Spring 5-31-2012

Heterogeneity-aware and energy-aware scheduling and routing in wireless sensor networks

Mahesh Kumar Vasanthu Somashekar
New Jersey Institute of Technology

ABSTRACT

HETEROGENEITY-AWARE AND ENERGY-AWARE SCHEDULING AND ROUTING IN WIRELESS SENSOR NETWORKS

by
Mahesh Kumar Vasanthu Somashekar

A Wireless Sensor Network (WSN) is a group of specialized transducers, called sensor nodes, with a communication infrastructure intended to monitor and record conditions at diverse locations. Since WSN applications are usually deployed in an open environment, the network is exposed to rough weather conditions, such as rain and snow. Another problem that WSN applications need to deal with is the energy constraints of sensor nodes. Both problems adversely affect the lifetime of WSN applications. A lot of research has been conducted to prolong the lifetime of WSN applications considering energy constraints of sensor nodes, but not much research has gone into tackling both the environmental effects and energy constraints. The goal of this research is to efficiently deal with these two problems and provide a solution for scheduling and routing in a heterogeneous sensor network.

The research has been divided into two phases - Scheduling and Routing. In the scheduling phase, only some sensor nodes are scheduled to run for a particular timeslot and during that timeslot other sensor nodes are kept in sleep mode. A set of sensor nodes for a timeslot is chosen based on their positional information. In the routing phase, a least cost route from a sensor to the sink is dynamically determined to prolong the lifetime of the sensor network.

HETEROGENEITY-AWARE AND ENERGY-AWARE SCHEDULING AND ROUTING IN WIRELESS SENSOR NETWORKS

by
Mahesh Kumar Vasanthu Somashekar

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

Department of Computer Science

May 2012

Blank Page

APPROVAL PAGE

HETEROGENEITY-AWARE AND ENERGY-AWARE SCHEDULING AND ROUTING IN WIRELESS SENSOR NETWORKS

Mahesh Kumar Vasanthu Somashekar

Dr. Guiling Wang, Thesis Advisor — Date
Associate Professor of Computer Science, NJIT

Dr. Vincent Oria, Committee Member — Date
Associate Professor of Computer Science, NJIT

Dr. Usman W Roshan, Committee Member — Date
Associate Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Mahesh Kumar Vasanthu Somashekar

Degree: Master of Science

Date: May 2012

Undergraduate and Graduate Education:

- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2012
- Bachelor of Engineering in Computer Science & Engineering,
  National Institute of Engineering, Karnataka, India, 2007

Major: Computer Science

For my family,
this triumph is as much theirs as it is my own.

ACKNOWLEDGMENT

First of all I would like to express my gratitude to my advisor, Dr. Guiling Wang, for providing me with an opportunity to work under her supervision. Dr. Wang has been extremely helpful and encouraging during this research work. It has been a wonderful experience working under her and I greatly thank her for that.

I would like to thank my committee members: Dr. Vincent Oria & Dr. Usman Roshan for taking time out from their busy schedules to review the research work. I would also like to thank the distinguished faculty members and administrative staff of the Department of Computer Science for its support.

I would also like to thank Mr. Jie Tian and Mr. Tan Yan for their support throughout this period of research and development. Special thanks to Mr. Jie Tian for his guidance during each and every phase of this work. The invaluable guidance of both, Dr. Guiling Wang & Mr. Jie Tian, was integral to the successful culmination of the research work.

Last, but not the least, I would like to thank my family and friends for their love and support. I am completely indebted to them for all that I have accomplished and will accomplish in the future.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

A wireless sensor is a device which monitors and collects useful data from its surroundings. Sensor nodes communicate each other via wireless channels to transmit data. Sensor nodes are usually battery driven. A Wireless Sensor Network (WSN) is a network infrastructure consisting of a group of wireless sensor nodes. In WSN applications, sensor collected data is transmitted to a common sink node or base station. Sink node processes the collected data to yield productive results. A simple WSN is shown in Figure 1.1.

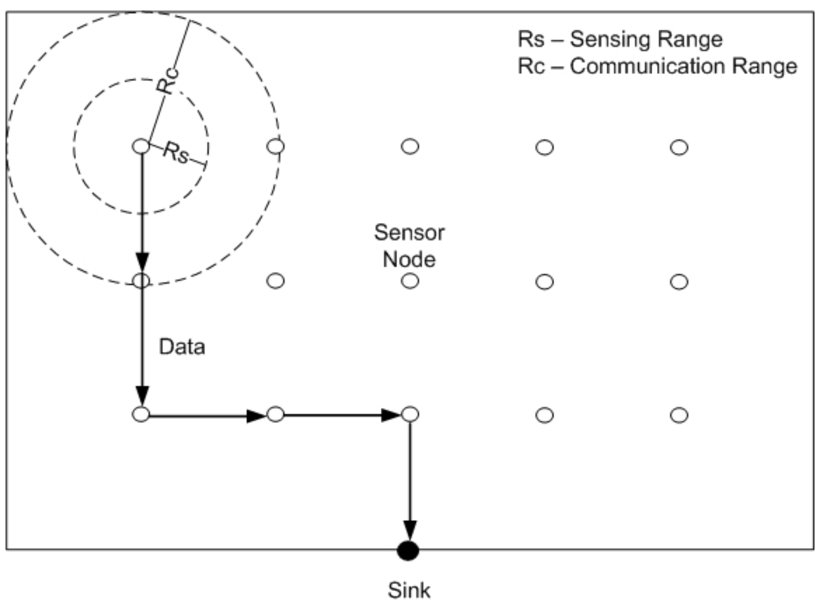

Figure 1.1 Wireless Sensor Network.

WSN technology has significantly impacted the efficiency of many military and civil applications such as military surveillance, structural monitoring and robot control. Sensor nodes monitor and gather data of the area and transmit this data to the remote system for analysis. For example, in an air pollution management system, the sensor nodes are deployed in the area of interest. Sensor nodes monitor the concentration of hazardous gases in the area and control the air quality as instructed. WSN is more flexible than other typical air pollution management systems as they are easy to deploy, less expensive, scalable and efficient.

Sensor nodes are constrained by energy, storage capacity and bandwidth. These are the sensor-level challenges that affect the operation of the sensor network. Growing interests in the field of wireless communication has triggered the evolution of WSN technologies. Significant research has been taken undertaken to overcome the challenges of the sensor in a sensor network. This research work is to analyze and design scheduling and routing algorithm for the WSN which will efficiently handle the environmental and energy constraint problems. Heterogeneity in this research indicates the heterogeneity in the type of sensor nodes used and their behavior to the environmental forces.

1.2 Problem Statement

As many WSN applications are deployed in an open environment, the network is subject to the forces of the environment. Environmental forces refers to rain, extreme temperature, the possibility of an animal stepping on a sensor node and other factors. Energy is another constraint of the sensor node as these nodes run on battery. Both problems adversely affect the lifetime of the WSN applications.

Environmental forces can have a severe impact on sensor networks. If the area of deployment is windy then there is a high probability of the sensor node getting damaged physically. If the area has a high prediction of snow storm then during that period only a few of sensor nodes survive depending on their specifications. It is also very important and difficult to express the effects of these environmental forces on the sensor network. A sensor node's behavior changes under different weather conditions. A sensor node might temporarily or permanently stop working depending upon the impact of these forces on the sensor node.

Energy is another important factor that affects the functionality of the WSN application. Sensor nodes have limited battery charge and their durability depends on how fast the battery drains. Sources of energy consumption in sensor nodes are data transmission, collisions, idle listening and others. Impacts of sensor energy constraints on the sensor network are network partition, periodic decrement in coverage area and improper functioning of the application.

Environmental forces can also affect the energy consumption of a sensor node. During some environmental conditions, some sensor nodes stop operating causing the surviving sensor nodes to take more load and/or the surviving nodes require transmitting data to longer distance consuming more power. Proper handling of these problems will drastically increase the performance and lifetime of the sensor network.

CHAPTER 2

ASSUMPTIONS

A series of realistic assumptions have been made about Wireless Sensor Networks in the research. WSNs have a wide range of applications and these assumptions would suit to most of the applications.

Assumptions made on WSN are as follows:

- Environmental conditions are among the prime parameters considered in this research. Environmental conditions considered include sunny and rainy weather. The assumption is that, during rain, only robust sensor nodes sustain the weather and they continue to function normally, whereas, regular sensor nodes stop functioning. As the weather changes from rainy to sunny, a few of the regular sensor nodes might not come back up depending on the intensity of rain. During sunny weather, both regular and robust sensor nodes run normally.
- The weather forecast system considered is hourly. Each sensor node will be knowledgable of the weather forecast of 5-10 hours from the current time.
- One of the main technical assumptions that this research uses is that the sink node will have a very dense distribution of immediate neighbor sensor nodes. This plays a vital role in evenly assigning set numbers to the sensor nodes and, thereby, provides consistent coverage by all set of sensor nodes.
- All sensor nodes have data to transmit during each timeslot.
- Wireless links between sensor nodes are bi-directional and the radio channel for communication is symmetric. This means that the energy required to transmit data from sensor-A to sensor-B and from sensor-B to sensor-A is the same.
- All sensor nodes are static and they know their location details either by using a GPS or by other localization methods [12].
- All sensor nodes are loosely time synchronized [7].
- Sink nodes have no energy constraints unlike other regular/robust sensor nodes.

CHAPTER 3

DESIGN

In Wireless Sensor Network, sequence of data transmission is dynamic because of its ad-hoc nature. The goal of the research is to develop a methodology for such ad-hoc networks to increase the lifetime of network. Some of the prime factors that the research needs to handle are dynamically adjusting to the changes in the weather condition, dynamically determining the surviving sensor nodes, maintaining network connectivity and handling other factors. The research work is a combination of scheduling and routing. Scheduling is used to schedule a set of sensor nodes to be in on-duty mode during some particular timeslots and to be in sleep mode during other timeslots. Sensor nodes in sleep mode will be in-directly contributing in preserving energy of the WSN application. Routing is used to find an intelligent route from a sensor node to the sink node considering energy level of sensor nodes and environmental condition. Efficient routing make sure that the energy consumption of the sensor nodes are evenly distributed. Scheduling and routing performing together greatly increase the lifetime of the WSN application. Section 3.1 and Section 3.2 will be explaining in-detail on scheduling and routing design respectively.

3.1 Scheduling Design

The purpose of scheduling in WSN is to schedule a set of sensor nodes to run in some particular timeslots. In each timeslot, only a set of sensor nodes will be in on-duty mode and other sensor nodes will be in sleep mode. In order to achieve this, first thing is to create disjoint set of sensor nodes and then schedule each set of sensor nodes in particular timeslots. In the research, creation of set of sensor nodes is performed hop-level in order that each set of sensor nodes can maintain network connectivity and cover the area of interest as much as possible.

3.1.1 Determine Minimum Hop Count to Sink

First operation is to determine minimum hop count to sink of all the sensor nodes. This initial step is very important as hop count will be used for creating sets and it also plays a vital role during routing phase. Figure 3.1 shows the pictorial representation of hop count to sink of all the sensor nodes.

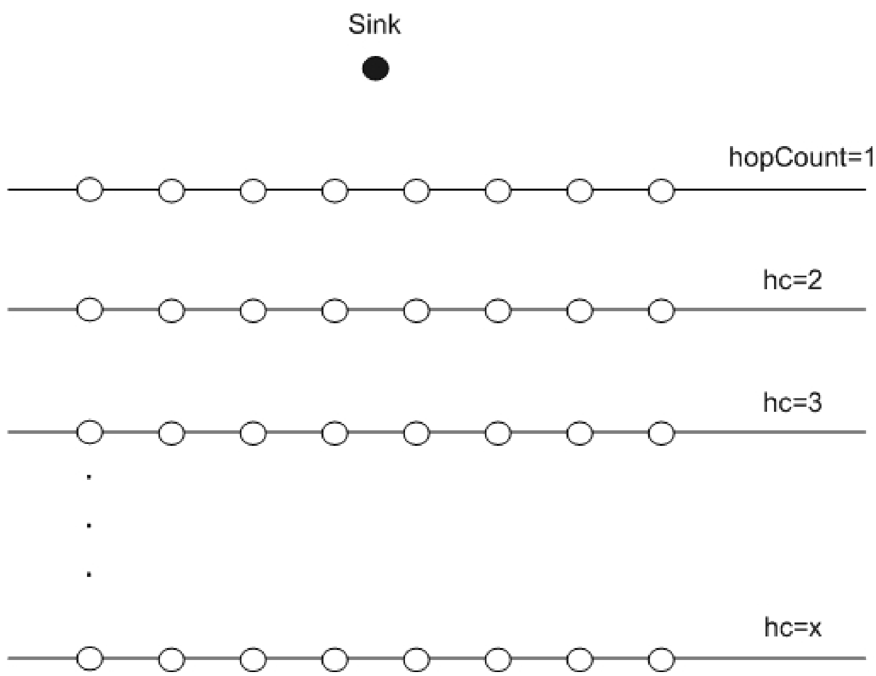

Figure 3.1 Minimum Hop Count of Sensors to Sink.

Steps to determine minimum hop count to sink are as follows:

1. The process is initiated by the sink. Sink initializes a packet with hop_count=0 and broadcasts that hop information to its immediate neighbor sensor nodes.
2. Neighbor sensor node upon receiving hop_count information, increments the hop_count, stored in the packet, by 1.
3. Then sensor node checks whether this incremented packet hop_count value is lower than its internally stored hop_count, if it has already assigned hop_count.

4. If incremented packet hop_count value is lower or the sensor is receiving hop_count packet for the first time, then it stores that hop_count value and broadcasts incremented hop_count packet to its neighbor sensor nodes. This goes to Step-2. The process is repeated till all the sensor nodes in the network are assigned with minimum hop_count value.

5. If incremented hop_count is higher or equal to its stored hop_count, then the node will keep its current hop_count information.

3.1.2 Create Sets

The main idea behind creating set of sensor nodes in WSN application [4] is that the number of sensor nodes deployed is usually higher than the deterministic number of sensor nodes required and consequently, the coverage of most of the sensor nodes will be overlapping. Considering the overall picture of the network area, not all part of area will be covered by more than one sensor. Therefore, the goal in creating sets is that sensor nodes running in each set should provide consistent coverage of the area.

Before creating sets, it is required to determine the number of sets that can be created. Number of sets depends on the total number of sensor nodes and the area of the field. After determining the number of sets, next step is to assign set number to all the sensor nodes in the network. While creating sets, it is very important to make sure that each set of sensor nodes working independently will not cause network partitioning.

Creating sets is performed at hop-level. First, all 1-hop sensor nodes are assigned to set number, second, all 2-hop sensor nodes are assigned and so on. This is performed in two phases. The first phase is to assign set number to all 1-hop sensor nodes. The second phase is to assign set number to other hop sensor nodes considering the previous hop sensor node's set numbering. These two phases are explained in Section 3.1.2.1 and Section 3.1.2.2 respectively.

3.1.2.1 Set Numbering for 1-Hop Sensor Nodes

As per one of the assumptions, the sink will have very dense distribution of neighbor sensor nodes. This assumption is helpful in evenly assigning set numbers to the sensor nodes in 1-hop as well as in all subsequent hops.

Algorithm 1 Set Number Calculation for 1-Hop Nodes

```
Notations:
1hopNodes[]: all 1-hop nodes and their position details
distances[]: the distances between all 1-hop nodes and sink
numOf1HopNodes: number of 1-hop nodes
numOfSets: number of sets
determineDistance(): method to determine the distance between
all 1-hop nodes and sink
sortInDescending(): method to sort in decending order

distances[]=determineDistance(1hopNodes, sink)
distances[]=sortInDescending(distances)
for i = 1 → numOf1HopNodes do
  if 1hopNodes[i].set==0 then
    1hopNodes[i].set=1
    for j = 2 → numOfSets do
      minCost = infinite
      for k = 1 → numOf1HopNodes do
        if i!=k and 1hopNodes[k].set==0 then
          tmpDist = getDistance(1hopNodes[i], 1hopNodes[k])
          if tmpDist < minCost then
            minCost=tmpDist
            minNode =k
          end if
        end if
      end for
      if minCost != infinite then
        1hopNodes[minNode].set = j
      end if
    end for
  end if
end for
```

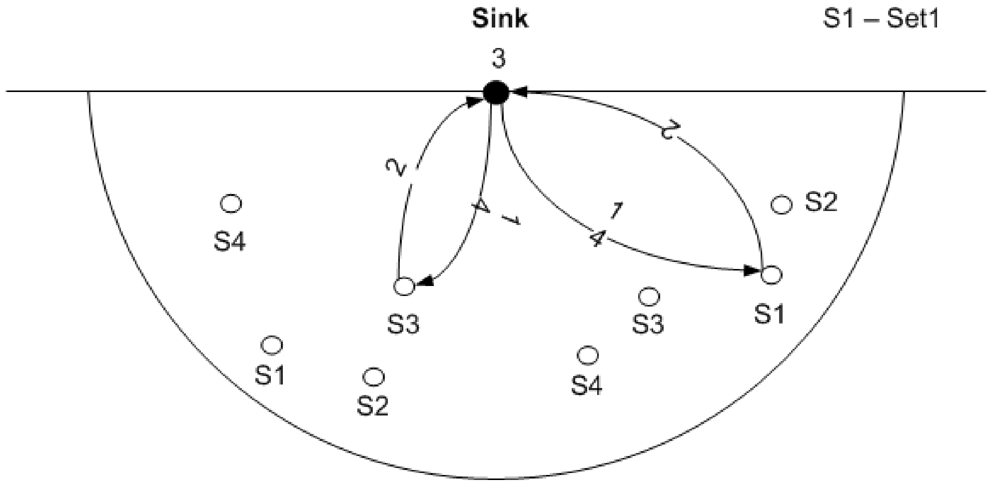

Figure 3.2 Steps for Assigning Set Numbers to 1-Hop Sensor Nodes.

Steps for assigning set numbers to 1-hop sensor nodes are as follows (depicted in Figure 3.2):

1. Sink requests all its immediate neighbor sensor nodes to provide their position details.

2. All neighbor sensor nodes upon receiving this request, transmits their position details to the sink.

3. Sink after receiving position details from all its neighbors, it intelligently determines set number using position details and assigns set number to all neighbor sensor nodes. Determining set numbers is described as below and algorithm for determining it is given in Algorithm 1.

   (a) Sink calculates distance from itself to all its neighbor sensor nodes using their position details.

   (b) It iterates through all its neighbors in descending order of their distance.

   (c) Assign further-most unassigned sink neighbor to Set-1. Then loop through other set numbers, i.e. set numbers 2, 3 & so on, and try to assign these set numbers to other unassigned neighbor sensor nodes. Sink performs all these calculations to assign set number to 1-hop sensor nodes based on their position details.

   (d) After assigning a neighbor to Set-1, determine the unassigned sink neighbor nodes which are close to that sensor node. Assign those sink neighbor nodes to the consequent set numbers, i.e. Set-2, Set-3 and so on. This means, assign a sensor node to Set-1, then find sensor nodes close to that sensor node and assign those sensor nodes to consequent set numbers. This is shown in Figure 3.3.

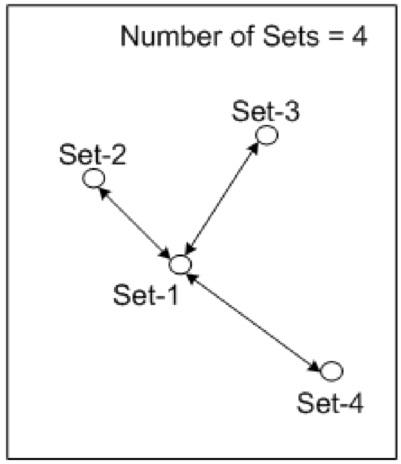

Figure 3.3 Determining Set Numbers for 1-Hop Sensor Nodes.

(e) Iterate till all sink neighbors are assigned to a set.

4. Then sink broadcasts calculated set number to all its neighbor sensor nodes.

3.1.2.2 Set Numbering for x-Hop Sensor Nodes, where x>1, based on (x-1)-Hop Sensor Nodes

Once all 1-hop sensor nodes are assigned with a set number, some of 1-hop sensor nodes inform 2-hop sensor nodes to start the process of determining and assigning set numbers to that hop sensor nodes. Once 2-hop sensor nodes are processed, it will transmit control to its next hop sensor nodes and it continues till all the sensor nodes in the network are assigned with set number.

Steps for assigning set numbers to x-hop sensor nodes, where x>1, are as follows (depicted in Figure 3.4):

1. After (x-1)-hop sensor nodes complete their processing, some of (x-1)-hop sensor nodes communicate x-hop sensor nodes to start set numbering process.

2. On receiving this message, a sensor node initiates by requesting its same-hop immediate neighbor sensor nodes to send their possible set assigning numbers. Possible set numbers of a sensor node is the set numbers of all its previous hop neighbor sensor nodes.

3. Neighbor sensor nodes broadcast their possible set assigning numbers.

Algorithm 2 Set Number Calculation for x-Hop Nodes, where x>1

```
Notations:
neighborDetails[]: same hop neighbor nodes details
counter[]: counter of set numbers assigned to nodes
frequency[]: frequency of possible set numbers of all unassigned nodes
numOfNeighbors: number of neighbors of process initiated node
1PossSetNeighbors: number of neighbors having only one possible set
minCountSet: set having minimum counter
maxFrequencySet: set having maximum frequency
matchCount: matching nodes count
assignSets(): calculates and assigns set numbers to nodes using
counter and frequency
getMinSet(): method to get mininum set number based on counter
getMaxSet(): method to get maximum set number based on frequency
findMatchingNode(): method to find matching node using matchcount,
mincounterset and maxfrequencyset

counter[]=updateCounter(neighborDetails)
frequency[]=updateFrequency(neighborDetails, Except Neighbors
                                   Having One Possible Set)
assignSets()
counter[]=updateCounter(neighborDetails)
frequency[]=updateFrequency(neighborDetails, Include Neighbors
                                   Having One Possible Set)
assignSets()

proc assignSets()
for i = 1 → (numOfNeighbors − 1PossSetNeighbors) do
  minCountSet = getMinSet(counter)
  maxFrequencySet = getMaxSet(frequency, ExceptMinCountSet)
  for matchCount = 2 → numOfSets do
    node = findMatchingNode(matchCount, minCountSet,
                                  maxFrequencySet)
    node.setNum = minCountSet
    counter[]=updateCounter(node)
    frequency[]=updateFrequency(node)
  end for
end for
end proc
```

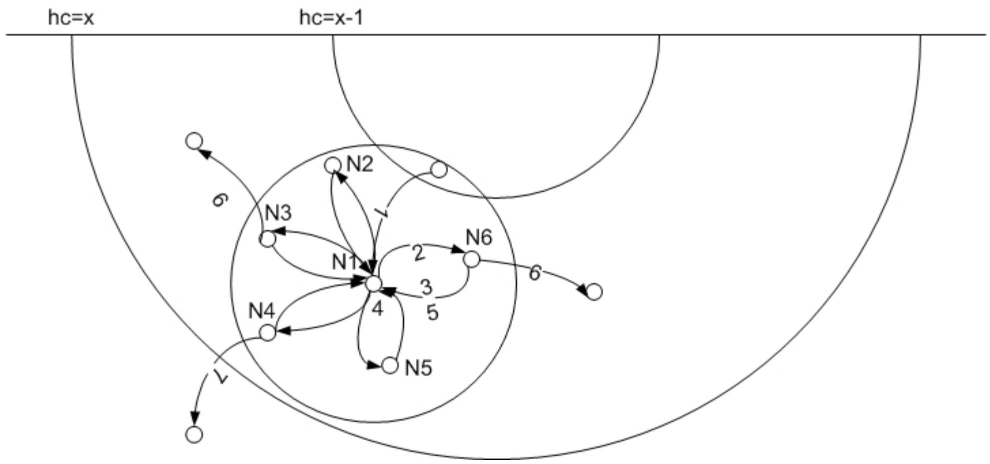

Figure 3.4 Steps for Assigning Set Numbers to x-Hop Sensor nodes.

4. After receiving possible set details from all the neighbor sensor nodes, the current sensor node determines the set numbering for all its neighbor sensor nodes. Determining set numbers is explained as below and detailed algorithm is provided in Algorithm 2. Example for this is shown in Example 1.

   (a) Node calculates counter and frequency using received neighbor sensor nodes details. Frequency is calculated using all neighbor sensor nodes, except for neighbor sensor nodes having only one possible set. Counter is the count of set numbers assigned to all neighbor sensor nodes and frequency is the number of possible set numbers of all unassigned neighbor sensor nodes.

   (b) Using counter and frequency, set numbers are calculated for all neighbor sensor nodes, except for one possible set neighbor sensor nodes.

   (c) Counter and frequency are recalculated including neighbor sensor nodes having one possible set.

   (d) Set numbers for one possible set neighbors are calculated in the same procedure using updated counter and frequency.

5. Then it broadcasts determined set numbers to all its neighbor sensor nodes.

6. Some of these assigned sensor nodes, say sensor nodes assigned to set-1, then communicates to unassigned sensor nodes of same hop to start the assigning process.

7. Some sensor nodes communicate with next hop sensor nodes to start their process.

Example 1 Set Number Calculation

Step-1: Counter and Frequency calculation without considering nodes having only one possible set

Initializing sensor node receives possible set details from all the neighbor sensor nodes. If a neighbor sensor node is already assigned to a set number, it will directly send its set number. In below table, Node-1 has only one possible set and Node-4 is assigned to Set-3. Initializing sensor node stores all these details as shown in table:

$Node$	$Possible\ Sets$
$Node-1$	$Set-\{1\}$
$Node-2$	$Set-\{2\|3\}$
$Node-3$	$Set-\{1\|3\}$
$Node-4$	$Set-3$
$Node-5$	$Set-\{1\|2\|3\}$
$Node-6$	$Set-\{2\|3\}$

Initializing sensor node initializes counter and frequency on the possible sets of all neighbor sensor nodes, except for one set possibility sensor nodes. Counter is the count of set numbers assigned to all neighbor sensor nodes and frequency is the number of possible set numbers of all unassigned neighbor sensor nodes.
Using the counter and frequency tables, set number is assigned iteratively to all unassigned sensor nodes. In each iteration, counter and frequency tables are updated. This calculation is described in-detail in Algorithm 2.

Counter calculation table:

Set	$Initially$	$Iteration-1$	$Iteration-2$	$Iteration-3$	$Final$
$Set-1$	$\underline{0}$	1	$\underline{1}$	2	2
$Set-2$	0	$\underline{0}$	1	$\underline{1}$	2
$Set-3$	1	1	1	1	1

Frequency calculation table:

Set	$Initially$	$Iteration-1$	$Iteration-2$	$Iteration-3$
$Set-1$	$\underline{2}$	1	$\underline{1}$	0
$Set-2$	3	$\underline{3}$	$\underline{2}$	$\underline{1}$
$Set-3$	$\underline{4}$	$\underline{3}$	$\underline{2}$	$\underline{1}$
	$N-3=S-1$	$N-2=S-2$	$N-5=S-1$	$N-6=S-2$

Step-2: Counter and Frequency calculation considering all nodes

Now, except for sensor nodes having only one possible set, all other sensor nodes are assigned to set number. For sensor nodes having only one possible set, possibility of assigning to set numbers is again calculated using even the sensor nodes which got assigned to set number in Step-1. In Step-1, Node-1 was having only one possible set (i.e. Set-1), after recalculation, Node-1 is now having Set-1—2—3 possible sets.

Recalculated Node and its Possible Sets table:

$Node$	$PossibleSets$
$Node-1$	$Set-\{1\|2\|3\}$
$Node-2$	$Set-2$
$Node-3$	$Set-1$
$Node-4$	$Set-3$
$Node-5$	$Set-1$
$Node-6$	$Set-2$

Counter calculation table:

Set	$Initially/Iteration-1$	$Final$
$Set-1$	2	2
$Set-2$	2	2
$Set-3$	$\underline{1}$	2

Frequency calculation table:

Set	$Initially/Iteration-1$
$Set-1$	$\underline{1}$
$Set-2$	$\underline{1}$
$Set-3$	$\underline{1}$
	$N-1=S-3$

Final: N-1=S-3, N-2=S-2, N-3=S-1, N-4=S-3, N-5=S-1, N-6=S-2

3.1.3 Scheduling

After sensor nodes are divided and assigned to sets, each set of sensor nodes are scheduled to run in particular timeslots. Scheduling sequence of this set of sensor nodes is a sequential sequence, such as set-1, set-2, set-3,...set-k, set-1, set-2 and so on. Example of scheduling sequence of set of sensor nodes is shown in Figure 3.5.

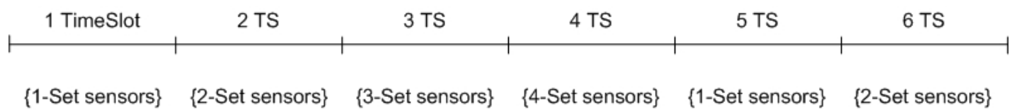

Figure 3.5 Scheduling Sequence of Set of Sensor Nodes, When Number of Set is Four.

3.2 Routing Design

Routing in WSN is the process of selecting route from a sensor node to sink to transmit data. While finding the route, algorithm is required to consider heterogeneity of sensor nodes, weather condition and energy-level of sensor nodes. Routing algorithm's moto would be to efficiently utilize the energy of the regular and robust sensor nodes considering all parameters. During rain, only robust sensor nodes work, as a result energy of robust sensor nodes will be drained more. Keeping all these into account, during sunny weather, it is best to over-utilize energy of regular sensor nodes. This routing idea, in combination with the sets concept of scheduling, will greatly influence in conserving the energy of sensor nodes and, thereby, increasing the lifetime of the WSN application.

3.2.1 Hop-by-Hop Routing Decisions

Based on the scheduling, in each timeslot only one set of sensor nodes will be in on-duty mode. In each timeslot, running sensor nodes have to take decisions to choose the best possible route to transmit data. These routing decisions are taken at hop-level to transmit data to previous hop sensor node, i.e., sensor node at x-hop takes decision to transmit data to (x-1)-hop sensor node and it continues till it reaches sink. Example of this is shown in Figure 3.6.

Routing uses the Energy Difference model, explained in Section 3.2.3, to determine best possible previous hop sensor node. The Energy Difference prediction algorithm calculates the energy difference between the current residual energy and

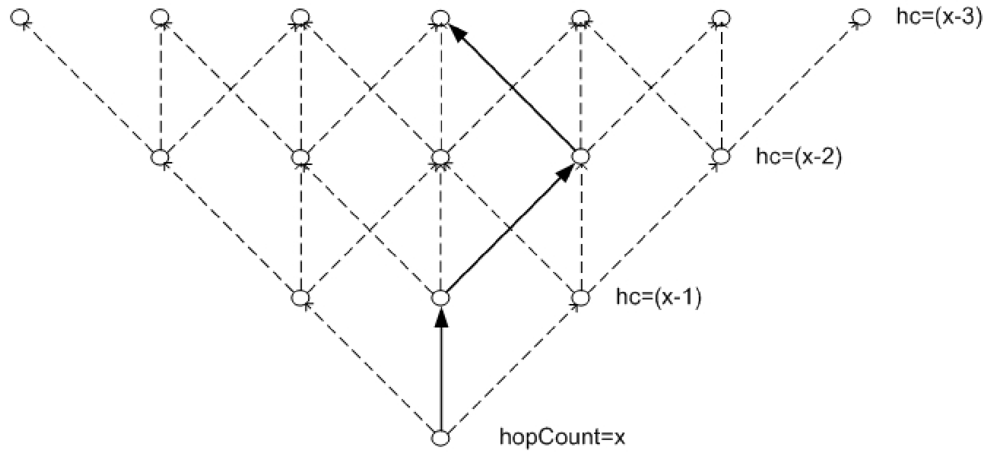

Figure 3.6 Routing Decisions taken 1 Hop at a Time.

predicted residual energy in one sensor node. When a current on-duty sensor node chooses the next on-duty sensor node from its neighbor, it will choose the sensor node with highest energy difference.

3.2.1.1 Routing Decision Hop Priority

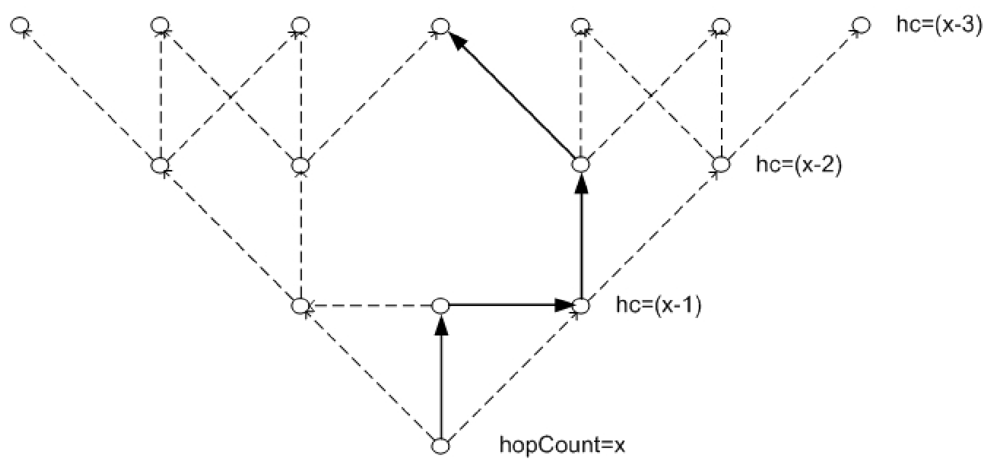

Figure 3.7 Routing Decision When a Sensor is not Connected to any Previous Hop Sensors.

As per the scheme used for assigning set number to the sensor nodes, it is possible that some sensor nodes, belonging to a set, might not be connected to any of its previous hop sensor nodes. This case has to be handled in routing phase. But the routing decision first always looks for route on lower hop sensor nodes and in case there is no lower hop sensor nodes connected, then only sensor node will look for route in same hop sensor nodes, i.e. priority is given more to the previous hop sensor nodes than the same hop sensor nodes while making routing decisions. This is represented in Figure 3.7.

3.2.2 Priority Based on Weather Forecast

Sensor nodes in WSN application are deployed in open field and consequently they are affected by the weather conditions. Based on the weather prediction, the routing has to efficiently utilize both regular and robust sensor nodes. During rain, it is most likely that only robust sensor nodes will be in running mode and regular sensor nodes will go down. When weather condition changes from rain to sunny weather, a few of the regular sensor nodes might will not come up and some of it will survive and start functioning. Considering all these conditions, routing will take decisions such that it consumes most of energy of regular sensor nodes during sunny weather. During rain, routing decisions considers only robust sensor nodes, as all regular sensor nodes will not be working. Depending upon the weather prediction, all these routing decisions are taken. This is depicted in Figure 3.8.

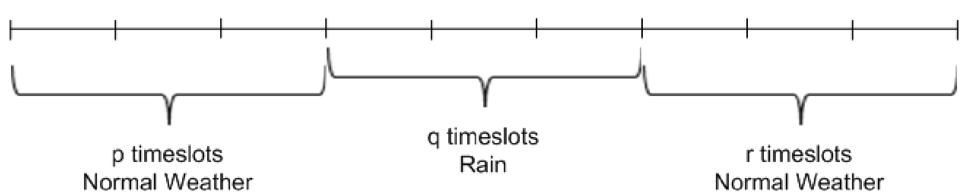

Figure 3.8 Example: Weather Condition During Different Timeslots.

As in the Figure 3.8, weather prediction of p-timeslots is sunny weather, next q-timeslots is rain and r-timeslots is again normal weather. It is known that during q-timeslots of rain, only robust sensor nodes will be working and when it changes from rain to normal weather, only few of the regular sensor nodes will come up and start functioning. The routing will try to utilize most of regular sensor node's energy during p-timeslots of normal weather, as later most of regular sensor nodes might stop functioning. During q-timeslots of rain, algorithm will determine route using only the robust sensor nodes. During next r-timeslots of normal weather, combination of robust and surviving regular sensor nodes will be used to determine the route.

The above discussion is more of general idea being used in the research, as in practical, application will not be knowing full weather forecast of the entire lifetime of the WSN application. As per the assumption, application will be knowledgable of only few hours, say 3-10 hours, of the weather forecast. This is as shown in the Figure 3.9. Therefore, based on current and future predicted weather timeslots, routing needs to determine which type of node, regular or robust, and which node is best to route data.

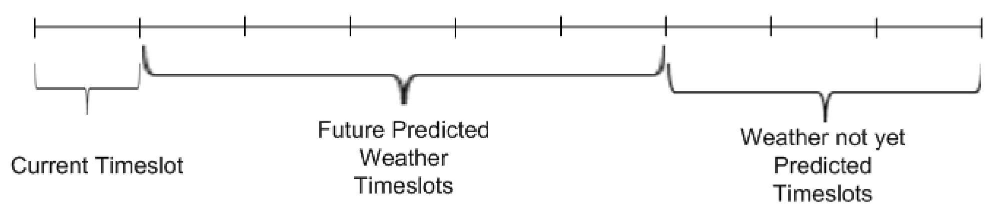

Figure 3.9 Weather Forecast in Practical.

If current timeslot is rainy, it is obvious that only robust sensor nodes will be in running mode. Consequently, routing needs to determine the best robust node to route data. This is shown in Figure 3.10.

If current timeslot is normal weather, as in Figure 3.11, the routing can choose either regular or robust sensor nodes. In this case, it mainly depends on the future predicted weather. If predicted weather has rain, then routing needs to give more priority to regular than robust sensor nodes. If predicted weather is full normal weather, then equal priority for regular and robust sensor nodes.

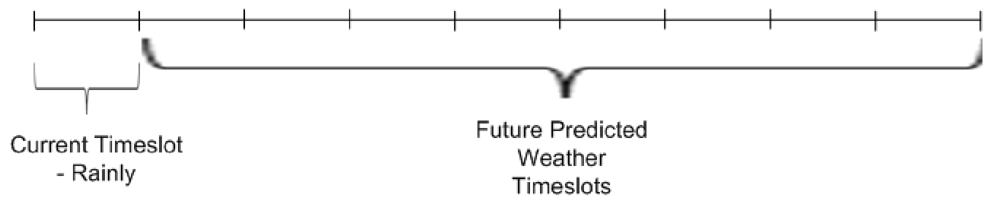

Figure 3.10 Current Timeslot is Rainy.

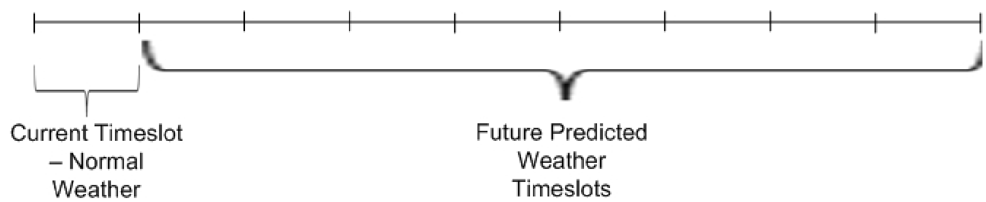

Figure 3.11 Current Timeslot is Normal Weather.

3.2.3 Energy-Difference Model

The Energy-Difference model calculates the energy difference between the current residual energy and predicted residual energy of a sensor node. This model is as shown in Figure 3.12. In the research, only one hop routing decisions are taken by a sensor node. This routing decision, of choosing the next sensor node to transmit data, is made by using the energy difference model. Sensor node, before transmitting the data, calculates the energy difference, using Energy-Difference model, of all the next possible sensor nodes. It will choose the sensor node with the highest energy difference to route data. The sensor node with higher energy difference means the sensor node has higher failure probability in the forecast hours comparing with other sensor nodes. The Energy-Difference model calculates the difference between current residual energy and predicted residual energy after few timeslots. Residual energy is a function of failure probability of a sensor node. During rain, regular sensor node's failure probability is higher than robust sensor node, and during sunny weather, both type of sensor nodes have same failure probability. This calculation is shown in Algorithm 3.

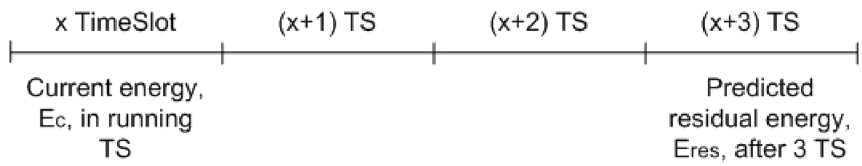

Figure 3.12 Energy-Difference Model of a Sensor Node.

Algorithm 3 Energy Difference Calculation of a Sensor Node

<u>`Notations:`</u>
T_fc: weather forecast hours
e_i: residual energy of a sensor node i
e'_i: predicted residual energy of a sensor node i
e_diff: predicted energy difference
pf_i: failure probability of sensor node i
e_slot: energy consumption of sensor node at a timeslot
N_slot: Number of unit time in a timeslot

$ec = e\_i$
$n = T\_fc$
for $j = 1 \rightarrow n$ **do**
  $ec = ec * (1 - pf^{(j)}_{i^{tp},w^{(k)}}) - e\_slot * N\_slot$
end for
$e'\_i = ec$
$e\_diff = e'\_i - e\_i$

The routing decision taken using Energy-Difference model is described as follows:

1. The sensor node, which requires to transmit collected data, determines hop priority and sensor node type priority based on its neighbor sensor nodes and weather forecast.

2. Sensor node short-lists all the possible neighbor sensor nodes based on the determined priorities.

3. Sensor node calculates energy differences, using Energy-Difference model, of all the possible neighbor sensor nodes. The energy difference calculation of a sensor node is as given in the Algorithm 3.

4. The neighbor sensor node with the highest energy difference is chosen to transmit data.

5. If more than one neighbor sensor node has same highest energy difference, then rotation mechanism, described in Section 3.2.4, is used to chose the sensor node between them.

3.2.4 Rotation Mechanism

While making routing decisions, sensor node determines the energy difference of all the possible next route sensor nodes. It is possible that two or more sensor nodes having same energy differences. This scenario is depicted in below Figure 3.13.

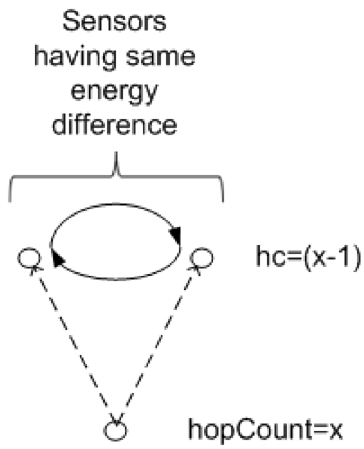

Figure 3.13 Sensor Nodes Having Same Energy Difference.

During this case, routing algorithm uses rotation mechanism between the sensor nodes having same energy difference to determine the route. In this rotation mechanism, each sensor records the previously made routing decisions and it chooses the sensor node which was not used in the nearest time. This mechanism provides fairness in choosing the sensor nodes.

CHAPTER 4

PERFORMANCE RESULTS

The research has been implemented and tested on ns2 simulator. Evaluations are conducted concentrating on how much it has prolonged the lifetime of the WSN application. The research has been divided into scheduling and routing phases. Each phase has its own goal and evaluations are conducted on both phases. The Section 4.1 and Section 4.2 are the evaluation of scheduling and routing phase respectively.

To evaluate the performance of the research, the schemes are simulated using several random 200 sensor node networks with different weather forecast. Sink is placed at (50, 0) in a 100m x 100m field. Simulated on different combinations of regular and robust sensor nodes. In most of the cases, the schemes are simulated using 100 regular and 100 robust sensor nodes. Simulation assumes all sensor nodes, both regular and robust sensor nodes, have same initial energy. Assumptions on weather forecast: during normal weather, failure probability of both regular and robust sensor nodes are same, whereas, during rain, regular sensor nodes have larger failure probability than the robust sensor nodes. Each sensor node will be knowledgable of the weather forecast of 5-10 hours from the current time. Simulations include evaluation of total sensing coverage, lifetime of WSN application, end-to-end packet delivery delay and other results.

4.1 Scheduling Phase Results

Scheduling plays a vital role in increasing the lifetime of WSN application. It divides the sensor nodes in the network into sets and each set of sensor nodes function in tandem. Evaluation of this phase would be to how efficiently it divides and assignes the sensor nodes to sets and how much coverage each set of sensors provide running individually.

The Figure 4.1 shows the number of sensor nodes and coverage provided by each set of sensor nodes. The experiment conducted to divide the sensor nodes into four sets. The graph shows that the coverage provided by each set of sensor nodes is greater than 80% of the actual area of interest. It is also observed that the count of sensor nodes in each set is consistent. While dividing sensor nodes into sets, it is not considering the type of sensor nodes, regular or robust. It is assumed that both regular and robust sensor nodes are evenly distributed in all the set of sensor nodes. Differentiation between regular and robust sensors will be done while finding the route to send data in routing phase. Scheduling phase mainly concentrates on proper distribution of sensor nodes between sets and the network connectivity because of this distribution.

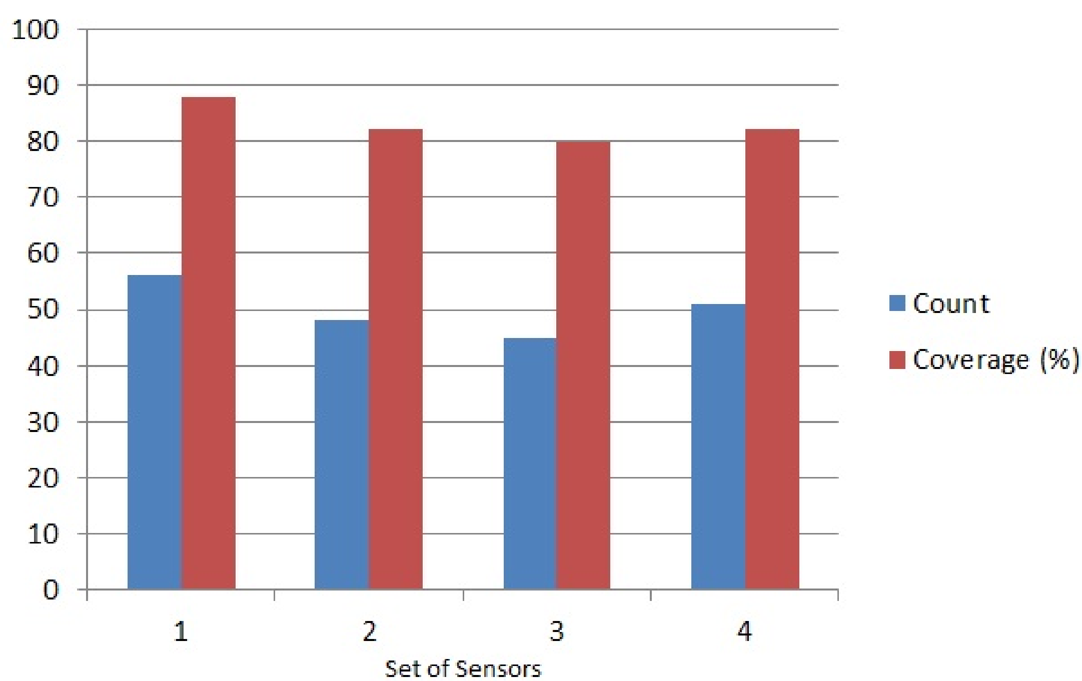

Figure 4.1 Count and Coverage of Each Set of Sensor Nodes, Where k = 4.

First task of scheduling phase is to determine how many number of sets it should divide the application sensor nodes. Determining this value is a trade-off between coverage provided by the set of sensors and the prolonged lifetime of the WSN application. Larger the number of sets value (k-value), larger the lifetime of the application and smaller the coverage provided by each set of sensor nodes. Likewise, smaller the k-value, smaller the application lifetime and larger the coverage. Choosing between this trade-off depends upon the requirements of the application. Experiments conducted uses intermittent k-value in order that it provides fair coverage and decent increment in lifetime of the application.

Figure 4.2 shows the experiments conducted on different number of sets value and output being the average percentage coverage provided by set of sensor nodes and how much percentage of lifetime it prolonged. The lifetime of application in this case does not include the routing phase.

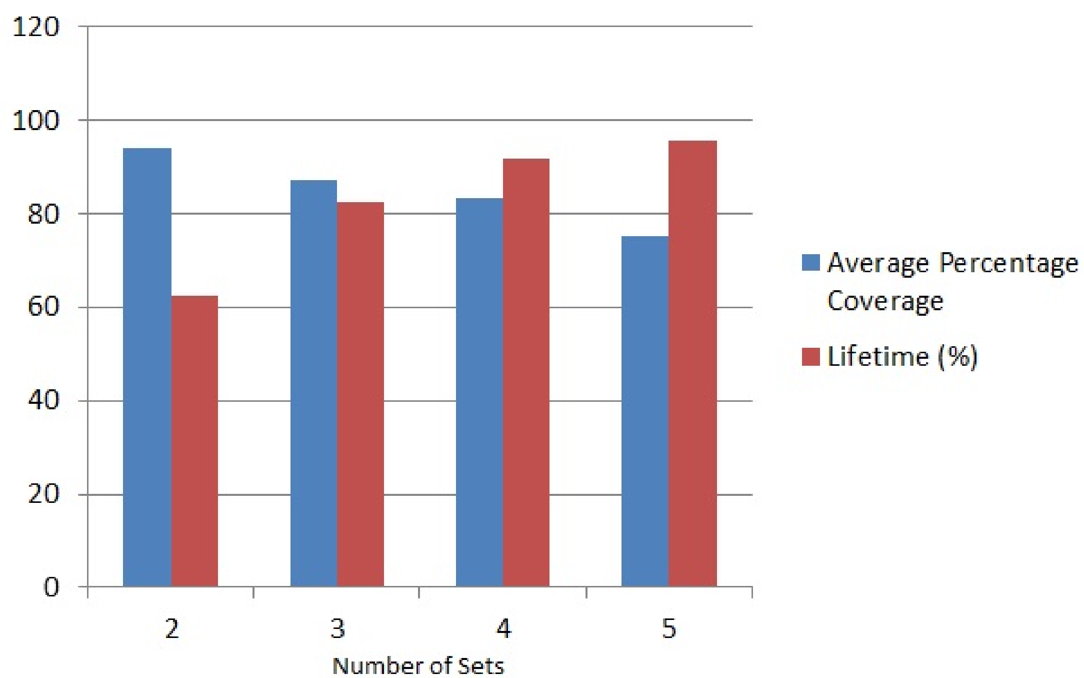

Figure 4.2 Average Coverage of Set of Sensors and Application Lifetime w.r.t. Number of Sets.

4.2 Routing Phase Results

Routing phase task is to intelligently determine the route to transmit data considering the environmental parameters. To deal with the environmental effects, routing takes advantage of regular and robust sensor nodes deployed in the WSN application. In the experiments, it is assumed that the number of regular and robust sensors in the application are equal. Key feature of robust sensor node is that it can keep functioning even during environmental effects, like rain, whereas regular sensors go down. It is also assumed that each sensor node, irrespective of type of sensor, will be knowing few hours (5 hours in the experiment) of weather forecast.

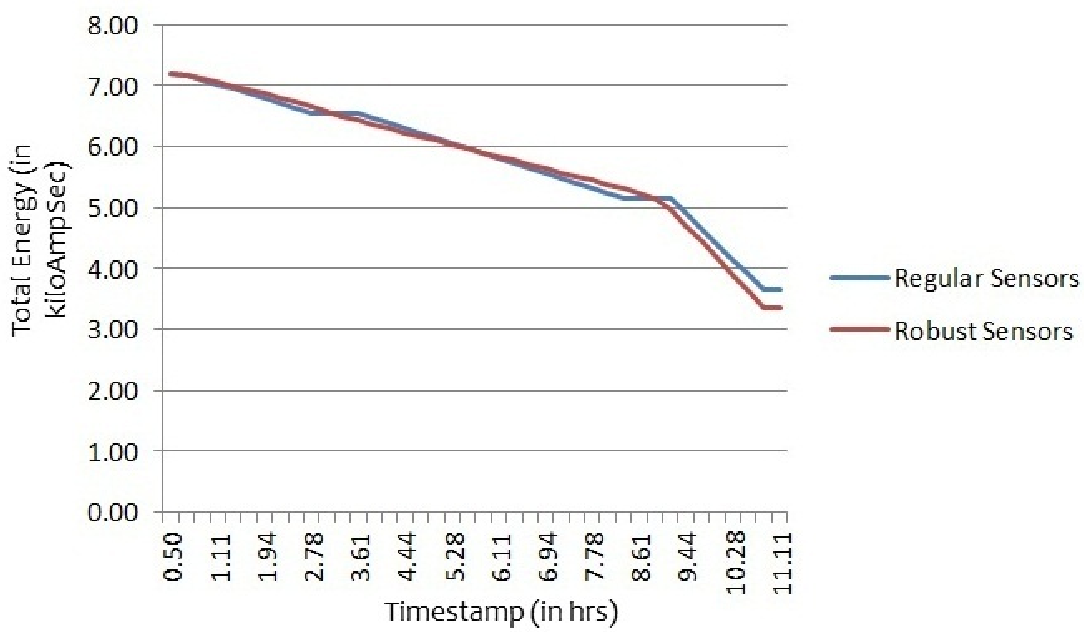

Figure 4.3 WSN Application Lifetime. Weather Forecast: Rain Between 5000 & 6500 and 15000 & 16500 Timeslots.

The main idea in this phase is that if there is a prediction of rain in the future, it should give more priority to regular sensor nodes than robust sensor nodes to route data. This is because, during rain, regular sensor nodes will go down and only robust sensor nodes will be in running state. In this manner, the routing phase is trying to distribute the energy consumption of regular and robust sensor nodes.

4.2.1 Energy Distribution between Regular and Robust Sensor Nodes

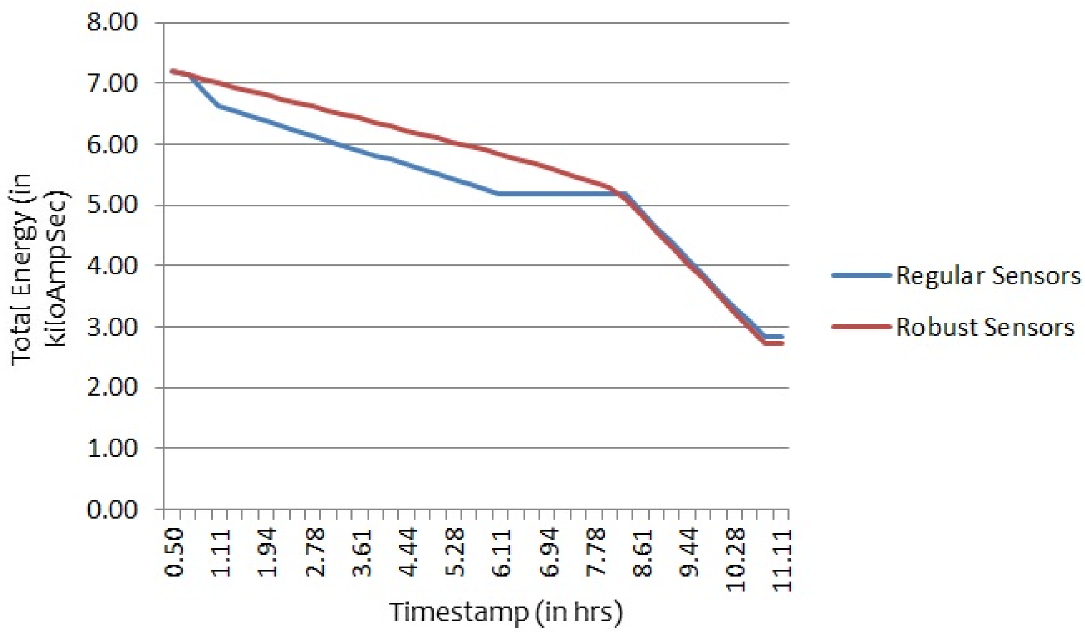

Figure 4.4 WSN Application Lifetime. Weather Forecast: Rain Between 11000 & 15000 Timeslots.

Figure 4.3 and Figure 4.4 show the results for different weather forecast. Experiments use 100 regular and 100 robust sensor nodes. Both type of sensor node have equal initial energy, but different failure probability during rain. Experiment using different ratio between regular and robust sensor nodes are also conducted and are explained in Figure 4.5.

In the experiment shown in Figure 4.3, weather forecast is rain between 5000 & 6500 and 15000 & 16500 timeslots and normal weather during other timeslots. It can be observed that priority is given regular sensor nodes few timeslots before 5000 and 15000 as there is prediction of rain. It can also be observed that during rain, only robust sensor nodes work and energy will be drained only from robust sensor nodes, not from regular sensor nodes. After 16500 timeslot, as there is no prediction of rain, equal priority is given to both regular and robust sensor nodes and thus, energy are equally consumed.

In the experiment shown in Figure 4.4, weather prediction is rain between 11000 & 15000 timeslots and normal weather during other timeslots. The graph clearly shows that regular sensor nodes are given more priority than robust sensor nodes before 11000 timeslot and equal priority after 15000 timeslot. From the graphs, the research has evenly distributed the energy consumption of both regular and robust sensor nodes even when the field is struck by rain.

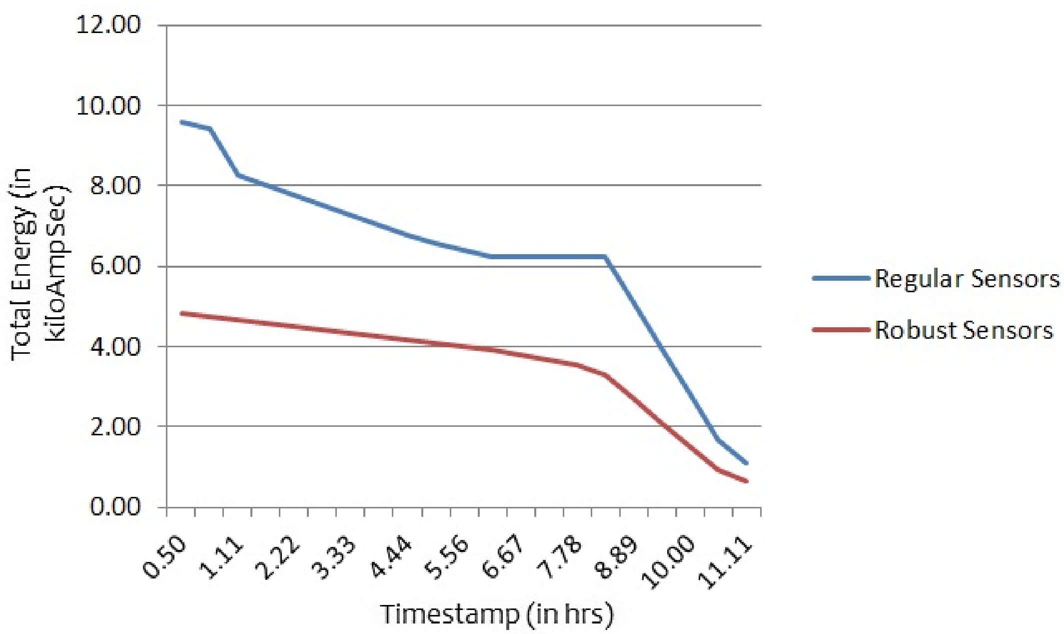

Figure 4.5 WSN Application Lifetime. Sensor Node Distribution: 133 Regular Sensor Nodes & 67 Robust Sensor Nodes.

Figure 4.5 shows the energy distribution of regular and robust sensor nodes, ratio between them being 2:1 respectively, i.e. 133 regular and 67 robust sensor nodes. As regular sensor nodes are cheaper than the robust sensor nodes, it is likely that the WSN application will using more of regular sensor nodes than robust sensor nodes. Therefore, most of the experiments conducted are for equal ratio or regular sensor nodes having higher ratio than robust sensor nodes. Energy distribution graphs shows that, even with varying ratio between regular and robust sensor nodes, the energy consumption is distributed properly.

4.2.2 End-to-End Packet Delivery Delay

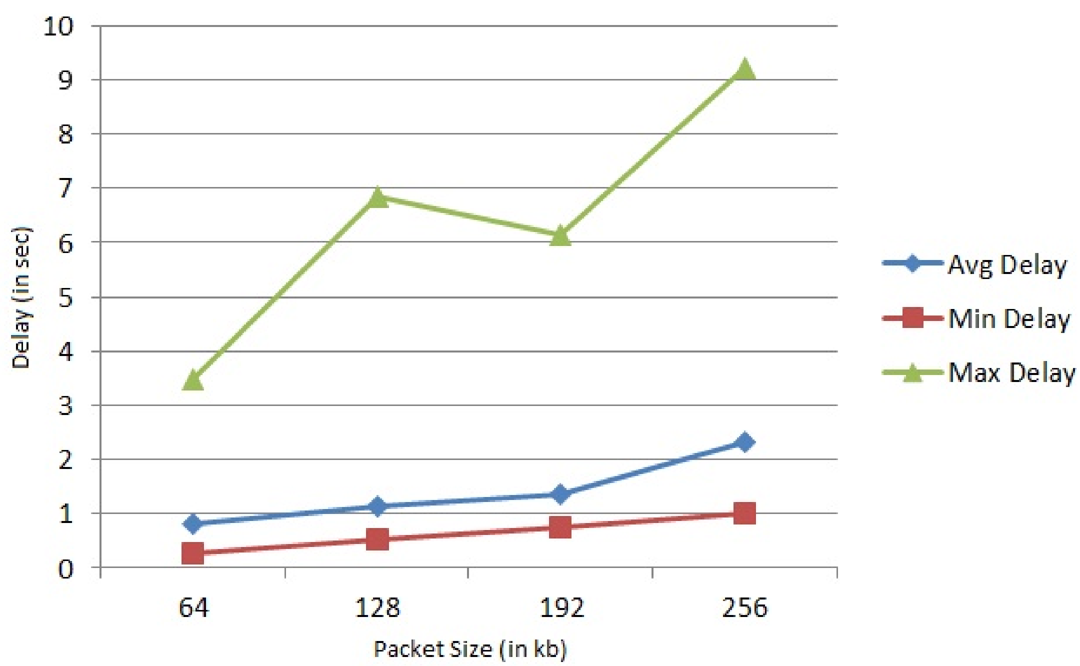

Figure 4.6 End-to-End Packet Delivery Delay w.r.t. Packet Size.

End-to-End packet delivery delay is the time taken by the data packet to traverse from the source sensor node to sink. The packet delivery delay graphs will point any packet loss during transmission of data and in what cases this packet loss has occurred. Figure 4.6 shows the average, minimum and maximum packet transmission delay w.r.t. the packet size. It is observed that packets with minimum packet delay are transmitted by 1-hop sensor nodes and packets with maximum packet delay are transmitted by the furthermost sensor nodes. In the simulation, 1 timeslot is equal to 10 seconds. As maximum packet delay for packet with size 256 kb is not more than

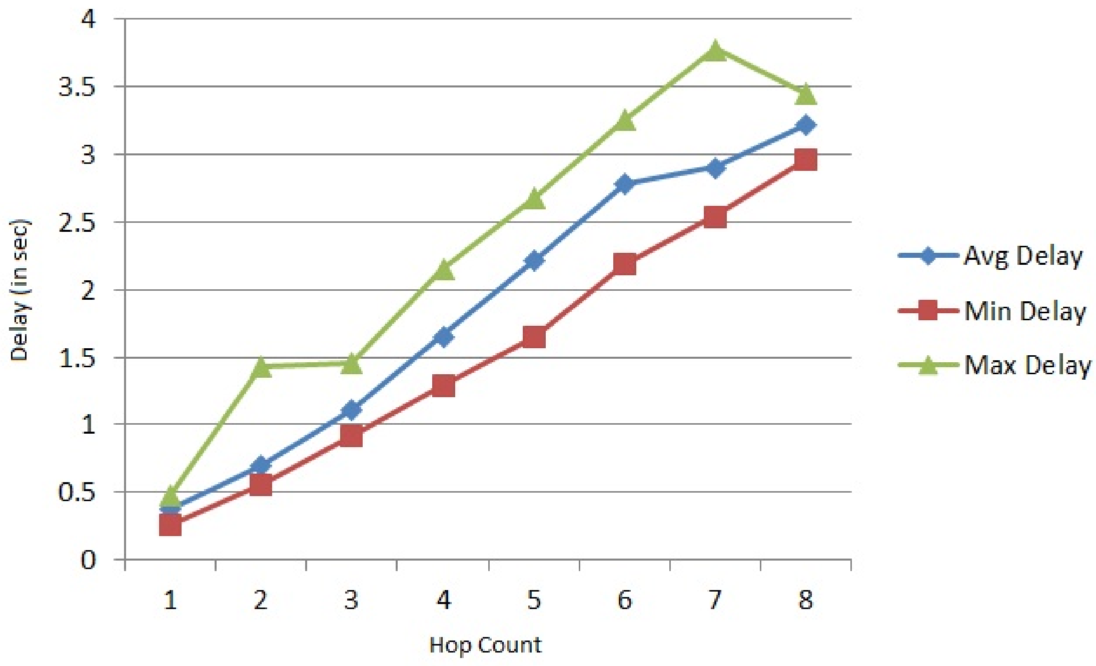

Figure 4.7 End-to-End Packet Delivery Delay w.r.t. Hop Count. Packet used is 64 kb.

1 timeslot, it is concluded that there is no packet loss during this experiment. If the packet size is increased further, then there is chances of packet loss.

Figure 4.7 shows the average, minimum and maximum packet transmission delay w.r.t. the sensor nodes hop count. The average packet delay is consistent w.r.t. the hop count sensor nodes, concluding that even by dividing the sensor nodes into sets, it hasn't introduced much packet delay.

4.2.3 Lifetime of WSN Application w.r.t. Different Schemes

Experiments are conducted with the combinations of the research scheduling, routing, random routing and no scheduling schemes. Figure 4.8 shows the result of these experiments. Lifetime calculation is performed with the benchmark using no scheduling and random routing schemes. It can be observed from the graph that by only using

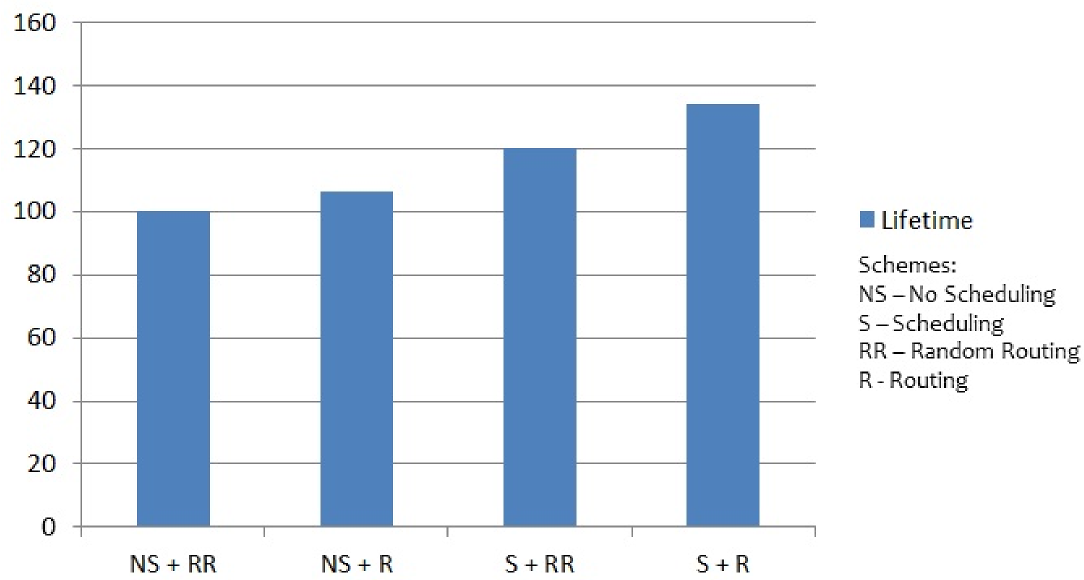

Figure 4.8 Lifetime of WSN Application w.r.t. Different Combination of Schemes.

the research scheduling or routing, lifetime of the WSN application has increased and by using both scheduling and routing, lifetime is prolonged significantly.

4.3 Other Results

Node-level evaluation is also very important to evaluate any protocol. From the experiments, results are process to find to what percentage the sensor nodes prolonged their lifetime in the WSN application. Based on the calculation, sensor node which failed first due to energy drain, prolonged its lifetime by 20-25% and sensor which failed last, prolonged its lifetime by 45-50%. On an average, each sensor stretched its survivability by 30-35%.

Assumption that the network is disconnected when all the 1-hop sensor nodes are down. Experiments show that the lifetime of the WSN application is prolonged by more than 30% before network being disconnected. This means, on an average all 1-hop sensor nodes go down and prolong lifetime of the application by more than 30%.

CHAPTER 5

RELATED WORK

In recent years, the research field of WSNs has been very active. Finding an energy efficient way to perform scheduling and routing is among the top research priorities. Many research works are conducted dedicated completely on scheduling or routing, and some works are a joint of both.

Many research works have been devoted to scheduling that concentrates on efficiently turning-off redundant sensor nodes for energy saving [21][2][16][4]. In all these studies, sensor nodes are deployed either in grids or randomly. Energy-efficient scheduling works are surveyed in [10]. Some scheduling works even work on solving coverage and connectivity problems [18][8][4]. Research works on routing protocols can be broadly classified [9] into four classes: data centric, hierarchical, location-based and network flow and QoS awareness.

One such similar research work in scheduling is conducted in randomized scheduling algorithm [4]. References [6][3][15][5] have studied sensing coverage problems and References [1][11][20] have studied connectivity problems. But it is hard to combine and solve sensing coverage and network connectivity problems together. Reference [4] provides solution to the joint problem without the availability of per-node location information. The paper mainly aims at providing three features: sensing coverage above the given requirement, all active sensor nodes are connected and each active sensor node known at least one shortest route to sink.

Another work, [19], mainly investigated the properties of randomized scheduling algorithms in sensor networks. This focuses on performance modeling and mathematical properties of a random coverage algorithm (also called k-set randomized scheduling algorithm) for WSNs. Many works, such as in References [21][6][4][3][15][5], use only network sensing coverage as the QoS constraint. The research in Reference [19], in addition to the coverage intensity, it also considers detection delay and detection probability as very important measures. It also provides an efficient search algorithm similar to binary search for obtaining the optimal solution is discovered based on the properties of performance metrics. The goal of scheduling in this research work is to provide consistent coverage by all the set of sensor nodes and each set of sensor nodes running independently should not cause network partitioning.

Previous research work on routing can be broadly classified as: data centric, hierarchical, location-based and network flow and QoS awareness. Hierarchical based work is more related to this research work. Hierarchical protocols separate the nodes into clusters in order to segregate the areas of the monitoring environment as LEACH [17] and PEGASIS [13] [14]. To allow communication between the clusters, a leader is selected from each cluster. Leaders are responsible for the management and transmission of the collected data in the region they control. The idea of Low-Energy Adaptive Clustering Hierarchy (LEACH) is to form clusters based on the received signal strength and use local cluster heads as routers to the sink. All the data processing such as data fusion and aggregation are local to the cluster. This will save energy since the transmissions will only be done by such cluster heads rather than all sensor nodes. Power-Efficient GAthering in Sensor Information Systems (PEGASIS) forms chains from sensor nodes so that each node transmits and receives from a neighbor and only one node is selected from that chain to transmit to the sink. Gathered data moves from node to node, aggregated and eventually sent to the base station. Hierarchical-PEGASIS is an extension to PEGASIS which aims at decreasing the delay incurred for packets during transmission to the base station and proposes a solution to the data gathering problem by considering energy*delay metric. Hierarchical protocol uses different clustering protocols to determine the energy efficient route. These do not consider the environmental forces which calculating the routing path as considered in this research scheme. In this research, Energy-Difference model, which considers the environmental forces, is used to take routing decisions. Other features that the work is embedding with the routing are: routing decision hop priority, priority based on weather forecast and rotation mechanism.

All of these previous works concentrate on determining the energy-efficient algorithm considering only the energy levels of the sensor nodes, but not the effects of environmental forces. This research considers both environmental conditions and the energy levels of each sensor nodes providing an energy-efficient scheme to prolong the lifetime of the sensor network.

CHAPTER 6

CONCLUSIONS

In recent years, routing has attracted a lot of attention in the field of sensor networks to solve the energy problem, but not much work has been conducted to overcome the environmental problems considering the heterogeneity of the sensor nodes. The research provides a combination of scheduling and routing protocols to solve both the environmental and energy problem and thereby increase the lifetime of the WSN application. The scheduling protocol properly distributes sensor nodes into the sets and schedules them. Routing protocol takes predictable environmental forces and the energy model of regular and robust sensor nodes into consideration and determines the best possible route from a sensor node to the sink. Evaluation of these two protocols, individually and together, has shown significant improvement in prolonging the lifetime of a WSN application.

REFERENCES

[1] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2001)*, July 2001.

[2] C. Hsin and M. Liu. Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithm. In *Processing International Symposium Information Processing in Sensor Networks (IPSN '04)*, 2004.

[3] C. Liu, K. Wu, and V. King. Randomized coverage-preserving scheduling schemes for wireless sensor networks. In *International Symposium Information Processing in Sensor Networks (IPSN '04)*, May 2004.

[4] C. Liu, K. Wu, Y. Xiao, and B. Sun. Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks. In *IEEE Transactions Parallel and Distributed Systems*, volume 17, pages 562–575, June 2006.

[5] D. Tian and D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the ACM Workshop on Wireless Sensor Networks and Applications*, October 2002.

[6] G. Wang, G. Cao, P. Berman, and T.F. La Porta. Bidding protocols for deploying mobile sensors. In *Processing IEEE Transactions Mobile Computing*, volume 6, pages 563–576, May 2007.

[7] H. Xiang and L. Bin. A loose time synchronization regime for wireless sensor network. In *2nd International Conference Signal Processing Systems (ICSPS '10)*, July 2010.

[8] H. Zhang and J. Hou. Maintaining coverage and connectivity in large sensor networks. In *Processing International Workshop Theoretical and Algorithmic Aspects of Sensor and Ad-Hoc Networks (WTASA '04)*, 2004.

[9] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. In *Ad Hoc Networks*, volume 3, pages 325–349, 2005.

[10] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. In *ACM/Springer Mobile Networks and Applications*, volume 11, pages 723–740, October 2006.

[11] P. Godfrey and D. Ratajczak. Robust topology management in wireless ad hoc networks. In *Proceedings of the Third International Symposium On Information Processing in Sensor Networks (IPSN 2004)*, April 2004.

[12] R. Weizheng. A rapid acquisition algorithm of wsn-aided gps location. In *Intelligent Information Technology and Security Informatics (IITSI '09)*, January 2009.

[13] S. Lindsey and C.S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2002.

[14] S. Lindsey, C.S. Raghavendra, and K. Sivalingam. Data gathering in sensor networks using the energy*delay metric. In *Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.

[15] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor neworks. In *Proceedings of IEEE Infocom 2001*, April 2001.

[16] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Processing IEEE International Conference Communications (ICC '01)*, 2001.

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

[18] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Processing ACM International Conference Embedded Networked Sensor Systems (SenSys '03)*, 2003.

[19] Y. Xiao, H. Chen, K. Wu, B. Sun, Y. Zhang, X. Sun, and Chong Liu. Coverage and detection of a randomized scheduling algorithm in wireless sensor networks. In *Processing IEEE Transactions on Computers*, November 2009.

[20] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin. Topology control protocols to conserve energy in wireless ad hoc networks. In *CENS Technical Report 0006*, January 2003.

[21] Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, 2004.