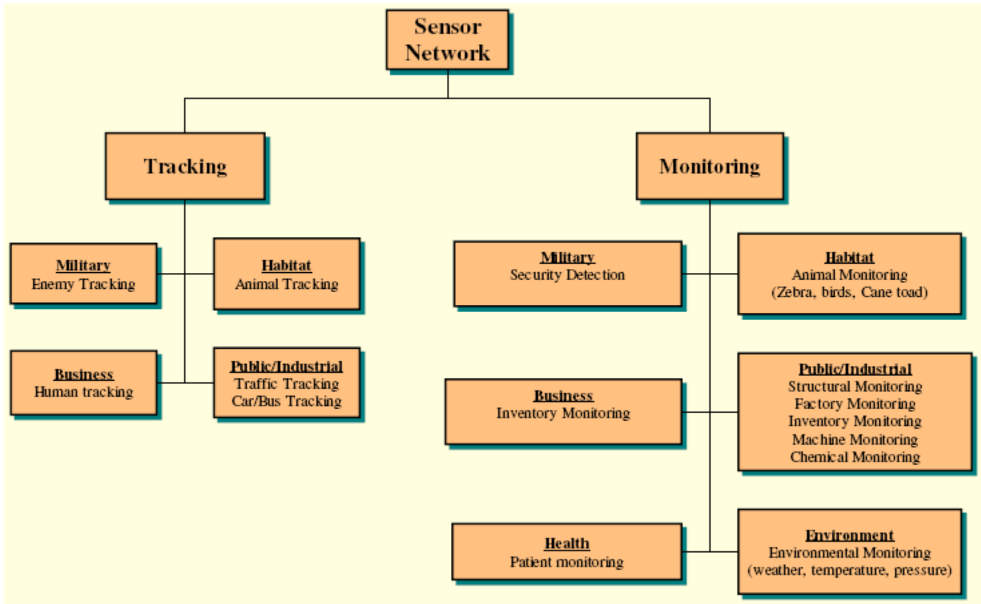


Figure 1.2: Categories of wireless sensor network application [2]

system. Wireless sensors can be rapidly deployed without an established infrastructure in a battlefield [11]. Wildlife conservation is another application for wireless sensor networks. The team of computer engineers of University of California, Berkeley installed wireless sensors that are used to monitor the habitat of the nesting petrels on Great Duck Island [29].

Fig. 1.3 and Fig 1.4 show application sensor nodes of the Crossbow company. Mote Processor has 8K bytes RAM or 4K bytes RAM. Also, it has Atmel ATMega 1281 Processor or Atmel ATMega 128L. As shown in Fig. 1.3, a sensor node has small size, and there are various capacity limitations such as the small amount of battery, limited storage, and short radio range [30, 31].

Even if each sensor node has small capacity, the large number of sensors can cover a large area by cooperating with each other to form a multi-hop wireless network. Nevertheless low battery power is one of the most crucial problems because it is hard to replace or recharge the battery in each sensor [30]. Generally, a sensor node consumes its energy during processing, sensing, receiving, transmitting and idling. Among those, data transmission consumes far more

Mote Processor / Radio Platforms

Features	XM2110	M2110	MPR2400	MPR2600	MPR400
Frequency Range	2.4GHz ISM Band	2.4GHz ISM Band	2.4GHz ISM Band	2.4GHz ISM Band	868-870; 902-928 MHz
Processor	Atmel ATMega 1281	Atmel ATMega 1281	Atmel ATMega 128L	Atmel ATMega 128L	Atmel ATMega 128L
Radio Transceiver	RF230 Atmel	RF230 Atmel	TI CC2420	TI CC2420	TI CC1000
Serial Flash	Atmel AT45DB41B (512 kB)	Atmel AT45DB41B (512 kB)	Atmel AT45DB41B (512 kB)	Atmel AT45DB41B (512 kB)	Atmel AT45DB41B (512 kB)
RAM	8K bytes	8K bytes	4K bytes	4K bytes	4K bytes

Figure 1.3: Crossbow Mote process and radio platforms [3]

energy than processing and sensing [32, 33]. It means that the lifetime of a sensor network depends on the number of data transmissions. Therefore we have challenges to design an energy efficient wireless network routing structure, protocol, database and operating system. For successful performance, we have following issues:

- Limited hardware resources

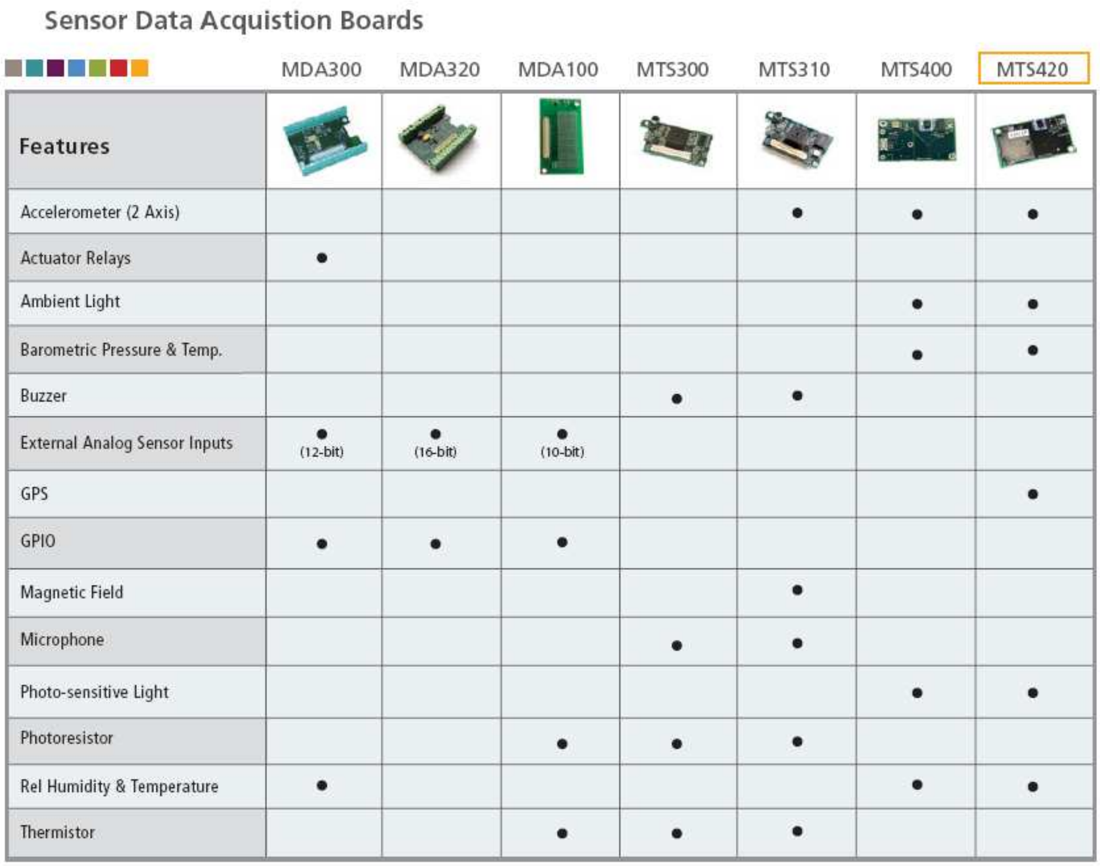
  Wireless sensor networks have limited hardware resources such as small battery power, short radio range, small processor and limited storage. These restrictions make traditional wireless technology not available for wireless sensor networks. For satisfaction of these hardware restrictions, a sensor network should be designed for power conservation.

- Scalability

  The large number of sensors can cover a large area by cooperating with each other to form a multi-hop wireless network. The routing protocols and network topology should be scalable for the high density network structure.

- Fault tolerance

  Because large number of sensors are deployed on the large sensor fields, there is a much higher probability than traditional networks to have to failure [34]. Also sensor nodes directly interact with the environment and will be subject to a variety of physical, chem-



Sensor Data Acquistion Boards

Features	MDA300	MDA320	MDA100	MTS300	MTS310	MTS400	MTS420
Accelerometer (2 Axis)					•	•	•
Actuator Relays	•						
Ambient Light						•	•
Barometric Pressure & Temp.						•	•
Buzzer				•	•		
External Analog Sensor Inputs	• (12-bit)	• (16-bit)	• (10-bit)				
GPS							•
GPIO	•	•	•				
Magnetic Field					•		
Microphone				•	•		
Photo-sensitive Light						•	•
Photoresistor			•	•	•		
Rel Humidity & Temperature	•					•	•
Thermistor			•	•	•		

Figure 1.4: Crossbow sensor data acquisition boards [3]

ical, and biological forces [35]. Therefore sensor networks have lower reliability than traditional networks.

- Security

  Sensor networks are used in several applications that handle sensitive information such as military and health care system [36, 37]. The environments of sensor networks are usually vulnerable to information attack. However it is hard to use traditional security mechanisms because sensor networks have limited hardware resource computational capacity. For tracking confidential information, we should design the sensor networks with security factored in.

- Synchronization

Sensor nodes exchange their sensing data time stamped by each sensor's local clock [38]. Therefore we may need protocols that provide time synchronization [38]. There are many clock synchronization protocols for traditional networks. However they are inappropriate for wireless sensor networks because it has different environment and different constraints..

- Restriction of software development

  Classical database management systems and operating systems were designed for a centralized large system [39, 40]. However wireless sensor networks have limited storage, low power processor, and small battery power. These differences give us new challenges.

1.2 Motivation of This Dissertation

Wireless sensor networks have different characteristic from other wireless network structures. Other wireless network systems are typically established in places where we can supply enough resources and power. However, wireless sensors are typically deployed in places that are hard to access such as military battlefields, environmental pollution areas, and disaster areas. It is very hard to supply enough power and resources to the sensor nodes. Wireless sensor network lifetime depends on batter power. If sensor nodes are out of battery, it reduces the wireless sensor network lifetime. Therefore reducing the energy consumption is a crucial issue in many wireless sensor networks.

Fig. 1.5 illustrates that energy consumption for data communication and idle is much larger when compared to the CPU and sensing and sleep energy consumption [4, 32]. In order to reduce the energy consumption of wireless sensor networks, we should have low data traffic and idle time.

Another aspect of energy consumption is energy load balancing. Usually wireless sensor networks may have tree structures for communication with the base station. In tree

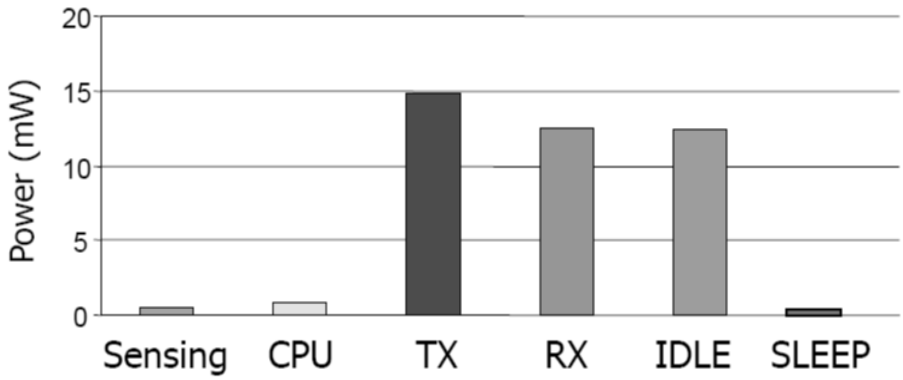


Figure 1.5: Comparison of sensor energy consumption [4]

structures, in-network aggregation is very useful to reduce the energy consumption of data transmissions. After parent nodes of a sensor field combine the results of children nodes, they send it to grand parent nodes. Otherwise parent nodes may send the result data of children nodes to grand parent node as many as the number of children nodes. However, even though in-network aggregation can decrease the number of transmissions, it has some problems. If some nodes are on the most frequently used paths in the network, they will die sooner than others [31]. Therefore even if we try to reduce the data transmission, some nodes concentrated by data traffic will consume more energy than other nodes.

Generally a sensor node consumes its energy during processing, receiving, transmitting and overhearing. Among those, overhearing energy consumption is not necessary for working of wireless sensor networks. A characteristic of wireless networks is that some nodes that are not a destination have to receive unnecessary messages because they are within the radio range. This is called overhearing. As we reduce the unnecessary energy consumption, wireless sensor network lifetime is extended.

1.3 Contributions of This Disseration

1.3.1 Energy Load Balance in Wireless Sensor Netowrks

The first contribution of this disseration is that we have tried to achieve energy load balancing in in-network wireless sensor networks. Many researchers proposed energy efficient techniques for reducing the data communication traffic. However, we have tried to address the other aspects for saving energy consumption in sensor networks. In this dissertation, we proposed energy load balancing with multiple tree structures in wireless sensor networks. The following schemes are our contributions.

- Hybrid Cluster (HYC)

  Hybrid cluster combines the centralized and decentralized methods for clustering. With decentralized methods, each sensor node finds the closest node and with centralized methods each node recognizes which nodes are in the same cluster.

- Multiple Tree Structure (MULT)

  To prevent concentrated traffic on any sensor node, we proposed Multiple tree structure to balance the data traffic.

1.3.2 Energy Efficient Routing Structure

Many The second contribution of this dissertation is that we propose the sensor network subtree merge (SNSM) algorithm for energy efficient data gathering. SNSM algorithm removes the redundant routing paths in the tree structure. SNSM algorithm has better performance with a large number of sensor nodes than with a small number of sensor nodes.

1.3.3 Energy Efficient Wakeup Schemes

The Third contribution of this dissertation is that we have tried to reduce the overhearing energy consumption with sensor nodes wakeup schemes. For reducing overhearing energy consumption, we propose a new wakeup scheme using different wakeup times between neighbor nodes. Usually specific time duration, all neighbor nodes wake up for receiving data from each other. In our case, if neighbor nodes have different wakeup time scheduling, all neighbor nodes do not need to receive the unnecessary data because a sensor node can not received wireless signal during the sleep mode. The following schemes are our contributions for reducing the overhearing energy consumption.

- Odd and Even Wakeup Scheme (OEWS)

  Odd and Even Wakeup Scheme adjusts different wakeup times for children nodes. Each child node wakeup alternately based on its unique ID. But there is a trade-off between energy efficiency and data delay time (latency). For decreasing the latency, we also propose the Double Tree Structure (DTS).

- Individual Wakeup Scheme (IWS)

  In Individual Wakeup Scheme, each child node has a different wakeup time schedule. Therefore at some specific time, only one child node wakes up and the other children nodes are in sleep mode.

- Double Tree Structure (DTS)

  Double Tree Structure consists of short rings topology called SRT and long rings topolgy called LRT. Ring topology makes a tree structure based on the radio range. For reducing the latency, we propose a multi routing path with SRT and LRT.

1.4 Thesis Organization

The rest of this dissertation is organized as follows. In chapter 2, we provide a discussion of related work regarding to the in-network aggregation, sensor network routing structure and S-MAC protocol. In chapter3, we show network and wakeup model. In chapter 4, we describe multi tree sensor network routing structures. We present the three steps to make energy balanced multi tree structures. First step is creating the clustering, second step is connecting the clustering, and the last step is making the multi tree structure. Also we show the simulation results comparing energy consumption and sensor network longevity. In chapter 5, we describe the strategy for energy efficient data gathering. It uses the disjoint set forests algorithm in in-network tree structure. We simulate several aspects related to energy consumption for spatial queries, comparing sensor network density and ratio of spatial query area. In chapter 6, we describe the energy saving wakeup schemes which are odd and even wakeup scheduling and individual wakeup scheduling. Also, we create the double tree structure with short rings topology and long rings topology. We simulate the efficiency of overhearing energy consumption and latency. Finally, we give our conclusion and discuss future work in chapter 7.

CHAPTER 2

RELATED WORKS

2.1 In-Network Aggregation

In wireless sensor networks, limited energy of the sensor nodes is one of the critical problems [41, 42, 43]. If we consider energy consumption, it is known that energy cost for communication of each node is much larger than computational cost. Therefore it is important to reduce the number and range of data transmissions. In [5], an in-network approach for wireless sensor networks for reducing data communication is proposed. The key idea is that each node computes the query in its own place, and produces a local result. For example, if base station receives the MAX aggregation query, each node receives the max values of the sensing data from children nodes, and then applies its sensing data to the max value and sends the result to the parent node. Fig. 2.1 (a) shows the centralized approach, and Fig. 2.1 (b) illustrates the in-network aggregation approach.

In the centralized approach, 16 messages are required for gathering from all sensor nodes. For example, root node needs one time message transmission for sending its data to base station. However, leaf nodes in the bottom need four hop routing path to the base station. So, base station needs a total of 16 message transmission to obtain aggregation information from six sensor nodes. In the in-network aggregation approach, base station requires only 6 message transmissions to acquire aggregation data. Intermediate nodes compute partial aggregation result for all data from its children nodes. Therefore we can reduce the number of data transmissions with in-network aggregation.

In order to send and receive messages, all nodes are synchronized [44]. For example, when a node A has two children,, it is allotted sufficient time called epoch in TAG [44], for

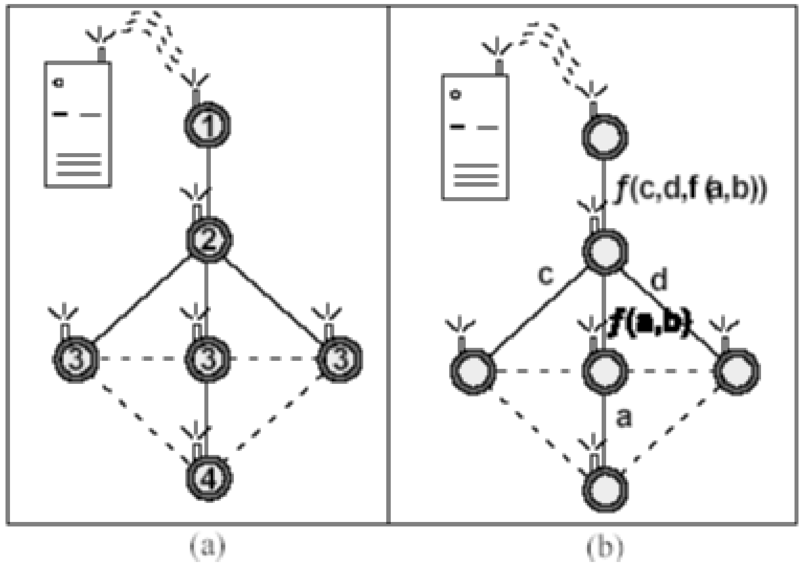


Figure 2.1: (a) Server based versus (b) in-network aggregation [5]

receiving sensing data from its children nodes. If node A is not allotted enough time, node A sends its result to its parent without receiving the sensed data from its children nodes. Therefore query response time is affected by epoch duration. Also epoch duration is dominated by the depth of the routing tree. In [2], the timing strategies are classified as following :

- Periodic simple aggregation

  All sensor nodes have pre-defined period schedule. Within pre-defined duration, intermediate nodes receive the aggregated data from children nodes. If a parent node does not receive the aggregated data from children node within the pre-defined period, it is possible to lose data.

- Periodic per-hop aggregation

  This strategy is similar to periodic simple aggregation. But intermediate nodes send the aggregation data right after it receives data from all its children nodes.

- Periodic per-hop adjusted aggregation

In this strategy, an intermediate node adjust its timeout based on is position in the tree structure.

2.2 Sensor Network Routing Structures

2.2.1 Flat Routing Structure

Flat routing structure [45, 46, 47, 48] is the basic model of sensor network structure. Each node plays the same role in the network structure. If some nodes receive a query from the base station, these sensor nodes try to get the information by activating the sensing device. For covering a specific area, sensor nodes collaborate for gathering the data. After they obtain the information, they send the data to the neighbor nodes that are on the routing path. Therefore each sensor node plays a role as either routing or sensing node. There are many sensor network routing structures based on the flat routing structure such as Directed Diffusion [6], SPIN [49], TAG [44].

In [49], SPIN is proposed to overcome the problems of conventional classic flooding such as implosion, overlap and resource blindness. To overcome these problems, SPIN uses negotiation and resource-adaptation. Each node transmits useful information by negotiation to avoid multiple transmissions of the same data. Also each sensor node has a resource manager. Sensor node checks its resources before sending data. It can measure the cost of energy for processing, sending and receiving. It allows sensor node control its energy consumption.

In [44], Tiny AGregation (TAG) is proposed based on aggregation tree structure. TAG has two main parts. The first part is the dissemination of a query to the sensor nodes and the second part is the data collection from the sensor nodes. For dissemination, it makes tree structure having a base station as root node. The base station sends the query to the sensor node along with tree structure. When the base station collects the data from the sensor node, it uses the in-network aggregation to reduce the data transmissions. Therefore each intermediate

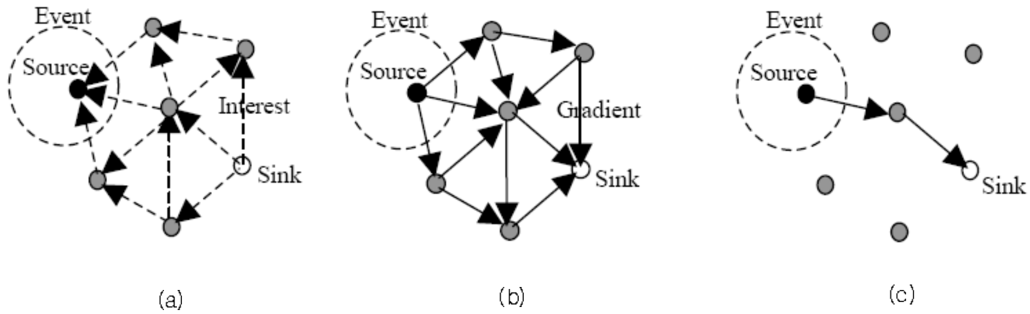


Figure 2.2: Example of directed diffusion (a) Interest propagation (b) Initial gradients setup (c) Data delivery along reinforced[6]

node needs enough time to receive data from its children nodes. After a intermediate node receives the data from its children nodes, it processes the data and transmits it to parent node. TAG has query formed as follows:

SELECT $\{agg(expr), attrs\}$ FROM $\{sensors\}$

WHERE$\{selPreds\}$

GROUP BY $\{attrs\}$

HAVING $\{havingPreds\}$

EPOCH DURATION $\{i\}$

In [6], they propose a data aggregation tree structure called Directed Diffusion. The main goal of Directed Diffusion is to reduce data traffic and transmission delay. Direct Diffusion has three parts. The first part is interest dissemination, the second part is making gradients, and the third part is sending the data on the reinforced path.

Fig. 2.2 illustrates an example of Directed Diffusion. The sink node sends a interest which is a message required by the sink node. Each node having interest broadcasts to neighbor nodes. When each node sends the interest to neighbor nodes, it sets up gradients and then it builds the multipath for the result of the query to send back to the sink node. After it sets up the gradients, only one path is reinforced for routing path from source to sink nodes.

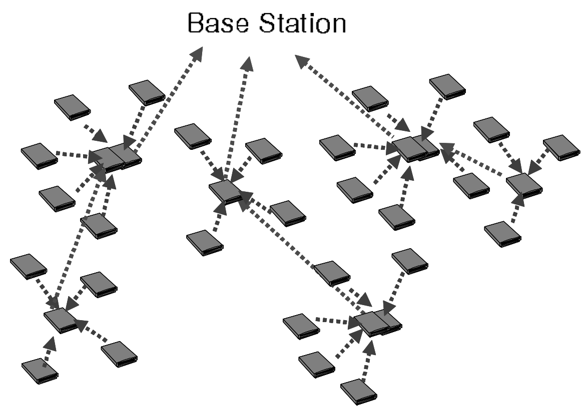


Figure 2.3: An example of clustered routing structure

2.2.2 Clustered Routing Structure

In clustered routing structure [50, 51, 52, 53], some sensor nodes have different roles in the network structure. There are two kinds of sensor nodes in a clustered routing structure. One is a cluster head node and the other is a non-cluster head node. Cluster head nodes collect the data from other sensor nodes within the cluster head area and then send the gathered data to other cluster head nodes for transmission to the base station. Fig. 2.3 illustrates the clustered routing structure.

The cluster head nodes are elected from other nodes within the radio range of the cluster head nodes. The advantage of clustered routing structure is more scalability and energy efficiency to reduce the data transmissions. The disadvantage is added complexity to elect the head nodes. There is an overhead for management of sensor network structure. There are several clustered routing structure such as LEACH [54], PEGASIS [7] and Cougar [55].

In [54], they propose a Low Energy Adaptive Clustering hierarchy (LEACH) based on the clustered routing structure. LEACH has two phases. In the first phase, it makes several cluster and cluster head nodes randomly elected by sensor nodes within the cluster area. The second phase, actual data transfer is made to the base station. When sensor nodes elect the

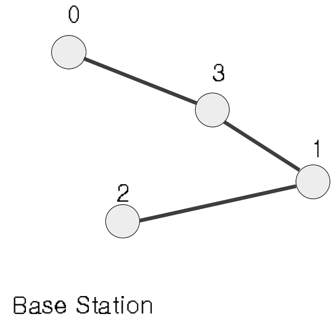


Figure 2.4: An example of chain [7]

cluster head node, sensor node chooses a random number between 0 and 1. If the number is less than a threshold T(n), the node becomes a cluster head node.

$$T(n) = \frac{p}{1 - p(rmod(1/p))} ifn \in G \tag{2.1}$$

In this formula, p is the desired percentage of cluster heads, r is the current round, and G is the set of nodes.

After cluster head node election, each cluster head node sends the message to the non-cluster head nodes within the radio range of a cluster head node. After the non-cluster head nodes receive the message, they decide their cluster head nodes with message signal strength. A cluster head node receives data from non-cluster head nodes in the cluster area. After a cluster head node gathers the data, it sends the gathering data to the base station directly.

In [7], they propose Power-Efficient Gathering in Sensor Information System (PEGASIS). The main structure is to create a chain with the sensor nodes. It has better performance than LEACH. PEGASIS assumes that all nodes have global knowledge of the network and employ the greedy algorithm [7]

For making a chain, it starts from the node which is the farthest away from the base station. In Fig. 2.4, for example, the chain starts from node 0. And this node makes chain with the closest neighbor node which is the second farthest away from the base station. Therefore, in Fig. 2.4, node 3 is connected to node 0. In this way, PEGASIS makes chains. A randomly

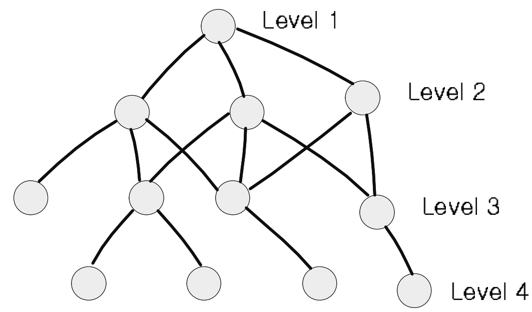


Figure 2.5: Tributaries and deltas [8]

chosen node in the chain sends the aggregated data to the base station. This reduces the number of transmissions and increases the life time more than LEACH

2.2.3 Multi Path Routing Structure

In wireless sensor network, reliability is one of the critical problems. In wired networks, because they are supplied with sufficient energy and resource, data transmission is more reliable than wireless networks. In particular, wireless sensor networks have almost 30% transmission failure rate [56]. In order to increase the reliability in wireless sensor networks, multi path routing structure is useful in unreliable environments. In the single path routing structure, if any parent node does not work, its children nodes could not send the data to the base station. If these children nodes have multiple paths, it overcomes data loss. However multi path routing structure has trade off between transmission reliability and data traffic. In [8], it proposes Tributaries and Deltas approach to make balance between reliability and data traffic. Tributaries and Deltas approach uses the multi path on the unreliable routing path and single path on the reliable routing path. In Fig. 2.5, from level 1 to level 3 are delta region (multi path) and level 4 is tributary region (single path). According to the data loss rate, delta region is to shrink or expand in the sensor field.

2.3 S-MAC Protocol

One of the major objectives of sensor network research is to prolong the network lifetime. With this condition, a medium access control (MAC) protocol was proposed, which reduces energy consumption of a sensor node [9]. This is known as S-MAC. One of the primary duties of the MAC protocol is to prevent data packet collision over the networks. To achieve this purpose, we need to know which parts are inefficient sources in the original MAC protocol.

2.3.1 Source of Energy Waste

- Collision

  When several nodes are trying to send a packet to one destination at the same time, packet collisions happen. All packets under collision are required to be discarded and retransmitted. Therefore nodes send the packet again, and sensor node energy is wasted.

- Overhearing

  When one node is trying to send the data to one destination node, other destination nodes within the radio range of sender also receive the packets. This is called overhearing. In this case, each node wastes the energy of overhearing.

- Control-Packet Overhead

  When each sensor node sends the data, it contains control packets for sending the data. If we make small control packets, we can save energy consumption for data transmission.

- Idle Listening

  Sensor nodes have idle condition for long time when there is no sensing event. Usually sensor networks have very low data rates before occurring events. Stemm and Katz show that the ratios of idel:receive:send are 1:1.05:1.4 respectively. Therefore idle listening is one of the dominant factors of energy waste.

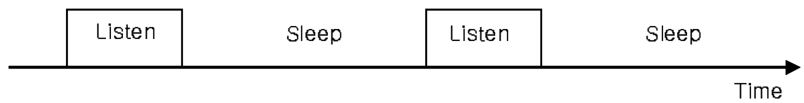


Figure 2.6: Periodic listen and sleep [9]

2.3.2 Periodic Listen and Sleep for Idle Listening

If no sensing event happens, sensor nodes have idle condition for a long time. It causes energy waste. S-MAC protocol reduces the idle listening with Periodic Listen and Sleep method. Each node goes to sleep mode and wakes up for listening to messages from neighbor nodes periodically. When each node goes to sleep mode, sensor node turns off its radio power and set time for waking. Fig. 2.6 shows Periodic Listen and Sleep scheme.

2.3.3 Collision Avoidance

At the same time, when two or more nodes send the data to one destination node, data collision happens. For preventing data collision, S-MAC protocol uses carrier sense (CS), and Request To Send (RTS) / Clear To Send (CTS) for the hidden terminal problem. First sender performs carrier sense before sending RTS. If a medium is not busy, sender sends RTS packet to receiver and sender's neighborhood nodes. After neighborhood nodes receive RTS, they go to sleep mode. Receiver sends CTS packet to sender and receiver's neighborhood nodes. After receiver's neighborhood nodes receive RTS, they also go to sleep mode. After neighborhood nodes of sender and receiver go to sleep mode, sender and receiver start to transmit messages.

2.3.4 Overhearing Avoidance

For overhearing avoidance, S-MAC protocol uses Network Allocation Vector (NAV). When they send the RTS/CTS packets, they also contain the time duration field which is how long is remaining for data transmission. Therefore neighbor nodes know how long they are in sleep mode. Each node stores this value called NAV. If NAV is not zero, that means the medium is busy. When the medium is busy, each node keeps sleep mode for overhearing avoidance.

2.3.5 Message Passing for Reducing Overhead

Transmitting a long single packet has the high cost of re-transmitting. However if we divide a long packet into several small packets, packet overhead will be increased because RTS and CTS packets are used in each small packet. In S-MAC, they use small packets and only once use RTS and CTS packets. When sender transmits its message to receiver, it makes a reservation of the medium for time to send all small packets. For that reason, S-MAC can use only RTS and CTS once for several small packets.

CHAPTER 3

NETWORK AND WAKEUP MODEL

3.1 Wakeup Model

In wireless sensor networks, we can divide data flow into two directions [57]. In the down direction, data flows from the base station to children nodes. In the up direction, data flows from children nodes to the base station. Our goal is to reduce the overhearing energy consumption when the base station transmits queries or data to children nodes. Hence in our wakeup model, we consider only down directional data flow. Fig. 3.1 shows our basic wakeup model based on [44]. In Fig. 3.1 (a), the radio range of node 1 covers node 2, node 3 and node 4. Therefore if node 1 intends to send the data to node 2, node 3 and node 4, they could all receive the data from node 1. However, node 5, node 6, node 7 and node 8 could not receive the data from node 1 directly because they are not within its radio range. When node 1 intends to send data to nodes within its radio range, wakeup of nodes not in its radio range such as node 5, 6, 7 and 8 is wasteful of energy, because idle listening consumes energy between 50% and 100% of receiving energy consumption [58]. In [59], Stemm and Katez show that the ratios of idle:receive:send are 1:1.05:1.4 respectively. Also the Digitan 2 Mbps Wireless LAN module specification illustrates idle:receive:send ratios is 1:2:2.5. Therefore, when node 1 tries to send the data to nodes 2, 3, and 4, the nodes which are not within its radio range such as node 5, 6, 7, and 8 should be synchronized to enter sleep mode for saving idle listening energy. In sleep mode, sensor nodes turn off their power. At the next step, after node 3 or node 4 receives data from node 1, when node 3 or node 4 intend to send the data to nodes within the radio range of sender, the nodes in level 3 wake up and they are ready to receive data from a node in level 2. Nodes in level 1 such as node 1 go to sleep mode again after they send the data to nodes

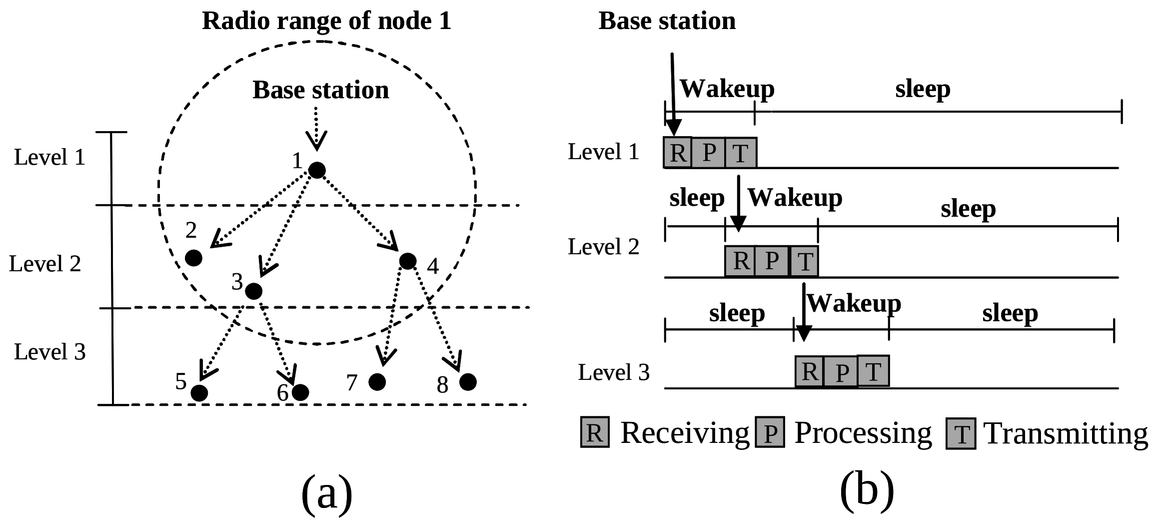


Figure 3.1: Example of wakeup model (a) Wakeup routing model (b) Wakeup time model

in level 2. All sensor nodes already know their level and their wakeup/sleep synchronized schedule through the setup of the initial tree structure. When the base station makes the initial tree structure, it sets the wakeup duration of each level in advance. For example, in Fig. 3.1 (b), if the base station decides that nodes wake up for 1 second for data transmission and wakeup for 1 second for data receiving, the nodes in level 1 wake up for 1 second for receiving data from the base station and also wakeup for another 1 second for transmission to the nodes of level 2. And then nodes of level 1 go to sleep mode. The time interval for data transmission between levels is allocated sufficient time to complete the process, otherwise failure of data transmission will be increased [44].

3.2 Time Synchronization for Wireless Sensor Networks

Wireless sensor networks consist of large number of sensor nodes. Each sensor node has local clock to be synchronized. Due to the unique characteristic such as limited battery power, memory size, wireless bandwidth and so on, traditional synchronization methods is not suitable for wireless sensor networks. Therefore, many researchers have investigated to design

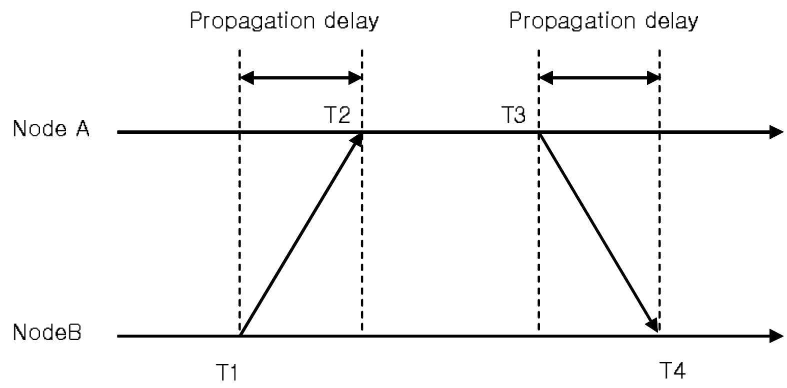


Figure 3.2: Two way message exchange [10]

synchronization protocol for wireless sensor networks. Usually wireless sensor networks do not have any special infrastructure for routing structure. Each sensor node collaborates to collect the sensing data for sending it to the base station by multi hop networks. In the multi hop network structure, time synchronization is one of the important issues. It is very hard to have same time on all the sensor nodes because of duration of time interval. If each sensor node has different time clock, it is very hard to combine the correct information from the neighbor nodes. Also, sensor network wakeup scheduling is useful to extend the sensor network lifetime as sensor node goes to sleep mode when there is no event. Therefore sensor nodes need to wakeup and sleep at the same time [10].

In [60], it proposed timing-syn protocol for sensor networks (TPSN). This protocol has two phases. First phase is level discovery phase and second phase is synchronization phase. In the level discovery phase, it makes root node which have a GPS. The root node becomes a level 0 and starts the level discovery phase. The root node sends the level discovery packet to the neighbor nodes. If neighbor nodes receive the level discovery packet, these nodes become level 1 nodes. As repeated by this way, all sensor node are assigned each level. In the synchronization phase, it makes the two way message exchange. In Fig.3.2, it illustrate two way message exchange. When node B sends the synchronization pulse packet at T1 time, node A receives it at T2 time. T2 has following value:

$$T_2 = T_1 + \triangle + d \tag{3.1}$$

Where $\triangle$ means time difference between node A and node B, d is propagation delay time for sending the message. After node A receives message, it sends acknowledgement packet to the node B. Therefore node B can recognize the time difference and propagation delay as below:

$$\triangle = \frac{(T2 - T1) - (T4 - T3)}{2} \tag{3.2}$$

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \tag{3.3}$$

3.3 Sensing Model

In wireless sensor networks, object sensing is an essential function. Sensor node has sensing device such as temperature, light, acoustic, magnetic and so on. Each sensor node has collaborative processing, information sharing, and group management [11]. Sensor network is defined by following set [11]:

$$G = \{V, E, C_V, C_E\} \tag{3.4}$$

In this set, V describe each sensor node, and E means edge of sensor network. C_V is the characteristic of each node such as processor capacity, memory size, location and so on. C_E is the characteristic of each edge of sensor network such as link quality and data sending rate and so on.

Fig. 3.3 illustrates the example of target sensing in a sensor field. The sensors located near by object are sensing the target object. Each sensor node sends the sensing data to the base station by collaborative processing. We can represent the sensor measurement as following model:

$${Z_i}^t = h(X^t, \lambda_i^t) \tag{3.5}$$

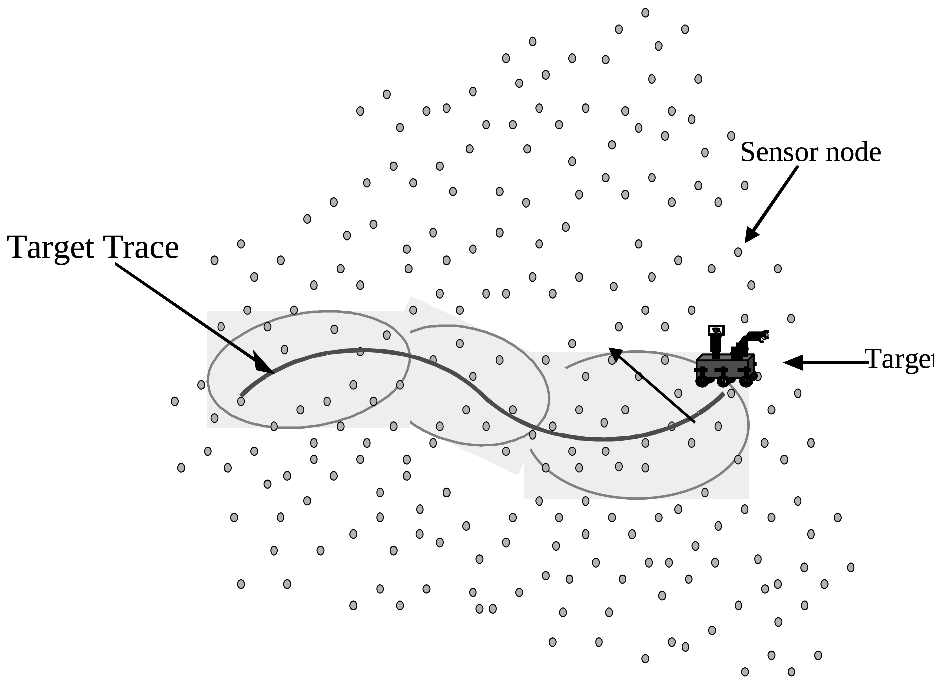


Figure 3.3: Sensing model in wireless sensor networks [11]

In this model, Z_i^t means measurements of sensor node I at t time. X^t means location of target object at t time. X_i^t represents the characteristic of sensor node i such as location and capacity and so on. Therefore $\lambda_i = [\zeta_i, \sigma_i^2]^T$ where ζ_i is the sensor position and σ_i^2 means additional noise variance.

CHAPTER 4

MULTI TREE SENSOR NETWORK ROUTING STRUCTURE

4.1 Introduction

Advances in wireless networks are expected to play an increasing role in systems that are aimed at collecting information. One of the main challenges in wireless sensor networks is that a sensor node has limited battery power. Therefore in order to increase the lifetime of sensor nodes, we need to reduce the amount of energy consumption. For reducing energy consumption in sensor networks, in-network aggregation is one of the proposed methods. However in-network aggregation does not keep the energy balance if some nodes are on the most frequently used paths in a network such as sink node. In order to consider more energy efficiency through load balancing, we propose a new in-network aggregation structure based on multiple trees, called MULT, for further extending the lifetime of in-network aggregation. Unlike existing in-network aggregation structures, which aim to reduce communication cost, the proposed MULT further provides energy balance. MULT has 3-phases: first building the clusters, second connecting the clusters and third making multiple trees. MULT is based on creating node clusters using distance between nodes. In addition, a new clustering method, called HYC (HYbrid Cluster) is introduced for MULT structure. We compare the MULT with LEACH and EAD, which are popular in-network aggregation methods. MULT outperforms LEACH and EAD for energy load balance.

The remainder of this chapter is organized as follows. In section 4.2, we provide a problem definition. In section 4.3, we introduce the three phases of the MULT algorithm including the HYC algorithm for clustering. The performance study is reported in section 4.4. Finally, section 4.5 presents the summary of this chapter.

4.2 Problem Definition

Wireless Sensor Network (WSN) technology is increasingly applied to research applications in the world of ad-hoc networks. Sensor nodes are usually less mobile and more densely deployed than ad-hoc networks [31]. A wireless sensor network is deployed with numerous small, battery driven sensor nodes with limited CPUs and memory, and finite radio range and bandwidth [61]. Among these challenges, the limited battery power is one of the main concerns in these sensor networks. Architecturally, with tens or thousands of sensors it is highly difficult to replace the battery. If we consider performance, it is observed that energy cost for communication is much larger when compared to the computational cost [62]. Hence to accomplish the task of reducing the energy consumption for sending messages, in-network aggregation processing is often used in sensor networks [44]. With the combining of partial results of sensor nodes in the intermediate nodes of the sensor fields, the in-network method reduces the energy consumption caused by communication cost. In the case of aggregation functions such as Max, Min, Sum, Count, and Average, intermediate nodes of the sensor field send the partially aggregated data to the parent nodes only one time. It saves considerable energy over the whole sensor field [8].

In in-network aggregation, sensor nodes may have a tree structure with the base station as its root [63]. Tree structures for in-network aggregation can be classified into two categories: tree structure with clustering [54, 64, 65, 55], and tree structure without clustering [32, 44, 55]. The aggregated data is computed in network by following the tree structure from the leaf nodes to the root of the tree, which is the sink node. This reduces the energy consumption by decreasing the number of messages transmitted to each node. However even though these approaches have reduced the number of transmissions to a great extent, they have some problems. If some nodes are on the most frequently used paths in the network, they will die sooner than others [31]. This causes the sensor network lifetime to rapidly decrease by collapsing the network structure. If we can attain energy load balance, it extends sensor network lifetime.

To achieve energy load balance in in-network aggregation, we propose to use multiple tree structures. We consider random deployment of sensor nodes, which are self-organizing and form a network in an adaptive way. Our method has three phases. The first phase constitutes assembling the sensors together to make clusters and then the second phase connects each head node of a cluster into a binary tree structure. In this case, the closest node from the base station becomes the root node of the first tree structure in the sensor field. In the third phase, multiple tree structures are constructed in a manner similar to the second phase but we do not use the same root node of the previous tree. Therefore, in the second tree structure case, the root node is the second closest node to the base station. For example, if the sensor network is queried from the base station for obtaining aggregation data, first, we use the first tree of the several tree structures. If we send another aggregate query, we use the second tree structure, and so on. This distributes traffic load balance by preventing the concentration of traffic to one sink node.

We also propose the hybrid clustering algorithm called HYC (HYbrid Cluster) for the first phase. For HYC, we combine the centralized and decentralized methods for clustering. Tree structures with clustering have been shown to be more efficient than tree structures without clustering [66] because they preserve limited energy resources and improve energy efficiency and provide scalability and robustness for the network.

4.2.1 Energy Model

For measuring energy consumption in sensor network, we assume the simple First Order Radio Model presented in LEACH [54]. In LEACH, transmitter and receiver defined as E_{elec} dissipate $50nJ/bit$ and transmit amplifier defined as amp consumes $100pJ/bit/m^2$. It also assumes the radio channel to be symmetric, which means the cost of transmitting a message from A to B is same as the cost for transmitting from B to A [54]. This radio model calculates the energy spent for k-bit packet to send over a distance 'd' as

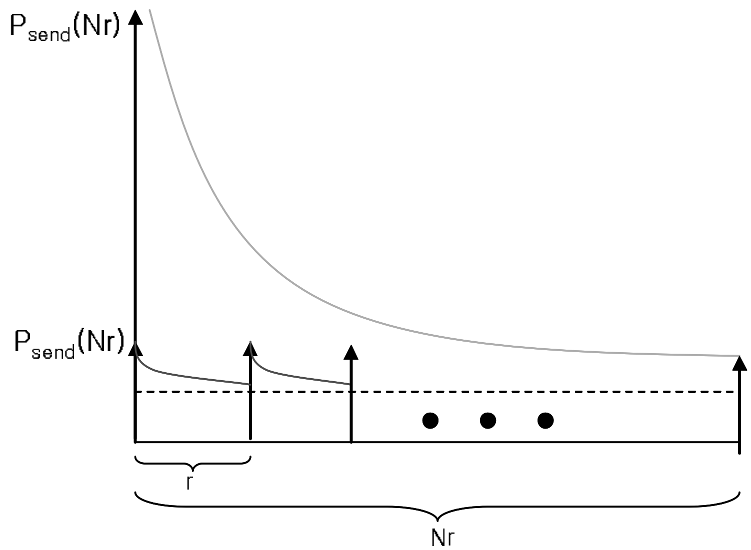


Figure 4.1: The power advantage of multihop structure [11]

$$E_{Tx}(k,d) = E_{elec} \times k + \varepsilon_{amp} \times k \times d^2 \tag{4.1}$$

$$E_{Rx}(k) = E_{elec} \times k \tag{4.2}$$

E_{Tx} is the energy used for transmission, and E_{Rx} is the energy used for receiving. It is clear from the above equations that the transmission energy is dependent on the distance parameter. The energy will be increasing at a high rate as the distance increases.Therefore it is very important to reduce the number of transmission because data transmission is the largest part of sensor network energy consumption. If we use the multi-hop routing structure. it is more energy efficacy than single-hop routing structure with long distance. Because as we decrease the value of distance parameter, we can save the the transmission energy consumption. Even though the cost of receiving is relatively low, we have to consider the design of protocols that can reduce energy cost for both the transmission and receiver circuitry communication. In Fig. 4.1, it shows relationship between multi-hop energy consumption and single-hop energy consumption. r means one hop distance of total distance and Nr means total distance.

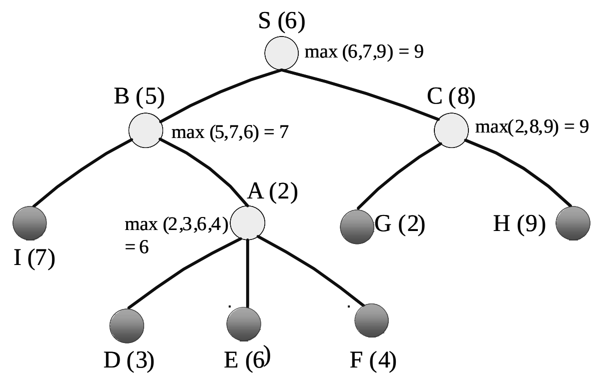


Figure 4.2: Example of data aggregation using binary tree

4.2.2 Routing Model

In wireless sensor network, constrained energy is one of the critical problems. For reducing the energy consumption, in-network aggregation is used in sensor network [67]. In our routing model, we adjust binary tree to in-network aggregation. MULT structure has two kinds of connections: the first connection is between head nodes of the clusters and the second connection is between the head node and non-head nodes within a cluster. Inside a cluster, each node is connected by HYC clustering algorithm introduced in section 4.3. Each head node of clustering is connected by a binary tree structure. Each node located within the cluster sends its data to the cluster head by in-network aggregation. After a cluster head node aggregates the data from each node in the cluster, it send the other cluster head nodes using multi-hop structure by in-network aggregation. in Fig. 4.2 illustrates the example of binary tree adjusted to in-network aggregation.

In Fig. 4.2, sensor node G aggregates the data from the sensor nodes F and I. Also sensor node O aggregates from sensor nodes M and Q and D from A and G. Finally the rooted sink K aggregates data from nodes D and O. When we retrieve any node, we can also go through the binary tree efficiently.

4.3 Energy Balanced Multi Tree Structure

The longevity of wireless sensor network is affected by each sensor node lifetime. If any sensor node is out of battery, wireless sensor network may change its routing structure to eliminate that node. As removing a dead node, distance of wireless transmission is extended. In wireless sensor network based on tree structure, the nodes located nearby base station will be out of battery sooner than other nodes which is located far away from the base station. Because Network traffic concentrates around the base station, it causes that the amount of the transmission is increased. Therefore we suggest the multi tree structure to prevent the concentrated traffic on sensor nodes located nearby the base station.

In this section, we describe the MULT and HYC clustering algorithm in detail. First, we describe how sensor nodes are assigned to each cluster during the clustering process. Second, we present how clusters are connected into a tree structure. Finally, we discuss how the tree structure extends to multiple tree structures. We assume the following properties for the MULT algorithm.

- Each node has a unique id and knows its position.
- Any node can send a message to the base station via multi-hop routing.
- All nodes have equal capabilities.
- All nodes can control the power of radio range.
- All nodes know the position of the base station.

4.3.1 Step 1: Creating the Clustering

In Step-1, we present the clustering algorithm, called HYC, based on the shortest distance from each node. HYC algorithm consists of two parts. The first part is to find the closest node from each sensor node distributed in the sensor field by a decentralized method. The second part is to determine which nodes are in the same cluster by a centralized method.

Algorithm 1 : Finding the closest node

Input to each node: position and unique id information of all nodes in the radio range of each node.

1. Compare distance to all nodes in the radio range.
2. Find $N_{closest}(v_i) = \{v_j \in V_I' \mid min\{d(v_j, v_i)\}\}$.

Output: *unique id of the* $N_{closest}(v_i)$.

Figure 4.3: Algorithm 1: finding the closest node

In the first part, all sensor nodes have the program that finds the closest node from each node. Let all sensor nodes set be $V = \{v_1, v_2, v_3 \cdots\cdots v_n\}$ and all sensor nodes except v_i be $V_i = V - \{v_i\}$. The closest node defined as $N_{closest}$ to v_i is defined as follows

$$N_{closest}(v_i) = \{v_j \in V_i^{'} | min\{d(v_j, v_i)\}\} \tag{4.3}$$

and $d(v_j, v_i)$ is the distance from v_j to v_i , so v_j is the closest node to v_i. All nodes can obtain the unique id of the closest node using Algorithm 1. Algorithm 1 works as follows: each node sends its position and node id to all nodes within its radio range. Each node then calculates from the received information its closest node. For example, in Fig. 4.4, node 3 and node 1 are in the radio range of node 2. The node 6 is out of radio range of node 2. Therefore node 2 received $id, position$ from nodes 1 and 3, and calculates node 3 as its closest node. Similarly, node 3 chooses node 2 as its closest node. All nodes send the output of Algorithm 1 to the one node among the neighbor nodes in radio range that is the closest node to the base station. Repeatedly, this neighbor node sends the information received from previous nodes to another neighbor node which is the closest node to base station. In this way, the base station obtains the output of Algorithm 1 from all sensor nodes through multi-hop routing.

In the second part, the base station which has received the result of Algorithm 1 from all sensor nodes creates Table 4.1. Table 4.1 is the input data for Algorithm 2. In Algorithm 2, variable *i* is the id of each node and variable *j* is the id of the closest node to each node.

Table 4.1: The result of algorithm 1 in the base station

Id of each node	1	2	3	4	5	6	7
Id of closest node	2	3	2	5	6	5	5

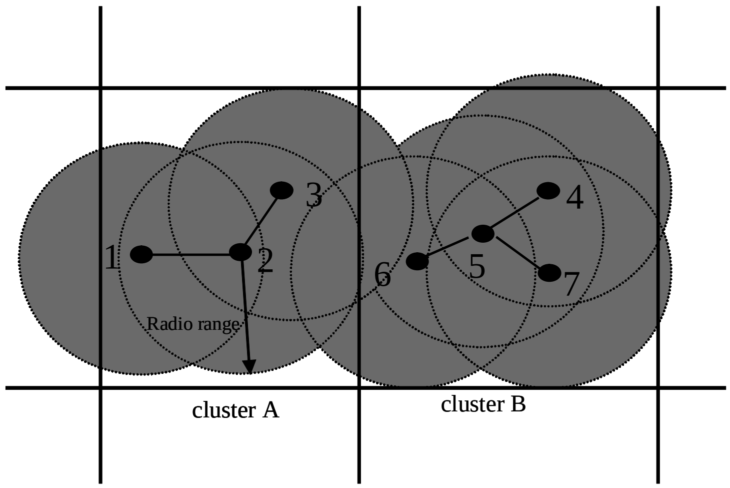


Figure 4.4: An example of making clusters

The result of Algorithm 2 is Table 4.2. In Table 4.2, node[1], node[2] and node[3] have same value as 3, also, node[4], node[5], node[6] and node[7] have same value as 7. This result implies nodes 1, 2, and 3 are in the same cluster and nodes 4, 5, 6 and 7 are in the same cluster. After phase-1, all the randomly placed sensor nodes in the wireless sensor network form into self-organized clusters.

Table 4.2: The result of algorithm 2 in the base station

Node[i]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Value of node	3	3	3	7	7	7	7

Also when we consider that all sensor nodes can form one gigantic cluster, we get a result that the possibility is very low. Fig. 4.6 shows that the number of clusters is increased as the number of nodes is increased.

Algorithm 2 : Finding the same cluster

```
Input: variable i = node id , variable j = the closest node.
Set all arrary node[i] = NULL
for(i=1 ; i<= number of node ; i++)
    if node[i] ≠ NULL
        tmp ← node[i]
        for(k=1 ; k <= number of node ; k++)
            if node[k] = tmp
                node[k] ← i
    if node[j] ≠ NULL
        tmp ← node[j]
        for(k=1;k<=number of node ; k++)
            if node[k] = temp
                node[k] ← i
    if node[j] = NULL
        node[j] ← i
    if node[i] = NULL
        node[i] ← i
```

Figure 4.5: Algorithm 2: finding the same cluster

4.3.2 Step 2: Connecting the Clustering

In Step-2, we illustrate how to decide the head node in each cluster and connect each head node into a tree structure. After the base station builds all tree structures using the data received from all sensor nodes, base station sends the information necessary to build the routing tree to all sensor nodes. As mentioned earlier in section I, the node which is the closest node from the base station become a first level root node of one tree structure of the sensor field and at the same time, also becomes the head node of a cluster. The first level root node is defined as R_{level0}. In the next step, the base station selects the closest node defined as $R1_{level1}$ and second closest node defined as $R2_{level1}$ from R_{level0} using the data sent from all sensor node. $R1_{level1}$ and $R2_{level1}$ are chosen to be in the different clusters than R_{level0}. Also $R1_{level1}$ and $R2_{level1}$ are not in the same cluster and become head nodes of each cluster. After constructing

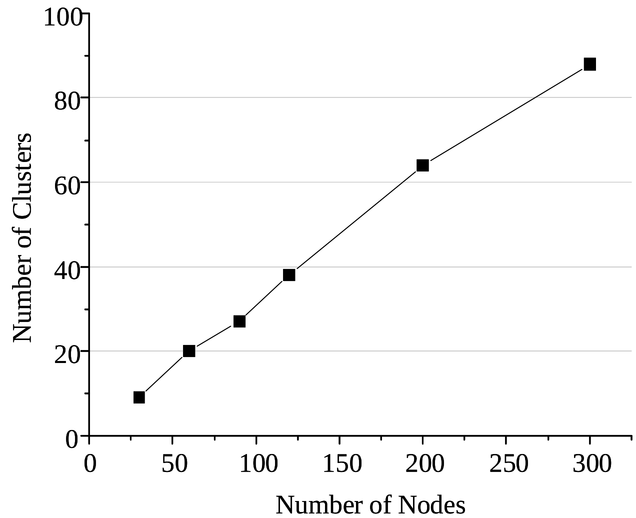


Figure 4.6: Number of clusters

the first tree, the base station sends the information describing the tree structure to the node R_{level0}. Also R_{level0} stores which nodes are in the same cluster with R_{level0} and which nodes become $R1_{level1}$ and $R2_{level1}$. Then, R_{level0} sends the information describing the tree structure to $R1_{level1}$ and $R2_{level1}$. This process is repeated recursively for subsequent levels. In this way, the tree routing information is transferred to the last level cluster head node. Fig. 4.7(b) shows the processing to build the first tree.

4.3.3 Step 3: Making Multi Tree Structure

In this subsection, we show how to make multiple trees. The previous section shows the construction of the first tree for aggregation of sensing data from all nodes. We reiterate the same method of making the first tree to make multiple trees. But the head node formed in the construction of first tree does not become a head node of another tree structure. Fig. 4.8 (a), (b) show the multi tree structure. Suppose that we built three binary tree structures in a particular sensor network. If the base station receives four queries from the user, the first tree structure is

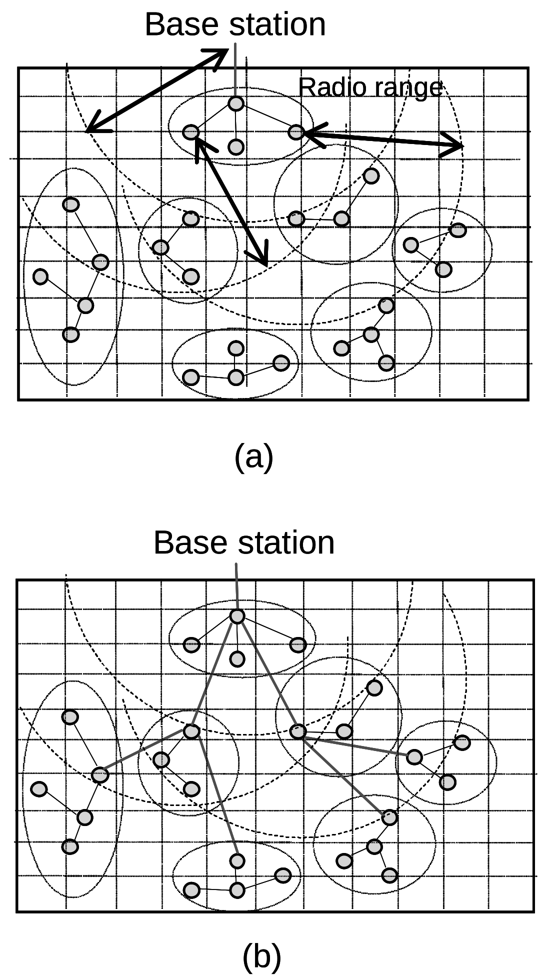


Figure 4.7: (a) Making cluster (b) Making first tree structure

used for the first query, second tree structure for the second query, third tree structure for the third query, and again, first tree structure is used for fourth query. Therefore we maintain the energy balance through the multiple trees to prevent concentrating data traffic to only one sink node.

In Step-3, the base station decides the number of tree structure. If the cluster which has the smallest number of nodes has two nodes, we can make two tree structures at the same

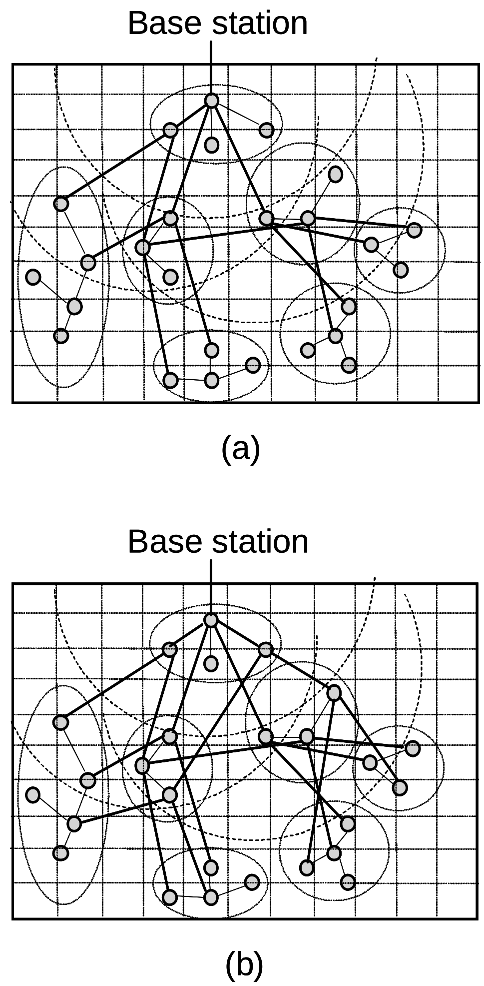


Figure 4.8: (a) Making second tree (b) Making third tree

time. But if the cluster having two nodes merges to nearest cluster, the number of tree structure could be increased.

4.4 Simulation Results

In this section, we present a simulation environment for MULT and the results of the simulation. We are interested in studying the energy efficiency for multiple tree structures based on clustering. Thus, we evaluated the performance of the multiple tree structure with the following metrics: 1) number of round times queries until the first node die, 2) energy consumption of the total sensor network field after one round time, and 3) remaining number of nodes after a certain number of query rounds times. We compare MULT with LEACH [8] and EAD[2] which are popular in-network aggregation methods. LEACH[8] structure has randomized cluster heads with single-hop wireless sensor networks and EAD[2] structure is based on a tree structure with multi-hop wireless sensor networks and without clustering.

In the experiments, homogeneous sensors are deployed in a $100 \times 100m^2$ network space. For the transmission distance, we used the four kinds of densities: 30 sensors/10^4m^2, 60 sensors/10^4m^2, 90 sensors/10^4m^2, 120 sensors/10^4m^2. For the radio range, all nodes can control the power of radio range. After building the multiple tree structure, the sensors transmit their sensed data to the base station when the base station sends a query. For measuring the energy consumption for transmitting the data, we used the LEACH energy model [8], using radio electronics energy *50nJ/bit* and radio amplifier energy *100pJ/bit*. We assume that the amount of energy in each node is considered as 1 Joule and the packet size is 2000 bits.

4.4.1 Aspect of Energy Duration

In sensor networks, it is important to reduce the energy consumption of each sensor node. We proposed the method to maintain energy balance using multiple tree structures to increase the sensor network lifetime. If any node dies in the sensor network, network structure rapidly collapses. In this experiment, we measure how long the energy balance is maintained. Fig. 4.9 (a) shows comparison results. In Fig. 4.9 (a), when we spread 30 nodes in $100 \times 100m^2$ sensor fields, in the case of MULT, the first dead node appears after 920 round times. In the case of LEACH [54] and EAD [32], the first dead node appears after 502 and 580 round times respectively. This result represents that MULT maintains energy balance longer than LEACH[8] and EAD [32] because MULT maintains sensor network without dead nodes for a longer time than EAD [32] and LEACH. Also this result shows that the higher the density

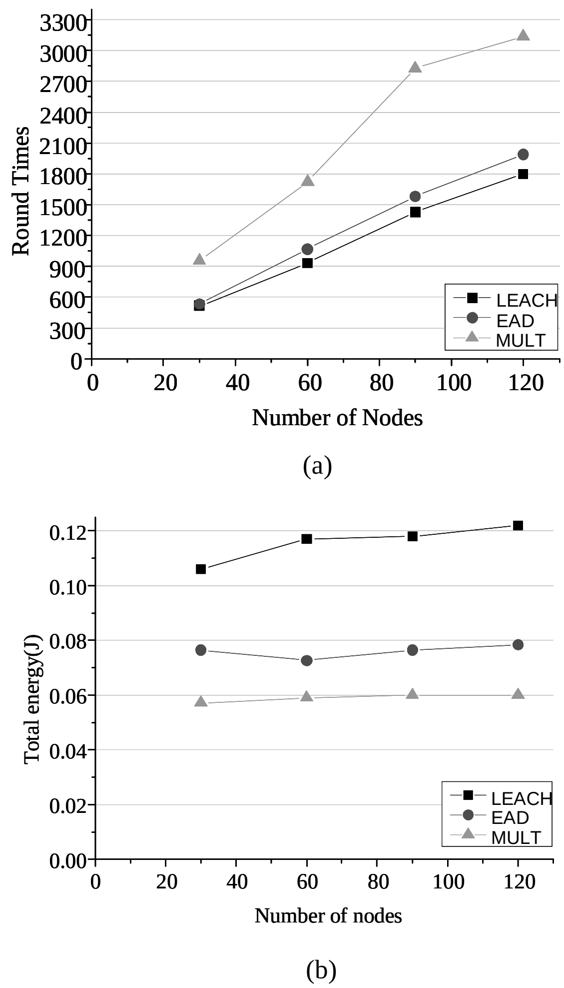


Figure 4.9: Simluation results (a) Round time (b) Total energy consumption

of sensors, the more increase in the number of round times because distance between nodes become closer.

4.4.2 Effects on Total Energy

In this experiment, we compared MULT with EAD [32] and LEACH [54] using the parameter of total energy consumption when the base station sends one query to the sensor network. Fig. 4.9 (b) show the result of this experiment. In Fig. 4.9 (b), when 30 sensor nodes are in the sensor network, the total energy consumption of the sensor field in MULT, EAD [32] and LEACH [54] is 0.0578J, 0.0786J and 0.1658J respectively. MULT structure uses the multi-hop routing structure with clustering. But in the LEACH [54] structure, head nodes of clusters send the sensed data to the base station directly without multi-hop routing structure and EAD [32] has no clustering even though it uses the multi-hop routing structure. Fig. 4.9 shows that multi-hop routing structure has better performance than single-hop routing structure and clustering structure is more energy efficient than non-clustering structure. Therefore MULT has better energy efficiency. In addition MULT can maintain the energy balance better as shown in Fig. 4.9.

4.4.3 Number of Nodes still Alive

In the third experiment, we examine how long MULT maintains the sensor network structure. Fig. 4.10 shows the result of this experiment. We experiment with three situations. First we spread 30 sensor nodes in $100 \times 100m^2$ sensor field and in the second and third cases, we spread 60 and 90 sensor nodes in $100 \times 100m^2$ sensor field. Even if we increase the number of node, the graph of the result has similar shape. So we did not increase the number of nodes more. In the experiment, if a head node dies, we choose another head node, which is the closest to the parent node of the dead head node. In Fig. 4.10, we can see that MULT sustains the network for a longer time than LEACH [54] and EAD [32]. Because MULT has multiple tree structures, it disperses the energy load balance. Therefore nodes can live longer than LEACH and EAD. Also, the gradient of MULT graph is higher than LEACH and EAD because of energy load balance. Hence, the number of dead nodes increases faster than LEACH and EAD. But total sensor network lifetime is longer than LEACH and EAD.

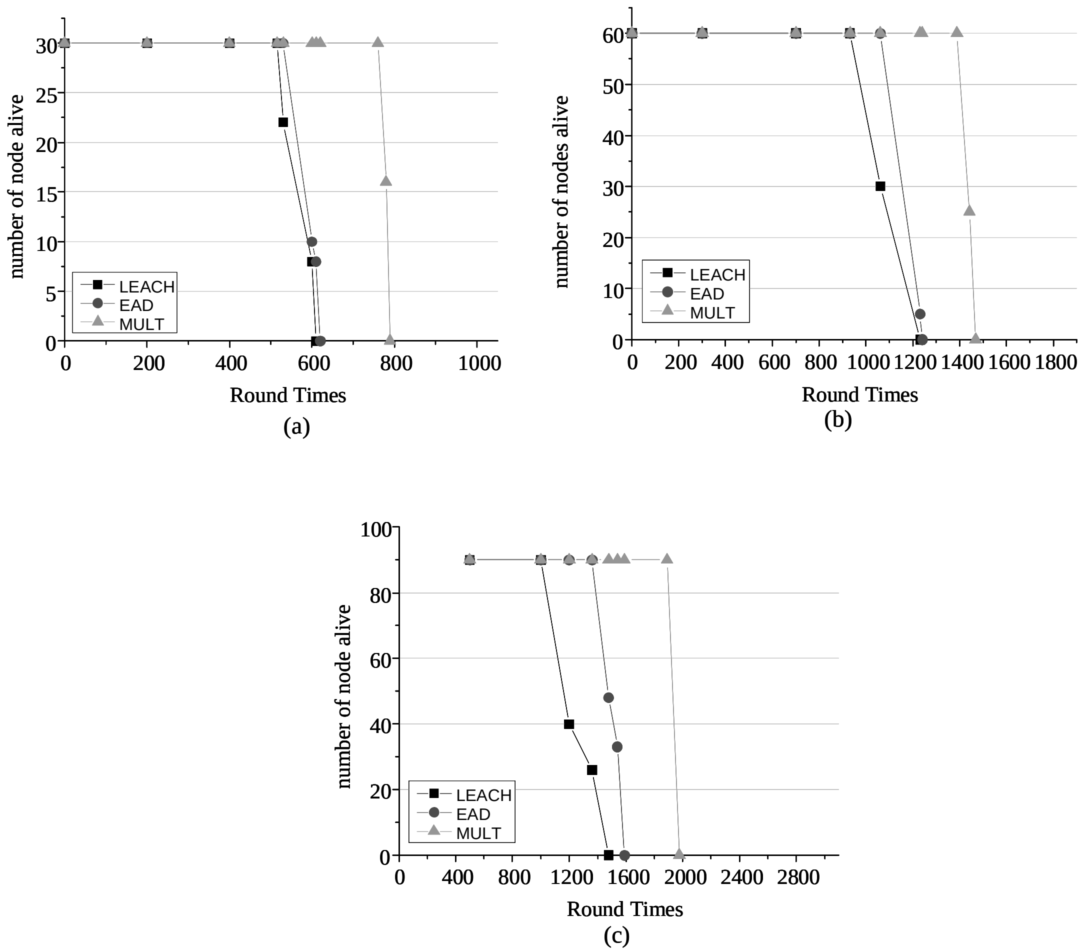


Figure 4.10: Number of nodes still alive (a) 30 nodes (b) 60 nodes (c) 90 nodes

4.5 Summary

We have described MULT, a multiple tree routing technique based on the minimum distances that keeps the energy balance in sensor networks. Effectively, we tried to reduce concentrating network traffic on special nodes. Also when we send the query to sensing nodes, we adjust the binary tree to the routing model. The advantage of using the binary tree connecting the head nodes is that we can retrieve the information through the binary search. For example, assume that there is query like ”What is the temperature of region A”. In this case, we can reach sensor nodes of region A through the binary tree structure. In the results of our simulation, MULT outperforms LEACH [54] and EAD [32] regarding the energy balance, and extending the lifetime of the WSN.

CHAPTER 5

STRATEGY FOR ENERGY EFFICIENT DATA GATHERING

5.1 Introduction

Advances in MEMS (Micro-Electro Mechanical System) have led to the emergence of wireless sensor networks. Each sensor consists of a processor unit, storage unit, wireless transmission unit, power unit and sensing unit [61, 66]. These sensor nodes are spread in a sensor field for measuring the environment. Each sensor node scattered in the sensor field is part of the network. When receiving a query from a user, the base station sends the query to nodes in the target area for collecting the information through the formed network. Because there is no wireless network infrastructure, each sensor node plays a role as either a routing or sensing node. The sensor node being of small size contains many restrictions such as limited battery power, low capability of processor, short radio range, and limited storage [67, 31]. The energy constraint is one of the most critical problems. It is almost impossible to replace the low level battery in many sensor nodes deployed in sensor fields. If we consider the aspect of energy consumption, we can observe that energy cost for transmitting is large when compared to the data processing cost [62]. Therefore in order to reduce the energy consumption for transmitting messages, many researchers try to find energy efficient techniques such as in-network aggregation [44], clustering [65, 64, 54, 68], various multi-hop routing schemes [32, 44, 55], and so on.

In the in-network aggregation, there are several advantages for minimizing the communication cost. Especially, partial results that come from children nodes are combined in each intermediate node and then the aggregated results go up to the parent node. This saves considerable energy over the entire sensor network [8].

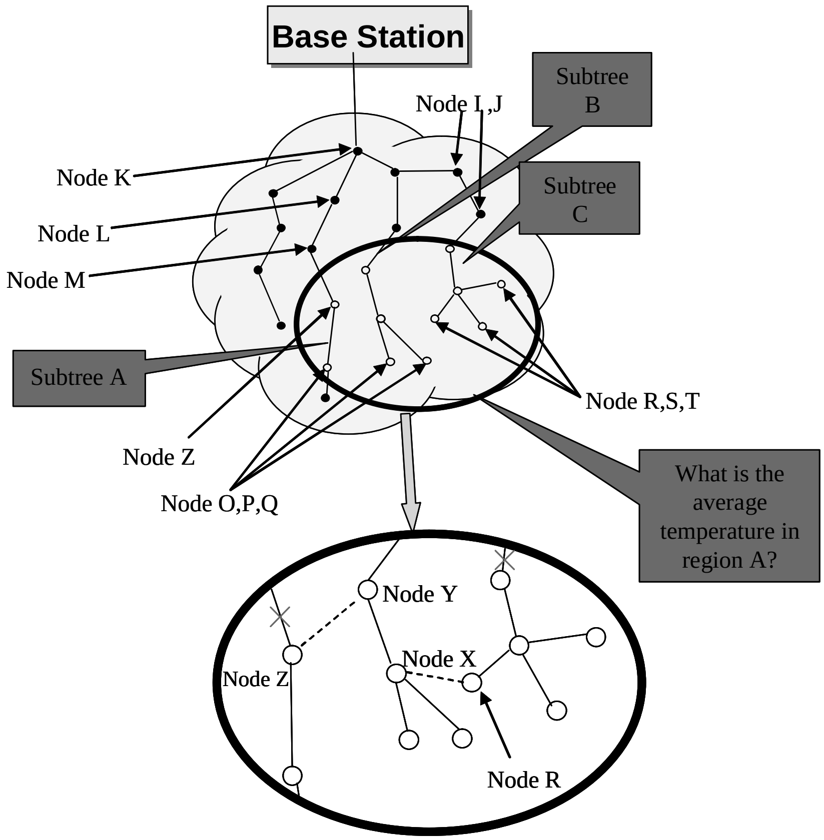


Figure 5.1: The example of merging the subtrees

One method to save energy consumption is through clustering [69, 70]. It can also be used to reduce the energy consumption for sending the messages. One cluster head collects the sensing data from neighborhood nodes and then transmits to the parent node or base station.

Clustering methods and in-network aggregation work in tandem with the routing schemes for the wireless sensor network. Usually, single-hop routing schemes, in which each sensor is directly connected to the base station, needs more energy than multi-hop routing schemes, in which sensor nodes are connected to the base station through intermediate nodes [32]. This is because energy consumption of transmission is relative to distance. However, even if we use methods such as in-network aggregation and clustering and multi-hop routing schemes, unnecessary energy can be used for routing. For example, in Fig. 5.1, if a base station receives a spatial query like *What is the average temperature in region A*, we have to use several routing subtrees to access the target area A. In this case, some ancestor nodes are used for routing unnecessarily. In the case of using the subtree B for routing, if we connect the subtrees A, B, and C to each other within the region A, we do not need to use the nodes K, L and M and also ancestor nodes of subtree C such as nodes I and J for routing. Therefore we can reduce the overall energy consumption.

For preventing the unnecessary energy consumption in ancestor nodes for routing, we propose the Sensor Network Subtree Merge algorithm, called SNSM, using the union of disjoint set forests algorithm [71]. SNSM algorithm has three phases. In the first phase, we find the disjoint set of the subtrees in the target area. In Fig. 5.1, there are three subtrees A, B and C in the target area A. Hence through phase 1, we recognize the disjoint subtrees in the target area. In the second phase, we try to connect each disjoint subtree with its closest node in the target area. In the third phase, we disconnect any subtrees connected to a new tree branch from the previous tree structure.

The remainder of this paper is organized as follows. In section 5.2, we provide a problem definition. In section 5.3, we introduce the three phases of SNSM algorithm. The performance study is reported in section 5.4. Finally, section 5.5 presents concluding remarks.

5.2 Problem Definition

Recently, wireless sensor networks have improved for many applications aimed at collecting information. However wireless sensor networks have many challenges to be solved. One of the most critical problems is the energy restriction. Therefore in order to extend the lifetime of sensor nodes, we need to minimize the amount of energy consumption. In many cases, sensor networks use routing schemes based on the tree routing structure. But when we collect information from a restricted area within the sensor field using the tree routing structure, the information is often assembled by sensor nodes located on different tree branches. In this case unnecessary energy consumption happens in ancestor nodes located out of the target area. In this paper, we propose the Sensor Network Subtree Merge algorithm, called SNSM, which uses the union of disjoint set forest algorithm for preventing unnecessary energy consumption in ancestor nodes for routing. SNSM algorithm has 3-phases: first finding the disjoint set of the subtree in the sensor field; second connecting each disjoint subtree with the closest node; and third virtually disconnect the subtree connected to new tree branch from previous tree structure. In the simulation, we apply SNSM algorithm to a minimum spanning tree structure. Simulation results show that SNSM algorithm reduces the energy consumption. Especially, SNSM is more efficient as number of sensor nodes in a sensor field increases.

5.2.1 Preliminaries

In the sensor network, a routing structure corresponds to undirected graph $G = (V, E)$. V is defined as the set of sensor nodes, $V = \{n_1, n_2, n_3 \cdots n_i\}$. E is the set of edges. Each sensor node $n_i \in V$ can send the sensing data within range of a radius denoted by R. If the distance between node n_i and n_j (denoted as d(n_i,n_j)) is within R, the edge between nodes n_i and n_j is defined as e_{ij}∈E. In this case, the weight w(n_i,n_j) denotes the cost to connect n_i and n_j, and is $d(n_i, n_j)$. In our sensor network, we apply SNSM algorithm to a minimum spanning tree structure used for routing. In this work, we assume the following properties for the suggested algorithm:

- The sensor nodes distributed over a geographical area are homogeneous and each has a unique node id.
- Each sensor node is aware of their position with the GPS.
- All sensor nodes can send a message to the base station via multi-hop routing and control the power of their radio range transmission depending on the distance

5.2.2 Energy Model

When transmitting and receiving the sensing data, each sensor node consumes energy. Therefore in order to measure the energy consumption in sensor network, we use the first order radio model presented in LEACH [10]. In this radio model, transmitter or receiver utilizes $E_{elec} = 50nJ/bit$ and there is a transmit amplifier defined as $amp = 100pJ/bit/m^2$. It also assumes the radio channel to be symmetric, which means the cost of transmitting a message between node A and node B is same bidirectionally. This radio model calculates the energy used for k-bit message to be sent over a distance 'd' as:

$$labelunicast E_{Tx}(k, d) = E_{elec} \times k + \varepsilon_{amp} \times k \times d^2 \tag{5.1}$$

$$E_{Rx}(k) = E_{elec} \times k \tag{5.2}$$

In formulas 5.1 and 5.2, E_{Tx} is the energy used for transmission and E_{Rx} is the energy used for receiving k bits of data. The transmission energy is dependent on the distance parameter 'd'. The energy will be increasing at a high rate as the distance increase. Hence as distances increase, multi-hop routing structure consumes less energy than single-hop routing.

5.2.3 In-Network Aggregation

In wireless sensor network, constrained energy is one of the critical problems. The advantages of using the in-network aggregation are to minimize energy consumption and incur no approximation error [8]. In in-network aggregation, each node computes the query in its own place, and produces a local result. For example, if base station receives the MAX aggregation query, each node receives the max values of the sensing data from children nodes, then applies its sensing data to the max value and sends the result to the parent node. In Fig. 5.2, sensor node G aggregates the data from the sensor nodes F and I. Also sensor node O aggregates from sensor nodes M and Q, and D from A and G. Finally the root node K aggregates data from nodes D and O. Hence in-network aggregation reduces the cost of transmission. Also, the

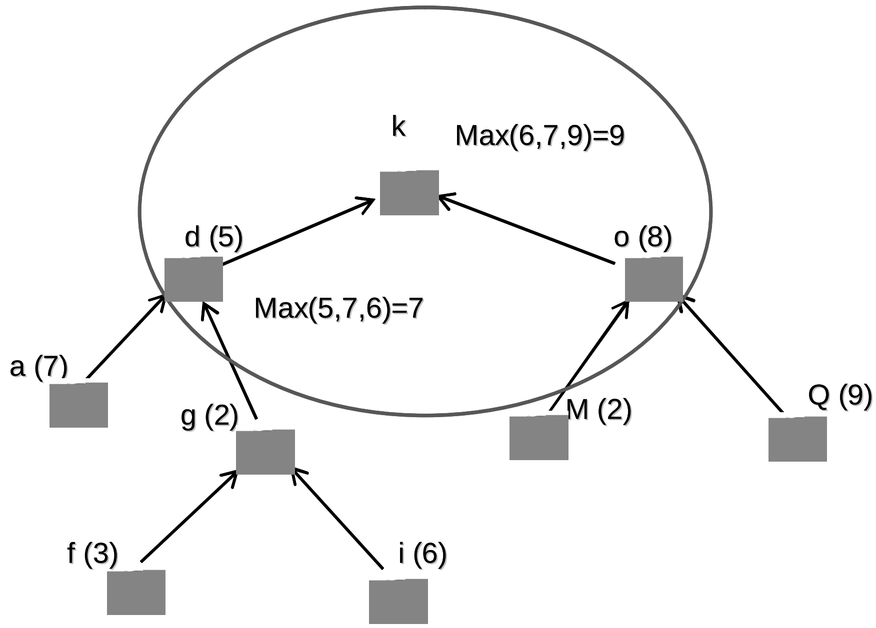


Figure 5.2: Example of in-network aggregation

value of max would be exact.

In order to send and receive messages, all nodes are synchronized [6]. For example, when node G has two children, node G is allotted enough time period, called epoch in TAG [6], for receiving sensing data from nodes F and I. If node G is not allotted enough time, node G sends the result to its parent without receiving the sensed data from nodes F or I. Therefore query response time is affected by epoch duration. Also epoch duration is dominated by the depth of the routing tree.

5.3 Sensor Network Subtree Merge Algorithm

In this section, we describe the sensor network subtree merge algorithm, called SNSM, using the union of disjoint set forest algorithm. The three phases of the algorithm are described in the following three subsections.

Algorithm 1 : Finding the disjoint subtree in target area.

```
Input : position of target area.
1:  find the leaf node in the target area
2:  for each n_i ∈ V[G]
3:     if n_i ∈ leaf node of target area
4:       then leaf[n_i] ← leaf node of target area
5:  for each n_i ∈ leaf[V]
6:     while ( leaf[n_i] has parent &&
              P(leaf[n_i]) is in the target area )
7:        do leaf[n_i] ← P(leaf[n_i]
8:     root_node[n_i] ← leaf[n_i]
9:     return root_node[n_i]
Output : root node of each subtree in target area.
```

Figure 5.3: Finding the disjoint subtree in target area

5.3.1 Phases-1: Finding the Disjoint Subtrees for a Given Range Query

In Phase-1, we describe Finding Disjoint Subtrees algorithm, called FD algorithm, based on the union of disjoint-set forests algorithm. After the sensor nodes are distributed in the sensor field, an initial routing tree is formed for sending the initial information of each sensor node to the base station. The initial information of each sensor node is denoted by D = {*position, unique id, id of neighborhood nodes*}. The base station constructs the minimum spanning tree for routing from the initial information D received from all the sensor nodes. This minimum spanning tree becomes the basic tree structure for routing. Once a spatial range query is submitted to the base station, the base station recognizes the disjoint subtrees within the target area of the query through the FD Algorithm $(Algorithm 1)$. For example, in the FD algorithm, if a base station receives the following spatial query from a user: "What is the average temperature in region A", the base station finds the sensor nodes within the target area A. Then the base station finds the leaf nodes within the target area (Line 1). Even if some sensor nodes have

Algorithm 2 : Finding the closest node over different subtree branch.

Input : Set of node in each subtree

for $n_i \in S_m$

  for $n_j \in S_n$

    do $min\_distance_i$ (n_i , n_j)

    if $min\_distanc_{i-1} > min\_distance_k$

    then *subtree_connector1* ← n_i

      subtree_connector2 ← n_j

connect (*subtree_connector1*, *subtree_connector2*)

Output : Connection of the two nodes with the closest distance over different subtree

Figure 5.4: Finding the closest node over different subtree branch

children nodes out of the target area, they become leaf nodes for this particular query. For example, in Fig. 5.1, nodes O, P, Q, R, S, T are leaf nodes in target area A. In lines 2 - 4, we save the leaf nodes to array leaf[n_i]. In line 6, $P(leaf[n_i])$ means parent of leaf node ni. Also $root\_node[n_i]$ is the root node of a subtree (Line 8). After finding the leaf nodes, we find the root nodes of the subtrees (Lines 5-9). If a leaf node has a parent node, its parent node become a leaf node recursively until we find a root node of a subtree within the target area (Lines 6-8). In Fig. 1, the node Z become a root node of subtree A within the target area. Therefore in Fig. 5.1, we find three root nodes in the target area A through the Algorithm 1.

5.3.2 Phase-2: Finding the Closest Node over Different Subtree Branches

In this section, we illustrate how to find the closest node in different subtree branches and connect the closest nodes to each other. Let all sensor nodes in subtree $'S_i'$ be the set $S_i = \{n_1, n_2, n_3 \cdots n_i\}$. For example, in Fig. 5.1, there are three subtree sets S_a, S_b, and S_c. In this case, the distance of node Z in subtree A and node Y in subtree B is the closest. Also, the distance of node X in subtree B and node R in subtree C is the closest. Hence through

Algorithm 2 we find the nodes which are connected with the closest distance in each pair of subtrees, such as node Z, Y, X, and R. Then, we connect node Z to node Y and node R to node Y. In Algorithm 2, n_i and n_j are nodes that belong to subtree S_m and S_n respectively (Line 1-2). In line 3, the $min\_distance_i(n_i, n_j)$ means minimum distance between ni and n_j. The minimum distance is defined as follows

$$min\_distance_i(n_i, n_j) = \{n_i \in S_m, n_j \in S_n \mid min\{distance(n_i, n_j)\}\} \tag{5.3}$$

In lines 5-6, $subtree\_connector1$ contains a node of the subtree S_m and $subtree\_connector2$ has a node of the subtree S_n. If we connect these two nodes, the distance of the two subtrees become the closest distance (Line 7).

5.3.3 Phase-3: Selecting the Routing Path

In this subsection, we show how to disconnect a subtree from the previous tree structure.The base station sends the changed routing information to subtree connectors such as Z, Y, X and R in Fig. 5.1. Among the subtrees in the target area, we chose only one subtree for routing. Therefore other subtrees send the sensing data to subtree through the subtree connectors. Also we have several criteria to select the subtree for routing in the target area. For example, in Fig. 5.1, because subtree B has a smaller number of ancestor node than node Y in its path to the base station as well as a longer depth in the target area A, node Y becomes the root node in the target area A. First, we check the number of ancestors and then if we have same number of ancestor, we check the depth of subtree for selecting the main subtree in a particular query target area. The subtrees A and C become the new branches of subtree B. Connection between node Z and node M is cut off for connecting between node Z and node Y.

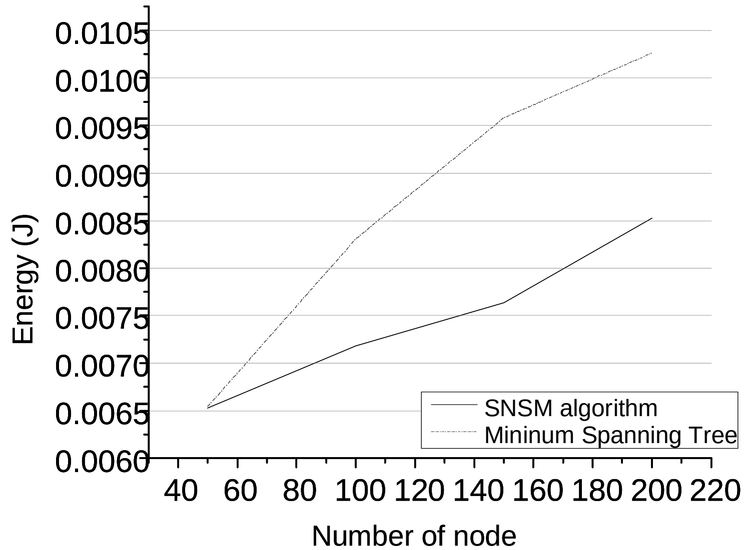


Figure 5.5: Energy consumption: SNSM vs MST

5.4 Simulation Results

In this section, we present a simulation environment for SNSM algorithm and the results of the simulation. We are interested in studying the energy efficiency for spatial query routing. Thus, we evaluated the performance of the SNSM algorithm with the following metrics: 1) energy consumption of the spatial query in sensor fields, 2) effect of sensor density, and 3) effect of range for spatial query area. We compare SNSM algorithm with minimum spanning tree used by many researchers [72]. In the experiments, homogeneous sensors are deployed in a 500 X 500m^2 sensor field area. We simulate 4 cases: 50 sensors/500 X 500m^2, 100 sensors/500 X 500m^2, 150 sensors/500X500m^2, 200 sensors/500 X 500m^2. For the radio range, all nodes can control the power of radio range. After connecting the disjoint set, each sensor in the target area transmits sensed data to the base station. Also for measuring the energy consumption for transmitting and receiving data, we used the LEACH energy model [54], using radio electronics energy 50nJ/bit and radio amplifier energy 100pJ/bit. We assume that the amount of energy in each node is considered as 1 Joule and the packet size is 1500 bits.

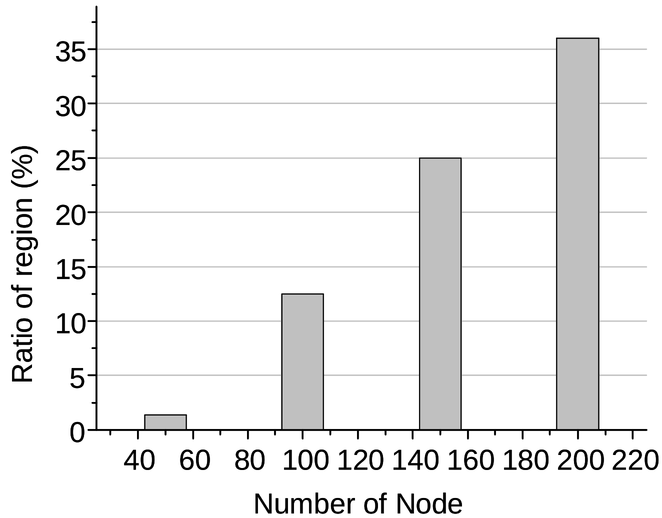


Figure 5.6: Effect of density

5.4.1 Energy Consumption for Spatial Query

In a sensor network, it is important to reduce the energy consumption of each sensor node. If some sensor nodes die earlier than others, even if other sensor nodes have enough energy, the entire sensor network structure can collapse rapidly. In this experiment, we compared SNSM algorithm with minimum spanning tree using the parameter of energy consumption of a spatial range query in the sensor field. For our experiment, we randomly chose the target area for the spatial query. The ratio of the query target area to the total sensor field area is 11%. In Fig. 5.5, as we increase the number of sensor nodes, energy consumption of total sensor nodes for spatial range query also increases. After we apply SNSM algorithm to minimum spanning tree, we obtain the result that energy efficiency of the merged tree is higher than minimum spanning tree structure. Especially, efficiency of energy consumption improves as the number of nodes increases. Therefore SNSM algorithm has better performance with a large number of sensor nodes than with a small number of sensor nodes.

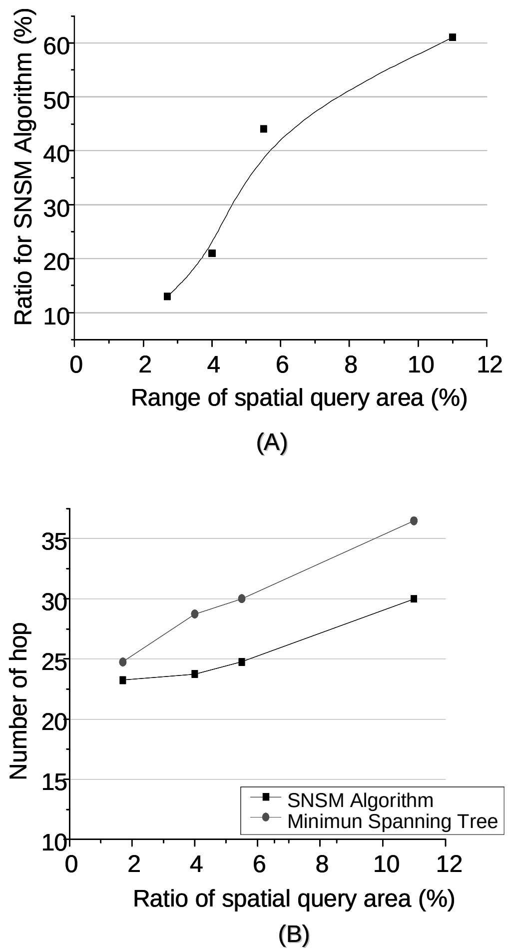


Figure 5.7: The efficiency on the ratio of a special query area (a) Ratio for SNSM algorithm (b) Number of hop

5.4.2 Effect on the Density of Sensor Nodes

In this experiment, we measured percentage of query regions to which the SNSM algorithm changes the tree structure as the total number of nodes increases. In Fig 5.6, when we spread 50 sensor nodes in the $500X500m^2$, SNSM algorithm improves performance in 1.38% out of the total sensor field area. But in the environment having 200 sensor nodes, the ratio of using SNSM increased to 36%. Therefore we can obtain the result that SNSM algorithm performs better in the environment that has high density of sensor nodes.

5.4.3 Ratio of Spatial Query Area

In the third experiment, we measured ratio for using SNSM algorithm and number of hops as we increased the range of the spatial query area. In Fig. 5.7 (a), we show the result of this simulation. In this result, as we increase the region of spatial query area, this increases the benefits of SNSM algorithm. For example, consider a spatial query like "What is the average temperature in region A". When the area of region A is 2.7% of entire sensor area, the ratio of using SNSM algorithm is 13%. Then, as we increase the area of region A to 11%, the ratio of using SNSM algorithm increases to 61%.

In the case of number of hops, as we increased the range of spatial query area, the number of hops also increases because of the larger target area, and the larger number of sensor nodes. In Fig. 5.7 (b), the number of hops in SNSM algorithm is less than minimum spanning tree structure. Therefore SNSM has better performance than minimum spanning tree structure.

5.5 Summary

In this paper, we have described SNSM algorithm based on the union of disjoint set algorithm when applied to minimum spanning tree. Also SNSM algorithm works on the in-network aggregation schemes for sensor networks. We reduce the energy consumption for routing in sensor network for spatial range query through the SNSM algorithm. In the simulation, we applied SNSM algorithm to minimum spanning tree. If there are other kinds of routing tree structures, SNSM algorithm can also be applied to those. As we mention in the simulation section, SNSM algorithm improves energy consumption in sensor networks with tree structures because we remove the redundant energy consumption in ancestor nodes for routing.

CHAPTER 6

ENERGY SAVING WAKEUP SCHEME

6.1 Introduction

A WSN (Wireless Sensor Networks) consists of a large number of sensor nodes. Each sensor node has limited battery, small storage, and short radio range. Many researchers have proposed various methods to reduce energy consumption in sensor nodes, since it is difficult to replace sensor node power sources [73, 74]. Generally, a sensor node consumes its energy during processing, receiving, transmitting and overhearing of messages that are directed to other nodes. Among those, overhearing is not necessary for correct operation of sensor networks.

In this chapter we propose a new synchronized wakeup scheme to reduce the overhearing energy consumption using different wakeup time scheduling for extending sensor network lifetime. The results of our simulation show that there is a trade-off between reducing overhearing energy and delay time. Therefore we propose Double Trees Structure, called DTS, having two routing trees, one based on Short Rings Topology and the other on Long Rings Topology. DTS has multi routing paths from base station to children nodes. If a node which is on the next routing path does not wakeup in time to receive the data, the sender node selects another path to connect to the destination. We can save the wait time until the next destination node wakes up. In the simulation result, our wakeup scheduling reduces overhearing energy consumption more than the S-MAC protocol. Using the double trees structure reduces the delay time.

The remainder of this chapter is organized as follows. In section 2, we provide a problem definition. In section 3, we introduce the OEWS and IWS for reducing overhearing energy consumption. In section 4, we suggests DTS for reducing delay time and we show the simulation results in section 5. Finally, section 5 presents summary of this chapter.

6.2 Problem Definition and Energy Model

6.2.1 Problem Definition

Wireless sensor networks are increasingly applied to various physical worlds for surveillance applications. Because large numbers of sensors are typically deployed, the trend has been to decrease the cost of each sensor node. As a result, a sensor node has smaller size than before. Therefore there are various capacity limitations such as the small amount of battery, limited storage, and short radio range [75, 31]. Even if each sensor node has small capacity, the large number of sensors can cover a large area by cooperating with each other to form a multi-hop wireless network. Nevertheless low battery power is one of the most crucial problems because it is hard to replace or recharge the battery in each sensor [30]. The life time of a sensor network depends on the energy in each senor node. To increase the life time of sensor networks, we need to reduce the energy consumption. Generally, a sensor node consumes its energy during processing, receiving, transmitting, and overhearing of parts of messages that are not directed to the node. Among those, the energy wasted by overhearing energy consumption is not necessary for correct working of the wireless sensor networks. A characteristic of wireless networks is that some nodes that are not a destination have to receive unnecessary messages because they are within the radio range. This is called overhearing. As node density increases and radio range grows, energy consumed by overhearing also will increase. In order to achieve the purpose of reducing the energy consumption, synchronized wakeup scheduling is used to make a node stay in sleep mode when messages are not directed to the node. We focus on reducing overhearing energy consumption with wakeup scheduling. For reducing overhearing energy consumption we propose a new wakeup scheme using different wakeup times between neighbor nodes. We call the new wakeup scheme Odd and Even Wakeup Scheduling (OEWS). We compare OEWS with the S-MAC protocol which is one of the popular MAC protocols for sensor networks [9]. In simulation, OEWS shows good results to reduce overhearing energy. This method improves the energy efficiency and increases the sensor network lifetime. OEWS adjusts different wakeup times for sibling nodes. A node in sleep mode will turn off its radio and will not overhear messages. There is a trade-off between energy efficiency and delay time because the node which intends to send the data has to wait until its next destination node wakes up according to a pre-defined synchronized schedule. For reducing delay time, we propose another new tree structure called Double Trees Structure (DTS). For reducing the data delay time, there are many methods using various wakeup scheduling patterns [76]. However, even if those wakeup patterns are efficient to reduce data delay time, it is hard to adjust those wakeup patterns to OEWS because they didn't consider overhearing energy consumption. Hence, we propose a new routing tree structure called DTS for reducing data delay time on OEWS. DTS has two tree structures called Short Rings Topology (SRT) and Long Rings Topology (LRT). SRT and LRT have the same number of hops from the base station to children nodes. There is no different delay time between SRT and LRT because they have the same number of hops. Therefore, by using multiple paths, we can save the waiting time for children nodes to wake up. The contribution of this paper is that we explore reducing both overhearing energy and latency together. We propose the OEWS to reduce overhearing energy and DTS for decreasing the latency. Overhearing energy is not necessary for operating the sensor network. Therefore it is important to reduce the overhearing energy for extending the lifetime of sensor networks. In the simulation, OEWS reduces the overhearing energy up to 43% compared to S-MAC protocol. Also, DTS helps OEWS reduce latency up to 30.4% than OEWS without DTS

6.2.2 Energy Model

For measuring energy consumption in a sensor network, we need an energy model. A sensor node consumes energy by transmitting, receiving and overhearing. E_{tx} is the energy used for transmitting and E_{rx} is the energy used for receiving. We assume the energy model including overhearing energy based on [77]. This radio model calculates the energy spent for one bit to send over a distance 'd' as

$$E_{tx} = E_{txelec} + \varepsilon d^2 \tag{6.1}$$

$$E_{rx} = E_{rxelec} \tag{6.2}$$

E_{txelec}, the energy consumption by transmitter electronics, dissipates $50nJ/bit$. We assume that E_{txelec} is the same as E_{rxelec} (receiving energy) based on [54]. We suppose that ε, which is an amplifier characteristic constant, is $100pJ/bit$. The model assumes the radio channel to be symmetric, which means the cost of transmitting a message from A to B is the same as the cost for transmitting from B to A [54]. Overhearing energy consumption is defined by E_{oh} as

$$E_{oh} = E_{rxelec} \tag{6.3}$$

In S-MAC protocol [9], when sending data from one sensor node to others, RTS(ready to send), CTS(clear to send), and ACK packets are necessary. Therefore, based on (6.1), (6.2), and (6.3), total energy consumption from node i to node j is represented by the following:

$$\begin{aligned} E_{ij} &= |RTS + DATA| \times (\varepsilon {d_i}^2 + E_{txelec}) + |CTS + ACK| \times (\varepsilon {d_j}^2 + E_{txelec}) \\ &+ |RTS + CTS + DATA + ACK| \times E_{rxelec} \\ &+ N_{RTS} \times |RTS| \times E_{rxelec} + N_{CTS} \times |CTS| \times E_{rxelec} \end{aligned} \tag{6.4}$$

Here, d_i is the radio range of node *i* and d_j is the radio range of node *j*. N_{RTS} and N_{CTS} is the number of neighbors which overhear the RTS and CTS packets respectively. $|RTS, CTS, ACK, or DATA|$ is the size of the packet in bits. In OEWS, total energy consumption is different from S-MAC. OEWS has two kinds of energy models. One is for nodes which have an odd id number and the other is for nodes which have an even id number. Therefore, total energy

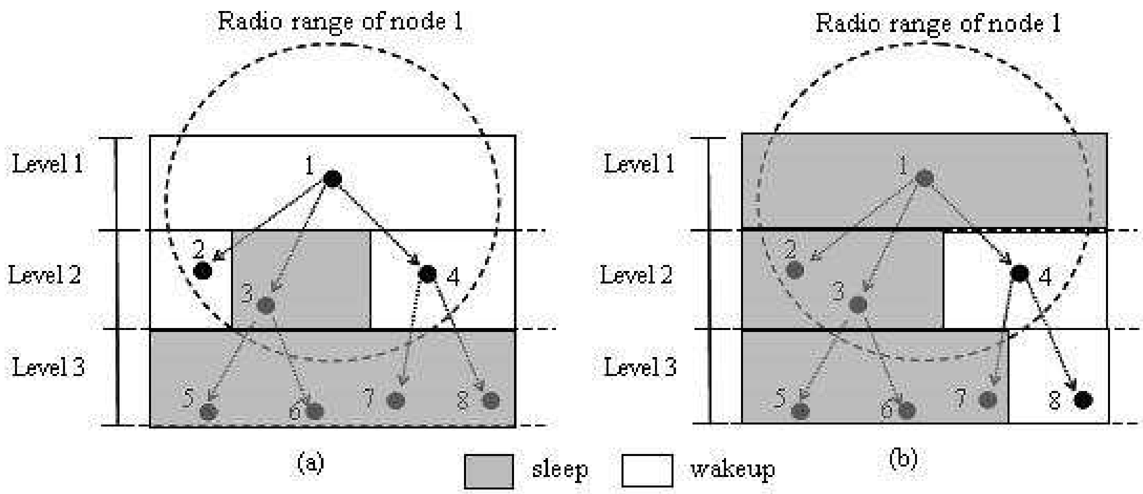


Figure 6.1: Odd and even wakeup scheduling example (a) Node 1 to 4 (b) Node 4 to 8

consumption from node *i* to node *j* which both have odd id number (or even id number) is represented by the following:

$$\begin{aligned} E_{ij} &= |RTS+DATA| \times (\varepsilon {d_i}^2 + E_{txelec}) + |CTS+ACK| \times (\varepsilon {d_j}^2 + E_{txelec}) \\ &+ |RTS+CTS+DATA+ACK| \times E_{rxelec} \\ &+ N_{oddRTS}(or N_{evenRTS}) \times |RTS| \times E_{rxelec} \\ &+ N_{oddCTS}(or N_{evenCTS} \times |CTS| \times E_{rxelec} \end{aligned} \quad (6.5)$$

Here, $N_{odd\_CTS}(N_{even\_CTS})$ is the number of odd (Even) id neighbors which overhear CTS and $N_{odd\_RTS}(N_{even\_RTS})$ is the number of odd (even) id neighbors which overhear RTS.

6.3 Energy Saving Wakeup Scheduling

In this section, first we describe the Odd and Even Wakeup Scheduling (OEWS) and Individual Wakeup Scheduling (IWS) techniques to reduce overhearing energy consumption and then, we describe the trade-off energy saving and delay time.

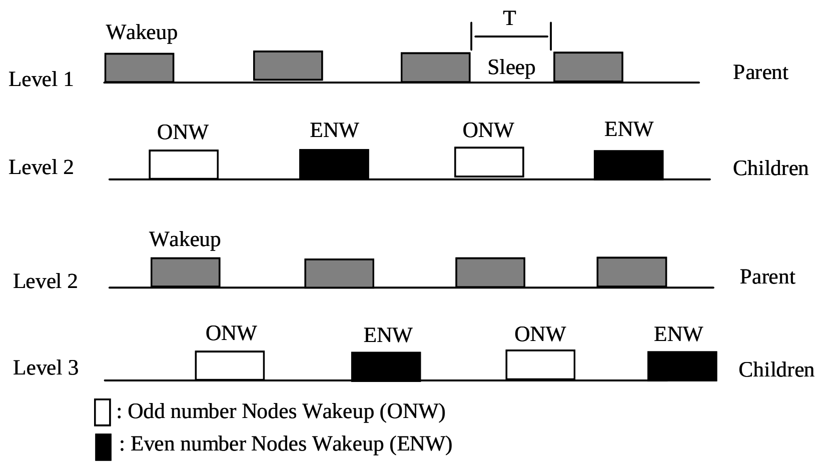


Figure 6.2: Odd and even wakeup time scheduling

6.3.1 Odd and Even Wakeup Scheduling (OEWS)

We propose the new wakeup schedule named Odd and Even Wakeup Scheduling (OEWS) whose purpose is to reduce the overhearing energy consumption as half the sensor nodes at the same level wake up alternately. In this scheme, sensor nodes having even id number and those having odd id number in the same level have different wakeup time schedule. Sensor nodes having even id number wake up at even time points and sensor nodes having odd id number wake up at odd time. For example, in Fig.6.1 (a), at a specific even time, sensor node 1 in level 1 can send the data to node 2 and 4 having even number id because node 3 having odd id number is in sleep mode at even time. But at a specific odd time, only node 3 can receive the data from node 1. If node 1 wants to send the data to node 8, node 1 sends the data at the specific even time for sending the data to node 4. Even if node 2 receives the data from node 1 at the same time with node 4, if the destination is not node 2, node 2 goes to the sleep mode. We still save the overhearing energy of node 3 and also we can reduce the wakeup time of node 2. In the next step as shown in Fig.6.1 (b), after node 4 receives data from node 1, nodes in level 1 fall in sleep mode again and nodes in level 3 wake up for receiving data from nodes in level 2 based on [44]. Nodes 5 and 6 in level 3, they go to sleep mode after they recognize

Parent Algorithm

```
Input : j=0, n_i = number of nodes,
if Schedule = children_wakeup_schedule
    then change to parent_wakeup_schedule
for n_i ∈ V[G]
    if n_i ∈ children of current parent node
        children[j] ← n_i
        j=j+1
for target_child_id ∈ children[j]
    if target_child_id ≠ children[j]
        then wait( )
    if target_child_id = children[j]
        then send ( data )
if level_wakeup_time = 0
    then sleep ( until next wakeup time )
```

Figure 6.3: Parent node algorithm

that there is no data from node 3 directed to them. Node 7 and 8 which are children of node 4 wake up alternately in even time and odd time. If node 7 does not receive the data at odd time, node 7 also directly goes to the sleep mode. And then node 8 can receive the data from node 4 at even time. Hence we can save overhearing energy through the odd-even wakeup schedule. Also, we can decrease the duration of wakeup time for nodes which are not a destination, as they can go to sleep mode if no message is received.

Fig. 6.2 shows the Odd and Even wakeup Scheduling. We use a synchronous wakeup schedule. Level 1 and level 2 in Fig. 6.2 correspond to node 1 and nodes 2, 3, 4 from Fig. 6.1 respectively. First when level 1 transmits data to level 2, the nodes in other levels are in sleep mode. Level 1 follows the schedule for a parent and level 2 follows the schedule for children.

Children Algorithm

```
1: if Schedule = parent_wakeup_schedule
2:     then change to children_wakeup_ schedule
3: find_parent ( )
4: if parent send the data
5:    then receive ( data )
6: if level_wakeup_time = 0
7:    then sleep (until next wakeup time)
```

Figure 6.4: Children node algorithm

A parent node wakes up at every time slot but children nodes wake up alternately. After a node in level 2 receives the data from a node in level 1, the node in level 2 switches from children wakeup time schedule to parent wakeup time schedule. And then the nodes in level 1 are in sleep mode and the nodes in level 3 wakeup and follow the children wakeup time schedule.

In Fig. 6.3, we present the algorithm for a parent node. In lines 1 - 2, a parent node checks its wakeup schedule. If it still follows the children schedule, it changes to parent schedule. In lines 3 - 6, we store the children nodes of the current parent into array children[j]. We defined G as an undirected graph and V is a set of sensor nodes. In lines 7 - 11, if parent wakeup time matches with a wakeup time of the target child node, the parent node sends the data to the target child node. Otherwise the parent node waits until its wakeup time matches with the target child wakeup time. In lines 12 - 13, if parent level has time out, parent goes into the sleep mode.

In Fig. 6.4, we also show the algorithm for children nodes. In lines 1 - 2 they check their wakeup schedule. If children follow the parent schedule, it changes to children schedule. In line 3, each child tries to find the parent node. In lines 4 - 5, they wait for the data from the parent node. If a specific child node receives the data from the parent node, it becomes a new

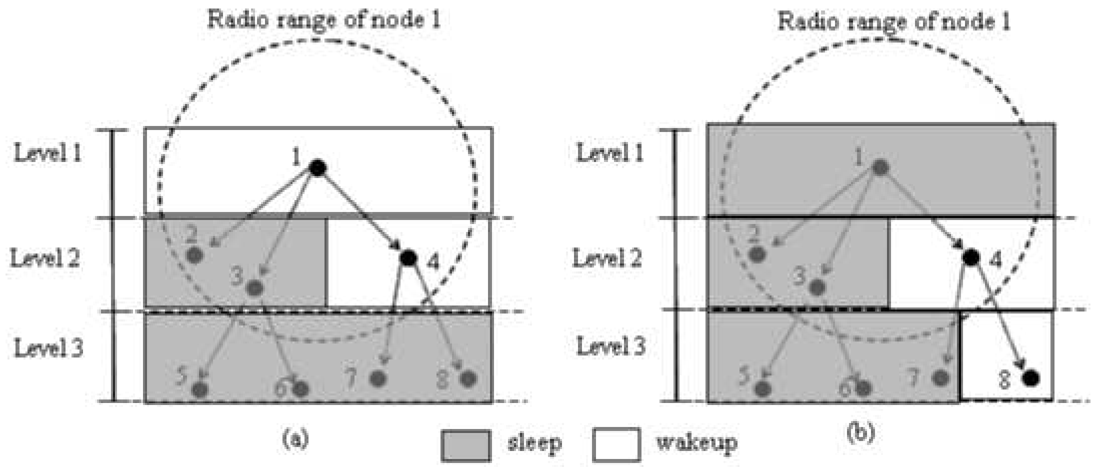


Figure 6.5: Individual wakeup scheduling example (a) Node 1 to 4 (b) Node 4 to 8

parent and other sibling nodes in same level go into sleep mode.

6.3.2 Individual Wakeup Scheduling (IWS)

We propose another wakeup scheduling named Individual Wakeup Scheduling (IWS). Fig. 6.6 shows the Individual Wakeup Scheduling. Each child node has a different wakeup time schedule than other children nodes in the same level. Therefore, at some specific time, only one child node wakes up and the other children nodes are in sleep mode. For example, in Fig. 6.5 (a), when node 1 intends to send the data to node 4, node 1 waits until node 4 is in its wakeup time. The parent knows the wakeup schedule of each child node. When node 4 wakes up, other child nodes like node 2 and 3 are in sleep mode. Hence node 2 and node 3 would not receive the data from node 1 and do not overhear the packet. In Fig. 6.5 (b), after node 4 receives the data from node 1, node 4 follows the parent wakeup schedule. In the parent level, even if other nodes such as node 2 and 3 are in level 2, only node 4 wakes up. In IWS, we can use the wakeup schedule for routing. In the case of the S-MAC protocol, control packets contain source and destination nodes. But IWS sends the data to the destination node

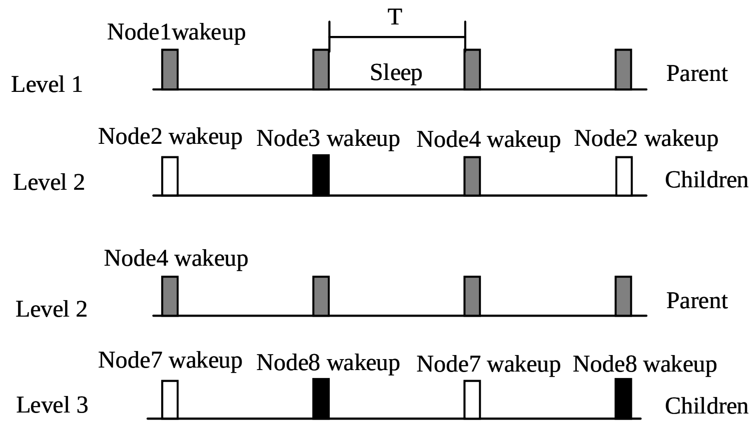
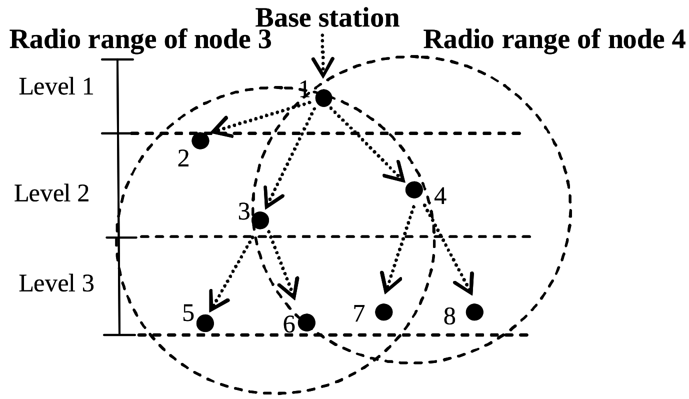


Figure 6.6: Individual wakeup time scheduling

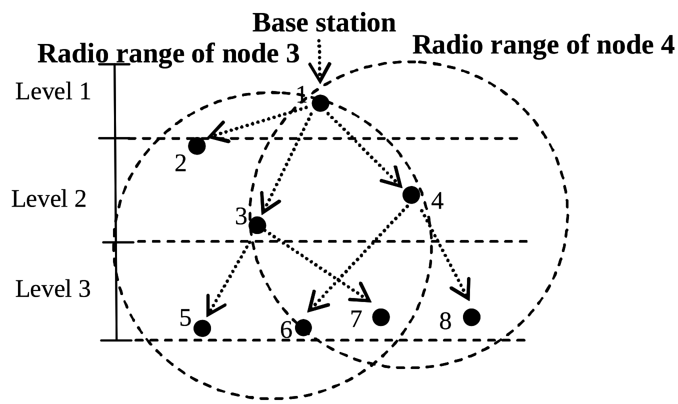
through the wakeup time schedule. Therefore we can save the control packet size of source and destination. Another advantage is that IWS has zero overhearing energy consumption.

6.3.3 Trade-off Energy Saving and Delay Time

When we try to send the data from a source to a destination in wireless sensor networks, there is a delay time. We assume the delay time based on [78]. Delay time is the time elapsed between the departure of a data packet from the source sensor and its arrival to destination [9]. Therefore we can denoted the delay by $DT(s,d) = (qd + td + pd + wd) \times Nd(s,d)$, where qd is queuing delay, td is transmission delay, pd is propagation delay and wd is waiting delay until the receiver node wakes up. Nd(s,d) denotes total number of data disseminators on the routing path between the source node 's' and the destination node 'd'. In OEWS, because we use the wakeup scheduling for the nodes to wake up alternately, it causes longer delay time. Therefore we suggest the Double Tree Structure called DTS to reduce the average delay time.



(a)



(b)

Figure 6.7: (a) Short rings topology (b) Long rings topology

6.4 Reducing Latency with Double Tree Structure (DTS)

For OEWS, we make two Rings topologies for the routing tree structure based on [72]. Two Rings topology consists of Short Rings topology called SRT and Long Rings topology called LRT. Rings topology makes a tree structure based on the radio range.

6.4.1 Short Rings Topology and Long Rings Topology

Fig. 6.7 (a) shows Short Rings Topology (SRT). SRT starts from the base station. In the first step, all nodes within the radio range of the base station become children nodes of the base station. For example, in Fig. 6.7 (a), the only node within the radio range of the base station is node 1. Therefore node 1 becomes a child node of the base station. In the second step, all nodes within the radio range of node 1 become children nodes of node 1. There are nodes 2, 3, and 4. In the third step, nodes 5, 6 and 7 become children nodes of node 3 because these are within the radio range of node 3. And nodes 6, 7, and 8 become children node of node 4 because these are within the radio range of node 4. Notice that node 6 and node 7 are included as children of both node 3 and node 4. In this case, we can divide them into two Rings topology with distance from parent node to children node. In the view of node 6, node 3 is the closest parent node. Therefore, if node 6 becomes a child of node 3, this topology is Short Rings topology. Otherwise, if node 6 is connected with node 4 which is the most far away from node 6 within the radio range, this topology is Long Rings topology. Fig. 6.7 (b) shows Long Rings Topology.

Our wakeup scheme uses both SRT and LRT in the wireless sensor network. Therefore, it is possible that there are several routing paths. For example, if the base station intends to send the data to node 6, in the first step, the base station sends data to node 1. In the second step, node 1 could send the data to node 3 or node 4. Node 4 can connect to node 6 through the LRT, and node 3 can connect to node 6 through the SRT. Therefore it does not need to wait until odd nodes wake up or even nodes wake up. If there is only one routing path, node 1 has to wait for even nodes wake up or odd nodes wakeup. As a result, we can reduce the delay time.

Even though SRT and LRT are decided by distance, energy consumption for transmitting, receiving and overhearing is the same whether we use SRT or LRT. Because all sensor nodes have the same fixed radio range, radio range of SRT and LRT is the same. Also, whether we use SRT or LRT, the number of hops from base station to destination is the same. Even if we change the routing path from SRT to LRT or from LRT to SRT, it does not change the number of hops or the radio transmission range.

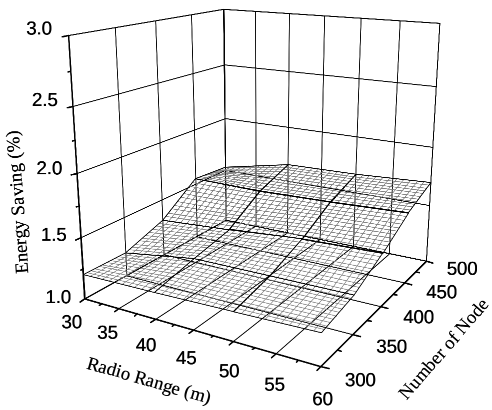
6.5 Simulation Results

In this section, we present simulation results of OEWS and DTS. Our simulation results show that OEWS helps to reduce the overhearing energy consumption and DTS decreases the data delay time. Therefore, we evaluated the efficiency of energy consumption and latency comparing with S-MAC protocol [9]. In S-MAC one of the sources of wasted energy which they tried to reduce is the overhearing energy.

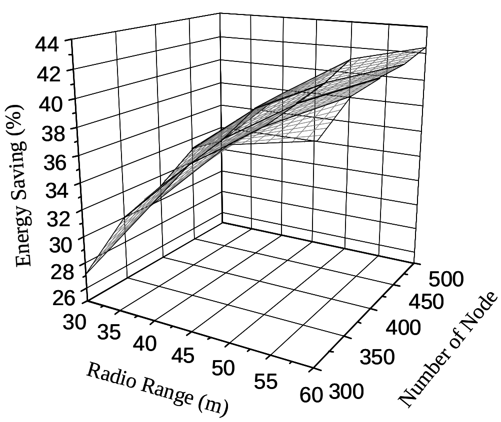
In the experiments, we randomly spread the homogeneous sensors in a $300 \times 300m^2$ sensor field area. All sensor nodes have the same fixed radio range and same energy. We use the DTS with both Short Rings topology and Long Rings topology for the initial routing tree structure. For measuring the energy consumption for transmitting, receiving, and overhearing data, we used the energy model based on [77].

6.5.1 Efficiency to Reduce Overhearing Energy

In one experiment, we measured the rate of energy saving comparing with S-MAC protocol. Even though S-MAC protocol already reduced the overhearing energy, the experiment result shows that OEWS and IWS reduces overhearing energy more than S-MAC. Fig. 6.8(a) and (b) show the energy saving results of OEWS comparing with S-MAC. We then increased the number of sensor nodes from 300 to 600 and the radio range from 30m to 60m. In Fig. 6.8(a), we compare the energy saving rate in total energy consumption including transmitting, receiving and overhearing. This result shows that OEWS reduces up to 1.7% more energy than S-MAC protocol. With the high density of sensor nodes, OEWS produces more saving of overhearing energy consumption. Fig. 6.8(b) shows the result where only compare the overhearing energy with S-MAC protocol. We see that OEWS can save the overhearing energy up to



(a)

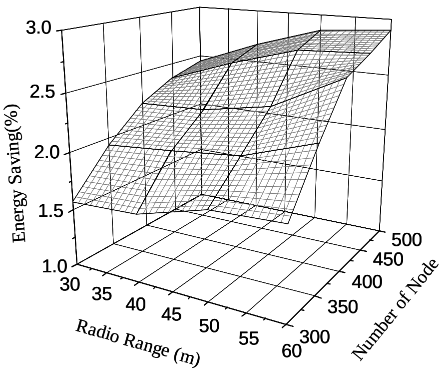


(b)

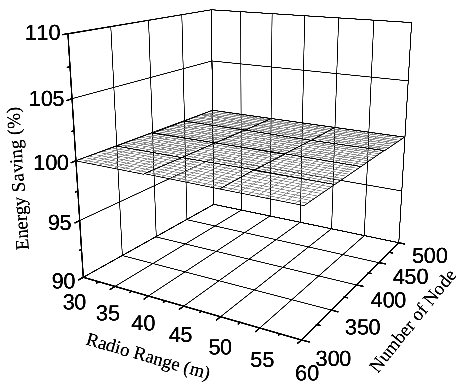
Figure 6.8: OEWS vs S-MAC energy saving (a) Total energy saving (b) Overhearing energy saving

43% more than S-MAC protocol. In this result, the more density of sensor nodes, the larger the decrease in overhearing energy consumption. Therefore, OEWS is more suitable for high density sensor networks than low density sensor networks.

Fig. 6.9 (a) and (b) show the result of energy saving rate in IWS. IWS saves more energy than OEWS comparing with S-MAC protocol. Because we remove the overhearing energy consumption, in Fig. 6.9 (b), the saving rate of overhearing energy is 100% comparing with S-MAC protocol's overhearing energy. In Fig. 6.9 (a), IWS reduces the energy up to 2.9%



(a)



(b)

Figure 6.9: IWS vs S-MAC energy saving (a) Total energy saving (b) Overhearing energy saving

more than S-MAC protocol. IWS also have better energy efficiency in higher density of sensor nodes.

6.5.2 Latency on OEWS, IWS and S-MAC

In this section, we analyze the data latency between OEWS, IWS and S-MAC protocol. In Fig. 6.2, for example, when a node in level 1 detects some events happening between the T period which is the duration of sleep, the node in level 1 waits until its next wakeup time. The probability of occurring event between T periods is uniformly distributed. Therefore we

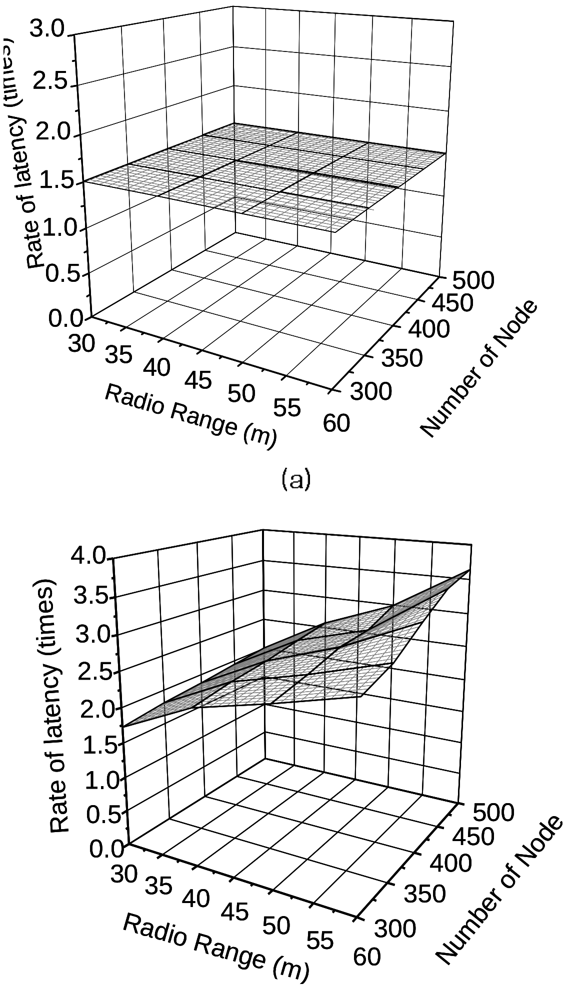


Figure 6.10: (a) Latency of OEWS (b) Latency of IWS

represent the uniform distribution between A and B as X U[A,B] based on [76]. X is random delay time. A and B are the smallest delay time and the largest delay time respectively. S-MAC has very similar performance to wakeup synchronized time where all the nodes in the sensor networks wake up and go to sleep mode at the same time with the same wakeup time. Hence delay time of S-MAC is represented by the following:

$$X \sim U[(h-1)T, hT] \tag{6.6}$$

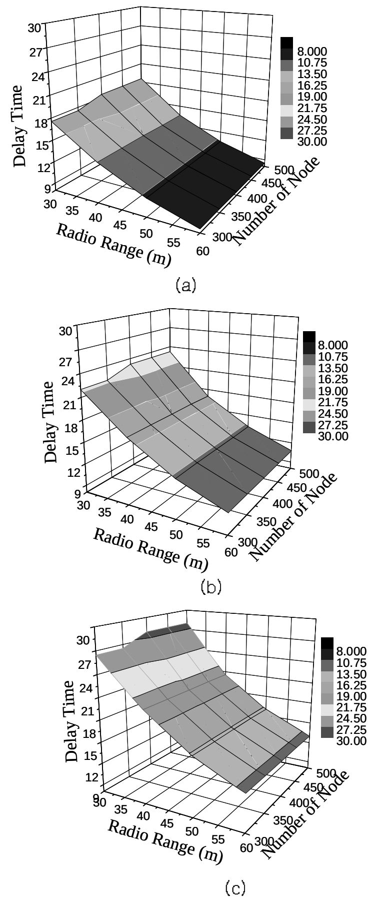


Figure 6.11: Data delay time (a) S-MAC (b) OEWS with DTS (c) OEWS without DTS

Therefore, average delay time is :

$$E(X) = (h - \frac{1}{2})T \tag{6.7}$$

In formulas (6.6) and (6.7), h means the number of hops. In OEWS, half of nodes of all sensor nodes wake up and then the other half wake up. Therefore a node in OEWS takes two times waiting time until the next wakeup time. Therefore, we can represent OEWS delay by the following:

$$X \sim U[(h-1)T, 2hT] \tag{6.8}$$

The average delay time of OEWS is the following:

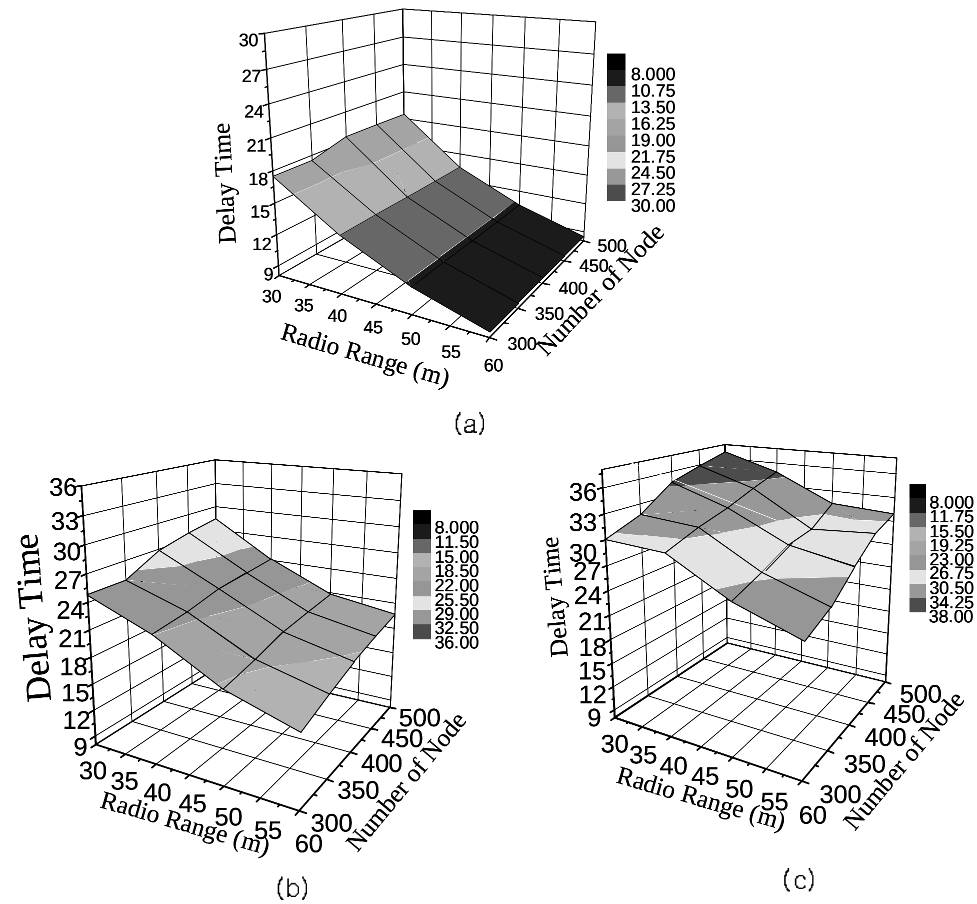


Figure 6.12: Data delay time (a) S-MAC (b) IWS with DTS (c) IWS without DTS

$$E(X) = (\frac{3}{2}h - \frac{1}{2})T \tag{6.9}$$

Fig. 6.10 (a) shows the result of a average latency in OEWS comparing with S-MAC. We simulated with number of nodes from 300 to 600 and radio range from 30 to 60m. In this environment, latency of OEWS is 1.51 times latency of S-MAC protocol. Hence this simulation shows that there is a trade-off between overhearing energy consumption and latency. Fig. 6.10 (b) shows the result of latency in IWS. In this case, as density of sensor field increases, latency is increased. With the same environment in Fig. 6.10 (b), latency of IWS is between 1.72 and 3.64 times more than latency of S-MAC protocol. In the next section, we show that DTS

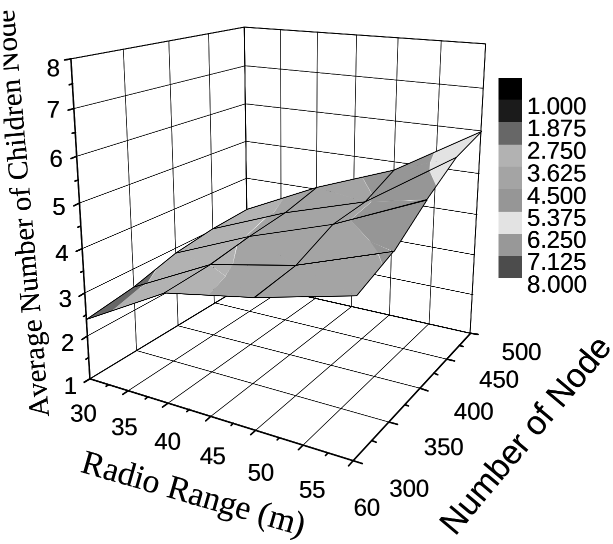


Figure 6.13: Average number of children nodes

which we proposed as a routing tree structure reduces the latency.

6.5.3 Effect on DTS in OEWS and IWS

We use the new routing tree structure called Double Tree Structures (DTS) to reduce the latency on OEWS and IWS. In this section, we compare latency between OEWS with DTS and OEWS without DTS. Fig. 6.11 (a) shows the delay time of S-MAC. Also, Fig. 6.11 (b) shows the delay time of OEWS with DTS and Fig. 6.11 (c) shows the delay time of OEWS without DTS. From these results, we know that DTS works to reduce the latency. DTS has Long Rings topology and Short Rings Topology. Therefore, there are two routing trees in the sensor network. If some nodes have alternative paths to the destination, we can reduce the waiting time. Hence average delay time of DTS with OEWS is represented by the following:

$$E(X) = \frac{(h - \frac{1}{2})T + (\frac{3}{2}h - \frac{1}{2})T}{2} \tag{6.10}$$

where, h is number of hops and T is the time duration of sleep mode. Also, we compare latency between OEWS with DTS and S-MAC. When we compare S-MAC delay time on Fig. 6.11

(b) with OEWS without DTS on Fig. 6.11 (c), even though OEWS reduce overhearing energy, OEWS has more delay time by about 50%. But when we used DTS on OEWS, OEWS with DTS has more delay time over S-MAC by about 13%. Therefore, we improved the trade-off between overhearing energy consumption and latency with proposed OEWS and DTS. In Fig. 6.12 (b) illustrates the delay time of IWS with DTS and Fig. 6.12 (c) shows the delay time of IWS without DTS. These results also show that DTS reduces the latency on IWS same as OEWS. Average delay time of IWS with DTS is represented by the following:

$$E(X) = \frac{3+N}{4}hT - \frac{1}{2}T \tag{6.11}$$

Where, N is average number of children nodes and h is number of hops. T is the time duration of sleep mode. In Fig. 6.13 shows the average number of children nodes on each level. As increased radio range and number of sensor nodes, the number of children nodes also increased.

6.6 Summary

In this paper, we have proposed OEWS and IWS for reducing the overhearing energy consumption with different wakeup times. Furthermore we also proposed DTS for decreasing the latency of OEWS and IWS with double tree structure. Even if advantage of OEWS and IWS is to reduce the overhearing energy consumption, there is a delay time because of a trade-of between energy saving and delay time. But DTS is useful to reduce the delay time. In DTS, Long Rings topology and Short Rings topology have the same number of hops from any node to the base station. Therefore using either Long Rings topology or Short Rings topology for routing path, the number of hop does not have effects on delay time.

Our simulation results also show OEWS and IWS with DTS have good performance. OEWS and IWS with DTS are more suitable for high density sensor network. Overhearing energy consumption is high when nodes are having many neighborhood sensor nodes.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusion of This Dissertation

Wireless sensor networks have different characteristic from existed wireless networks. Because of this reason, we need specific network protocol, routing structure, operating system, database, time synchronization, medium access control and so on. Recently, the focus of new mechanical design concentrates to extend the sensor network longevity. Advances in micro electro mechanical system have led to the long lifetime of wireless sensor network. However we need more development of new technology.

In this Thesis first, we discussed new multiple tree structure in wireless sensor network for energy load balance in in-network routing structure. We tried to decrease the network traffic concentrating on special sensor nodes. The concentrating traffic is one of the critical reasons to make unbalanced large amount of energy consumption in sensor networks. Specially, the nodes located near by base station make unbalanced energy consumption because network traffic has to pass through these nodes for going down to the low level sensor nodes in tree structure.

Second we proposed energy efficient data gathering. In many cases, sensor network use routing scheme based on the tree structure. However when we collect information from some specific area using tree routing structure, network traffic passes through the several tree branches. In this case we merged several routing path to remove unnecessary energy consumption for routing.

Third we proposed new wakeup scheme to reduce the overhearing energy consumption. Odd and even wakeup schedule and individual wakeup schedule are new wakeup scheme. In order to reduce overhearing energy, we use different wakeup time in children nodes. However, there is trade off between overhearing energy and latency. Therefore we proposed double tree structure for reducing the latency in OEWS or IWS. Double tree structure consists of two kinds of tree structure such as short rings topology and long rings topology. It is more suitable for high density sensor network.

7.2 Future Research Plan

In future works, first we are going to reduce more latency on odd and even wakeup scheme (OEWS) and individual wakeup scheme (IWS). Even though we decreased the latency on OEWS and IWS with double tree structure (DTS), still there is latency. Second we will investigate to extend OEWS and IWS to a new MAC protocol in sensor networks. Even if there are several energy efficient MAC protocols, they have still many challenges. Especially, we are going to focus on Energy Efficient MAC protocol in sensor networks. Energy efficiency is one of the primary goals in the sensor networks protocol design. Energy efficiency is very critical for real world sensor network applications. Therefore we will try to apply a new MAC protocol to real world applications such as a health application, environment monitoring application, and smart home application.

REFERENCES

[1] J. N. Al-karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, 2004.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, August 2008. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2008.04.002

[3] "http://xbow.com."

[4] M. Srivastava, "Power considerations for sensor networks," http://ipsn.acm.org/2001/slides/Srivastava.pdf, 2001.

[5] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor networks," *Mobile Computing Systems and Applications, IEEE Workshop on*, vol. 0, p. 49, 2002.

[6] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, February 2003. [Online]. Available: http://dx.doi.org/10.1109/TNET.2002.808417

[7] H. O. Tan and I. Körpeoğlu, "Power efficient data gathering and aggregation in wireless sensor networks," *SIGMOD Rec.*, vol. 32, no. 4, pp. 66–71, 2003.

[8] A. Manjhi, S. Nath, and P. B. Gibbons, "Tributaries and deltas: efficient and robust aggregation in sensor network streams," in *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2005, pp. 287–298.

[9] Wei, "An energy-efficient mac protocol for wireless sensor networks," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002.

[10] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.

[11] F. Zhao and L. Guibas, "Wireless sensor networks: An information processing approach," *Morgan Kaufmann*, 2004.

[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[13] A. Milenkovic, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Computer Communications*, vol. 29, no. 13-14, pp. 2521–33, 2006.

[14] J. S. Choi, B. Lee, K. Park, and R. Elmasri, "Robust tree-based in-network data processing for target tracking in wsns," in *International Conference on Parallel and Distributed Computing Systems (PDCS 2007)*, 2007.

[15] A. Ailamaki, C. Faloutsos, P. S. Fischbeck, M. J. Small, and J. VanBriesen, "An environmental sensor network to determine drinking water quality and security." *SIGMOD Record*, vol. 32, no. 4, pp. 47–52, 2003.

[16] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 46, no. 5, pp. 605–634, 2004.

[17] Q. Zhao, A. Swami, and L. Tong, "The interplay between signal processing and netwroking in sensor networks," *IEEE Signal Processing Magazine*, pp. 84–93, July 2006.

[18] C. R. Baker, K. Armijo, S. Belka, M. Benhabib, V. Bhargava, N. Burkhart, A. D. Minassians, G. Dervisoglu, L. Gutnik, M. B. Haick, C. Ho, M. Koplow, J. Mangold, S. Robinson, M. Rosa, M. Schwartz, C. Sims, H. Stoffregen, A. Waterbury, E. S. Leland, T. Pering, and P. K. Wright, "Wireless sensor networks for home health care," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 2, pp. 832–837, 2007.

[19] H. Lee, K. Park, B. Lee, J. Choi, and R. Elmasri, "Issues in data fusion for healthcare monitoring," in *to appear in PETRA 08*, 2008.

[20] U. Varshney, "Pervasive healthcare and wireless health monitoring," *Mob. Netw. Appl.*, vol. 12, no. 2-3, 2007.

[21] V. Shnayder, B. rong Chen, K. Lorincz, T. R. F. F. Jones, and M. Welsh, "Sensor networks for medical care," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2005, pp. 314–314.

[22] I. Vasilescu, C. Detweiler, and D. Rus, "Aquanodes: an underwater sensor network," in *WuWNet '07: Proceedings of the second workshop on Underwater networks*. New York, NY, USA: ACM, 2007, pp. 85–88.

[23] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.

[24] A. M. Mainwaring, D. E. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring." in *WSNA*, 2002, pp. 88–97.

[25] D. M. Doolin and N. Sitar, "Wireless sensors for wildfire monitoring," in *Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems.*, M. Tomizuka, Ed., may 2005, pp. 477–484.

[26] J. S. Choi, B. Lee, K. Park, and R. Elmasri, "Robust tree-based in-network data processing for target tracking in wsns," in *International Conference on Parallel and Distributed Computing Systems (PDCS 2007)*, 2007.

[27] M. A. Hammad, W. G. Aref, and A. K. Elmagarmid, "Stream window join: Tracking moving objects in sensor-network databases." in *SSDBM*, 2003, pp. 75–84.

[28] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system." in *INFOCOM*, 2000, pp. 775–784.

[29] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 88–97. [Online]. Available: http://dx.doi.org/10.1145/570738.570751

[30] R. Subramanian and F. Fekri, "Sleep scheduling and lifetime maximization in sensor networks: fundamental limits and optimal solutions," in *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*. New York, NY, USA: ACM, 2006, pp. 218–225.

[31] *Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach*, 2004. [Online]. Available: http://www.ece.arizona.edu/ younis/papers/infocom04.pdf

[32] A. Boukerche, X. Cheng, and J. Linus, "Energy-aware data-centric routing in microsensor networks," in *MSWIM '03: Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2003, pp. 42–49.

[33] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," 2002. [Online]. Available: http://citeseer.ist.psu.edu/schurgers02optimizing.html

[34] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and," Tech. Rep., 2001.

[35] T. Yuan and S. Zhang, "Secure fault tolerance in wireless sensor networks," *Computer and Information Technology, IEEE 8th International Conference on*, vol. 0, pp. 477–482, 2008.

[36] J. Dong, K. E. Ackermann, B. Bavar, and C. Nita-Rotaru, "Mitigating attacks against virtual coordinate based routing in wireless sensor networks," in *WISEC*, 2008, pp. 89–99.

[37] A. Perrig, J. Stankovic, D. Wagner, and C. Rosenblatt, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, pp. 53–57, 2004.

[38] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," *Parallel and Distributed Processing Symposium, International*, vol. 3, p. 30186b, 2001.

[39] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: An acquisitional query processing system for sensor networks," *ACM Trans. Database Syst*, vol. 30, p. 2005, 2005.

[40] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," 2005, pp. 115–148. [Online]. Available: http://dx.doi.org/10.1007/3-540-27139-2_7

[41] I. Ahmed, M. Peng, and W. Wang, "Energy efficient cooperative nodes selection in wireless sensor networks," *Parallel Processing Workshops, International Conference on*, vol. 0, p. 50, 2007.

[42] X. Dai, F. Xia, Z. Wang, and Y. Sun, "An energy-efficient in-network aggregation query algorithm for wireless sensor networks," *Innovative Computing ,Information and Control, International Conference on*, vol. 3, pp. 255–258, 2006.

[43] M. A. Rahman and S. Hussain, "Energy efficient query processing in wireless sensor network," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 2, pp. 696–700, 2007.

[44] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “Tag: A tiny aggregation service for ad-hoc sensor networks.” in *OSDI*, 2002.

[45] N. Sadagopan, B. Krishnamachari, and A. Helmy, “The acquire mechanism for efficient querying in sensor networks,” in *In IEEE International Workshop on Sensor Network Protocols and Applications (SNPA03*, 2003, pp. 149–155.

[46] R. C. Shah and J. M. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” 2002.

[47] D. Braginsky and D. Estrin, “Rumor routing algorithm for sensor networks,” in *In Proceedings of the First Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, September 28 2002. [Online]. Available: 077

[48] K. Joanna, H. Wendi, and B. Hari, “Negotiation-based protocols for disseminating information in wireless sensor networks,” in *Wirel. Netw.*, 2007, pp. 169–185.

[49] J. Kulik, W. Heinzelman, and H. Balakrishnan, “Negotiation-based protocols for disseminating information in wireless sensor networks,” *Wirel. Netw.*, vol. 8, no. 2/3, pp. 169–185, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1013715909417

[50] C. fabiana Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, “Energy efficient design of wireless ad hoc networks,” in *In Proceedings of European Wireless*. Springer, 2002.

[51] S. Banerjee and S. Khuller, “A clustering scheme for hierarchical control in wireless networks,” in *In Proceedings of IEEE INFOCOM 2001*, 2001, pp. 1028–1037.

[52] V. Kawadia and P. R. Kumar, “Power control and clustering in ad hoc networks,” in *In INFOCOM*, 2003, pp. 459–469.

[53] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” 2003.

[54] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks.” in *HICSS*, 2000.

[55] Y. Yao and J. Gehrke, “The cougar approach to in-network query processing in sensor networks,” *SIGMOD Record*, vol. 31, p. 2002, 2002.

[56] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *SenSys ’03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2003, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1145/958491.958493

[57] A. Keshavarzian, H. Lee, and L. Venkatraman, “Wakeup scheduling in wireless sensor networks,” in *MobiHoc*, 2006, pp. 322–333.

[58] Q. Wang and Z. J. Haas, “Bass: an adaptive sleeping scheme for wireless sensor network with bursty arrival,” in *IWCMC*, 2006, pp. 945–950.

[59] M. Stemm and R. H. Katz, “Measuring and reducing energy consumption of network interfaces in hand-held devices,” *IEICE Transactions on Communications*, vol. E80-B, no. 8, pp. 1125–31, 1997. [Online]. Available: http://citeseer.ist.psu.edu/stemm97measuring.html

[60] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proc. 1st ACM conference on Embedded networked sensor systems (SenSys)*, November 2003.

[61] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004. [Online]. Available: http://dx.doi.org/10.1109/MWC.2004.1368893

[62] V. Shnayder, M. Hempstead, B. rong Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *In Sensys*. ACM Press, 2004, pp. 188–200.

[63] A. L. P. K. C. Mohamed A. Sharaf, Jonathan Beaver, “Balancing energy efficiency and quality of aggregate data in sensor networks,” pp. 13(4): 384 – 403, December 2004.

[64] A. Manjeshwar and D. P. Agrawal, “Teen: A routing protocol for enhanced efficiency in wireless sensor networks,” *Parallel and Distributed Processing Symposium, International*, vol. 3, p. 30189a, 2001.

[65] Q. Li, J. Aslam, and D. Rus, “Hierarchical power-aware routing in sensor networks,” in *In Proceedings of the DIMACS Workshop on Pervasive Networking*, 2001.

[66] C.-Y. Wen and W. A. Sethares, “Automatic decentralized clustering for wireless sensor networks,” *EURASIP J. Wirel. Commun. Netw.*, vol. 5, no. 5, pp. 686–697, October 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1115494

[67] A. Deshp, E. C. Guestrin, and S. R. Madden, “Model-driven data acquisition in sensor networks,” in *In VLDB*, 2004, pp. 588–599.

[68] M. Younis, M. Youssef, and K. Arisha, “Energy-aware routing in cluster-based sensor networks,” *Modeling, Analysis, and Simulation of Computer Systems, International Symposium on*, vol. 0, p. 0129, 2002.

[69] W. Choi, P. Shah, and S. K. Das, “A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks,” in *Proc. of the First Annual International Conference on Mobile and Ubiquitous Systems: MobiQuitous ’04*, 2004, pp. 203–212.

[70] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” in *INFOCOM*, 2003.

[71] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0262531968

[72] H. Yu, P. B. Gibbons, S. Seshan, S. Nath, and S. Nath, “Synopsis diffusion for robust aggregation in sensor networks,” in *In SenSys*. ACM Press, 2004, pp. 250–262.

[73] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The platforms enabling wireless sensor networks." *Commun. ACM*, vol. 47, no. 6, pp. 41–46, 2004.

[74] A. Silberstein, R. Braynard, and J. Yang, "Constraint chaining: on energy-efficient continuous monitoring in sensor networks." in *SIGMOD Conference*, 2006, pp. 157–168.

[75] A. L. P. K. C. Mohamed A. Sharaf, Jonathan Beaver, "Balancing energy efficiency and quality of aggregate data in sensor networks," pp. 13(4): 384 – 403, December 2004.

[76] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks," in *MobiHoc*, 2006, pp. 322–333.

[77] P. Basu and J. Redi, "Effect of overhearing transmissions on energy efficiency in dense sensor networks," in *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*. New York, NY, USA: ACM, 2004, pp. 196–204.

[78] H. M. Ammari and S. K. Das, "Trade-off between energy savings and source-to-sink delay in data dissemination for wireless sensor networks," in *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2005, pp. 126–133.

BIOGRAPHICAL STATEMENT

Byoungyong Lee was born in Seoul, Korea. He received his Bachelor of Science degree in Computer Science from Kangnam University, Korea, in 1996 and his Master of Science degrees in Computer Science from Soongsil Univerity, Korea, in 1998. His Ph.D. degrees from The University of Texas at Arlington in 2009 in Computer Science Department. His current research interest is in the area of wireless sensor networks and network security.