Direct Diffusion Routing Algorithm for Wireless Sensor Networks (DDRA for WSN)

"Improving the Direct Diffusion with a Clustering Method based on network topology"

A Dissertation by
Triantari S. Foteini

Advisor: Professor Nicholas S. Samaras

**T.E.I. of Larissa
Greece**

Staffordshire University

**TECHNOLOGICAL EDUCATIONAL INSTITUTE OF LARISSA
STAFFORDSHIRE UNIVERSITY OF UK**

A dissertation submitted in partial fulfilment of the requirements of
Staffordshire University for the degree of M.Sc.in Computer Science

June 2014

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF LARISSA
STAFFORDSHIRE UNIVERSITY OF UK

Direct Diffusion Routing Algorithm for Wireless Sensor Networks (DDRA)
"Improving the Direct Diffusion with a Clustering Method based on network topology"

Thesis Summary

Routing protocols for Wireless Sensor Networks are challenging due to hardware constraints of sensors. Sensors are autonomous and must consume extremely low power. In addition, sensing task is a periodically process and most of the routing techniques are application-dependent.

In this thesis we propose an energy-efficient routing protocol for WSNs. Direct Diffusion Routing Algorithm for Wireless Sensor Networks (DDRA for WSN) combines two basic techniques of traditional routing approaches in WSNs. Passive Clustering (PC) and Direct Diffusion (DD). DD is a data-centric approach . This query dissemination method selects empirically energy efficient paths and by caching achieves data aggregation.

This routing protocol can be implied on application-layer.

PC on the other hand, is a well-known technique for saving energy and prolong network's lifetime. Clustering algorithm adds adaptive hierarchy in networks and reduce overall energy consumption by setting different roles to sensors according ,mostly on topology and energy cost.

In our DDRA approach there is a combination of PC and DD and new election scheme of clustering. Election of clusters is a hybrid method, which depends on three basic factors, network topology, energy resources and distance from destination. As in LEACH algorithm our approach change the role of a CH in the network based on a certain probability. The major aim is to reduce energy consumption of each node and generally of entire network.

Approaches based on DD and PC motivated many scientists, and the main goal of these researches was an improved energy efficient solution. Sensors are devices which must maintain connectivity and balance the load of network in order to maximize network longevity. Fault tolerance is another important issue. As sensors mostly deployed for monitoring of environment , physical damage, environmental interference and lack of power are crucial problems. On the other hand, high density areas like WSNs require also data aggregation, elimination of duplicates and traffic optimization. Routing protocols in WSN is still a challenging issue specially now where mobility and size reduction of devices are future trends.

To summarize, DDRA for WSN can be described as an approach for improving energy efficiency. Well-known problems of networks with sink and source communication are energy exhaustion of CHs and creation of "bottlenecks" near the region of transmitter and receiver. Despite the fact, clustering helps to distribute energy consumption into remote areas and avoid creation of energy holes, the problem of energy still remains significant.

Keywords

Clustering, direct diffusion, energy efficiency,network topology, node degree, graph diameter,residual energy, sensors, routing, intra-cluster communication, inter-cluster communication,delay.

1

Acknowledgements

I would like to express my deep gratitude to Professor Nicholas S. Samaras, my research advisor for his patient guidance, valuable suggestions and encouragement during the development of this thesis. He also helped me keeping this work on schedule and I appreciate the time he gave me so generously. Throughout this study he has been an excellent mentor and teacher and without his support this study would never have been possible.

I am particularly grateful for the love and emotional assistance of my family. Finally I would like to dedicate this thesis to my father, Serafeim Triantaris, who has always been there when I needed him. I think this is the least I can do for him.

Triantari Foteini
Larissa,June 2014

3

Contents

Appendix A

List of Figures

Chapter 8

List of Tables

List of Symbols

n_i	Node i
$N(n_i(Cid))$	Neighbor Cluster Head IDs of node n_i
n_j	Neighbor gateways of node j
$n_j(N(Cid))$	Neighbor Cluster Head IDs of n_i's neighbor gateways
E_r	Energy remaining in the node
E_m	Initial energy in the node (maximum energy)
N_{elt}	Number of neighbors nodes with residual energy less than a certain threshold
N_n	Number of all nodes in network
D_{ref}	Longest Distance from Base station (network diameter)
D_{CH_BS}	Distance from corresponding node and Base Station
E_n	Remaining energy of node
E_{CH}	Remaining energy of cluster head
D_{n_CH}	Distance between ordinary node and cluster head
D_{CH_BS}	Distance between cluster head and Base Station
E_{res}	Energy residual on the node
E_{max}	Initial energy on the node(maximum energy)
N_{neigh}	Number of nodes connected with node in 2-hops at maximum
N_{net}	Number of total nodes in the network
D_{ref}	The longest distance between sink and node, (in our example, we suppose source is the furthest node)-Graph Diameter
D_{ch_s}	Distance between candidate cluster head and source
$D_{maxh\_in\_cluster}$	Maximum number of hops in the cluster (set as default to 2-hops)
R_b	Transmission bit rate
P_{pa}	Power consumption of amplifiers
P_c	Power consumption of circuit blocks

11

V	Number of bytes
N	Number of messages
Ebt	Energy consumption per byte of a sensor
$E_{snd\_msg-rnd}$	Energy consumed to send a message to a random gate (next neighbor)
E_{snd_msg}	Energy consumed to send a message and track the path
E_{cl}	Energy consumed by a node while forming clusters
N	Number of bytes of messages
E_{exp_msg}	Energy consumed to send exploratory data message to next node
E_{grad_msg}	Energy consumed to send gradient message to next node
$Etx(i,j)$	Energy consumption of node I transmitting to node j
E_{RCV}	Energy consumption of a node when receiving data
Ec	Constant energy consumption of a sensor while there is no data collection
$PL(d)$	Path loss at distance d
$PL(d_0)$	Known path loss, when we have a known distance d_0
n	Path loss exponent
$X\sigma$	Gaussian zero-mean random variable with deviation σ

List of Abbreviations

WSN ....Wireless Sensor Networks
6LoWPAN ....IPv6 over Low-Power Personal Area Network
DD ....Direct Diffusion
PC...Passive Clustering
PCDD ....Direct Diffusion with Passive Clustering
DDRA...Direct Diffusion Routing Algorithm
CH ....Cluster Head
CM..Cluster Member
SN...Sensor Node
BS...Base Station
D...Distance
CCR..Corner Cube Reflector
ISM...Industrial ,Scientific and Medical

Chapter 1

Introduction

This chapter is a summary for the motivations of DDRA,the design of protocol and the methodology chosen for this implementation.

A Wireless Sensor Network (WSN) is a wireless network consists of autonomous devices called sensors. Sensors are deployed spatially in the area and usually monitor the environment and physical conditions. The main tasks of a sensor is data sensing, data processing and communication with other nodes in the network.

Sensors due to cost and size constrains have computational limitations, such as power consumption, battery resources and localization to provide connectivity with other nodes in the network. A typical architecture of WSN includes a sink/Base Station and sensors deployed randomly. Sensors collect data, do data aggregation and relay aggregated data to neighbors. Data must reach in a hop-by-hop communication to Sink and Sink process the data. As a sensor will not have only the duty of sensing and sending data, but also transmit data came from neighbors (retransmitting queries for example), energy cost will be increased. Especially, for nodes that are placed near Sink , in a typical WSN architecture of a static sink topology energy cost is a basic attribute. Sensors near static sink will have to accept and transmit massive data, leading to energy depletion. This is the hot-spot problem, where sensors farther from Sink have significant energy left and sensors near Sink will be soon exhausted and create an energy hole of the system.

Considering the characteristics of WSNs, energy resources of each node must be consumed wisely to prolong lifetime of sensors and generally, lifetime of entire network.

Routing in WSNs is a very challenging issue considering the design issues of a WSN. Factors like node deployment, connectivity, energy consumption avoiding loosing accuracy, data aggregation, fault tolerance, scalability and Quality of Service are very important in a routing algorithm.

Besides, sensors are not recently developed, have existed for several decades , the 1998 SmartDust Project and NASA Sensor Webs Project show that future trend is the size reduction of these autonomous devices(motes operate in a cubic millimeter space), the lowest energy consumption and easier communication and even interaction with users. In addition, there are many applications based on WSN architecture like environmental applications, health applications, home applications and many other commercial applications.

Related work on routing protocols on WSNs can be categorized to flat-based solutions, hierarchical-based solutions and location-based solutions. In the first category all nodes plays an equal role in routing, in second category nodes have different functions in the network communication and finally in last category the position of a node will affect the process of route discovery to the destination. Furthermore, routing protocols can be distinguished to proactive, reactive and hybrid approaches depending on how the route discovery from source to destination (sink) is done. In proactive approaches the route is computed before its necessity, in reactive approach the route is

discovered on demand and hybrid solution combines these two techniques. Many techniques introduce the issue of duty cycling. As sensors send periodically data, they can have a energy-save mode after their operation. In this this time period they can turn off their transmitters and receivers. Although there is a sufficient reduction in energy wastage, mechanism of sleep and wake up periods requires additional coordination between nodes.

In more details, many routing approaches on WSN are based in clustering method. Other solutions involve query-based routing . In our approach there is a combination of Clustering Method and Direct Diffusion (DD) routing method in WSN.

DD is a data-centric routing. This is a technique about local interactions between nodes of a system and is an application-layer based method. In DD, when sink has a request, an interest called so, floods the network with an interest message. This interest contains data about which query (which event) sink is interest to. Sensors get this interest, set up gradients for this request and finally sensed data that matches this request travel from sensors to Sink. Obviously, sensors have to maintain information in their cache memory , have to update their gradients and have to send periodically data to their neighbors.

Passive Clustering (PC) is a mechanism where the main objective is to set different roles in a set of sensors and optimize the exchange of flooded messages. Sensors depending on their energy resources or location play the role of Cluster Heads or Cluster Members. PC differs from traditional Clustering method because PC is done only after first message exchanging and flooding in system and not in the initial phase of the network communication. PC tends to reduce flooding messages and create a distributed solution on network making "strong" nodes play the role of routers and "weak" nodes play the role of sensing and sending data to corresponding Cluster Head.

To conclude, DD is a method for searching in a WSN for a specific event (interest) as a query-based routing and PC is a technique for minimizing energy costs and prevent hot-spot problem in communication between Source and Sink of a system.

As a WSN is a highly dynamic system and does not have a specific infrastructure the main challenges on routing can be summarized into four categories:

- Avoidance of single-point failure
- Energy conservation
- Connectivity over time
- Prolong of network lifetime

In our approach we concentrate into the energy consumption of each node and of entire network, considering three characteristics of network, a) topology and location of each node, b) energy resources of node and c) optimal route discovery between source and sink of a WSN system. Combining a data-centric method and clustering formation we expect to achieve balance of energy consumption particularly to avoid hot-spot problem, reduction of data redundancy caused by flooding messages of DD and prolong of network lifetime through election and creation of clusters. Clusters would help to prevent nodes close static sink accept massive data, network overload and become bottlenecks in data communication.

Apparently many limitations have been revealed through this research. The feature of DD is based on flooding messages in contrast with one of our main challenge which is to reduce the data redundancy (duplicated messages are generated) and the energy exhaustion of nodes near sink (Hot-Spot problem). Another important issue is

the creation of a backbone on CHs communicate each other. In many approaches, a backbone of nodes acting like gateways (accept and relay data of CHs) leads to great energy wastage and nodes placed on this backbone will not be easily recharged. These two facts lead to great energy cost and creation of single-point failure near destination.

Our protocol, DD routing approach (DDRA) manages to create distributed clusters farther from source , to discard an amount of duplicated data exchange and achieve an improvement of end-to-end delay. Apart from this, experimental results revealed the following: problem of hot-spot appears more in smaller network (consist of 12 nodes) and energy consumption tends to increase during simulation time near this area of simulation model.

In conclusion, our routing approach concerns subjects on energy wastage and data redundancy and by maximizing the size of network we can highlight a better performance of energy consumption and decrease of end-to-end delay of data to reach destination (sink).

1.1 Motivation

One of the most important question that may derive about the importance of creating a new algorithm is the following. *"Is this necessary or the already research in routing algorithms determine that no longer improvement will bring new solutions?"* , we may think. To explain the main motivation of this research, three main reasons makes this issue very interesting and open for further investigation.

Firstly, new wireless technology and WSN are very challenging systems concerning computation limitations, transmission medium on wireless communication and network connectivity of entire field they are placed and monitor.

This means that the architecture of sensors has characteristics like ad hoc network, with high mobility and definitely must be a fault tolerant system.

Secondly, the need of saving energy. Not only on WSN architecture but in other aspects of technology , for example , the huge expansion of green communications is a crucial and challenging issue in researches and scientist papers recently. A routing algorithm which gives to energy consumption a more important role will try to balance (as not in traditional approach between throughput and delay) , but between throughput, prolong of sensor lifetime and generally the whole network and saving energy of communication.

Moreover, the main trend of market which makes all hardware components be smaller at size, for example iPhones, iPads indicates that we are gradually driven to new technologies where size will be limited, obviously energy and battery resources will be limited too, and the need of new ways of communication will be an important issue. To strictly agree on that, minimized equipment is an innovative trend in technology for Wireless Body Personal Areas too.

This research has as main context the network layer of the architecture of sensors. The focus of this dissertation is the way data can be exchanged in order to set sensors consume less energy. There are lots of solutions about limited energy consumptions. Scientific papers about power mechanisms , for example *Power Saving Protocol for 802.11 Based on Multi-Hop Ad-Hoc Networks* proposed by *Yu-Chee Tseng* et al. in [1] sets the mobile hosts operate in low-power-switch mode. Although many other techniques have been discussed such as depending on localization factor, here this

proposal is about clustering algorithm. Clustering contributes to have a more distributed solution, avoiding bottlenecks on network and energy exhaustion of sensors close to the backbone of network.

The main motivation about this research paper has been given by research conducted at Technical University of Berlin. *Vlado Handziski, Andreas K¨opke, Holger Karl, Christian Frank, and Witold Drytkiewicz* presented in [2], a new idea about improving DD with PC. Their research paper, published on 4th European Conference in Wireless Sensor Networks on 2007, as title "***Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering***" achieved to combine energy efficiency, formation of clusters and DD, "*a data-centric routing protocol*", as they describe in their work. Data-centric routing refers to flow of data in one direction.

The aim of this research can be summarized as trying to explore more capabilities of PC and combine techniques to achieve greater energy saving. The main point is to observe how network will operate if into election of Cluster Heads and Cluster Members, we will count three parameters, residual energy of each node, distance of node from source, (the generator of Interests) and the number of neighbors near the corresponding node.

In this way, the main effort is about searching for solutions that will depend on energy and will select Cluster Heads in more effectively way.

1.2 Aims of Routing Protocol

The aims of DDRA for WSN can be summarized to the following points:
1. Reduction of Data Redundancy, by decreasing number of messages.
2. Avoidance "Hot Spots" problem by setting cluster heads having greater energy resources.
3. Decrease energy consumption by changing inter-cluster and intra-cluster communication.
4. Avoidance of Additional Headers and Filters on messages, to reduce network load and improve network bandwidth.
5. Maintenance of short length messages, by setting limited and constant length.
6. Permission to alternate nodes play the role of cluster heads when CHs are out of energy to sustain communication and connectivity.

In conclusion we can mention that the main goal of DDRA for WSN is first of all an energy-efficient solution, based on balanced groups of sensors (set as clusters) and the maintenance of an optimal path between source and final destination. DD has been exploited as an application-aware technique, like a query-based routing. On the other hand, PC is used to prevent unbalanced energy consumption. The main difference between our DD routing approach and *Vlado Handziski's et al.* solution [2] is that here we do not have a random and first declaration announcement of Cluster Heads. Here we impose a hybrid approach on CH election depending on network topology and energy resources. Proactive firstly and reactive in following phases of route discovery.

1.3 Research Methodology

The research methodology was conducted under these four basic steps:

1. Construction of a routing algorithm, based on energy and localization between nodes.
2. Simulation of three types of networks, having different number of nodes and representing the performance of this proposed solution.
3. Emphasis on the parameters, the factors and the results about implementation of this routing algorithm.
4. Conclusions on how this proposal can improve WSNs on energy efficiency and prolong of network lifetime.

Our routing approach borrows the idea of DD and PC. The roots of this DDRA can be described as a cluster-based scheme. By having a proactive election scheme on clusters we try to have only "advanced" nodes determine optimal routing paths. In addition, the cluster formation continues and keep choosing alternative nodes when cluster heads are reduced significantly in energy level. Clustering in this phase is like a reactive method in order to sustain connectivity of entire network , prevent single-point failure and future replacement of exhausted nodes.

1.3.1   Literature Review
This chapter presents a brief review of three models which played the role of prototypes to build our routing approach.

A. Direct Diffusion
DD is a data-centric mechanism. This mechanism has been inspired by environmental monitoring implementations. In this approach data messages travel on the sensor network between neighbor nodes but having a single direction to Sink. In the initial state of network, there is a query travelling (in many approaches is called Interest) and in the final state is the formation of a tree-based network. As the query travels from one node to the next neighbor, there is a data dissemination forming a tree-like network. The root of the tree-like network is the sink. Afterwards intermediate nodes between sink and source, exchange data messages . Obviously, there is a goal to find optimal paths for data propagation. To achieve shortest paths, PC was proposed in many research papers and also gradients are kept in cache memory of nodes. Gradients are a pair of attributes, showing also the source that a specific Interest came from. More specifically, gradient is a data structure consists of a) the Node the gradient corresponds to and b) the Source Node (ID of Node) that last transmitted the Interest to the corresponding node. So, is a pair of two fields, ID of the particular node and the neighbor node that has sent to it the Interest. E.g. Node C, Node A.
There are three main types of data in DD approach. Interests, Gradients and Data Messages. In the following paragraphs there will be an extensive explanation about the role and the phase of algorithm those data messages are exchanged.
In [3], which is a clustering proposed system, authors highlight that energy dissipated can be reduced if we set clusters in network topology. In case of flooding possible collisions may occur [3], and data redundancy may cause delay in message transmission.
In [4], Zuhal Can and Murat Demirbas, present a survey about "query and tracking services on wireless sensor networks". They notice that in DD every node keeps in its cache memory a gradient value. When a node receives a query (Interest), starts a comparison between if there is a match on the query received and the data can sense.

In other words, if the query asks a question like "How much degree of humidity there is in section area with coordinators x and y?" node starts to make a match if can sense environmental parameters on section x and y. If there is a match (like can sense data and query concerns this node), makes a data message and forward this to next neighbor.

Here there is an important point. The node will not just forward the data message to next node. Data message follows the reversal path that query followed to reach this node. Node checks if it has a gradient value and send this data to the next node indicated by gradient kept in cache [3].

In the previous mechanism there is a presentation on how a query is flooding in a WSN to reach source node and the way data travel after receiving a query message.

Now there is a presentation of how sink initiates and broadcast messages. Sink floods the network with query messages. These query messages are actually "attributes-value pairs", which means that describe the kind of data are requested .For example in what geographical area, in what time period and many more. Sink keeps sending periodically queries to nodes.

In the next step, nodes sense and propagate data, in the manner described previously(checking and matching). When the sink receives data messages by nodes, there are many paths created in network so far. Paths that data followed to reach sink have as starting point the nodes. The sink is the component of the network that will decide which path can be used as optimal. In other words, sink sends a path reinforcement message and indicate to nodes which path data messages must follow. Until now, a path has been created for data exchanging between source, nodes, and the root of network which is the sink.

In the following states of DD, the sink will accept data from other paths. If there is a better path, and data traveled with reduced delay time for example, sink will update this path, will send a negative message and the procedure for updating gradient values on each node will be repeated. So to conclude, after creating an optimal path chosen by sink, based on gradient values of intermediate nodes between source and sink, there are updates about using alternate paths, especially when a better path has been found. Sink will inform nodes to update gradients and a new path will be formed.

In Fig. 1.1 there are the three basic states in DD. Figure is included in [4].
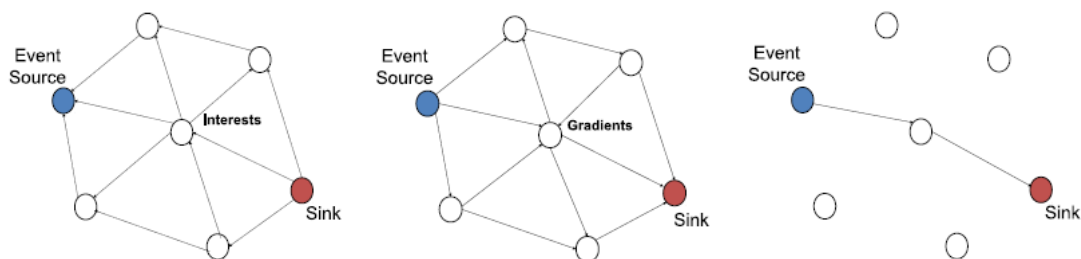
Figure 1.1 States of query flooding, data propagation and path reinforcement to sink

B. Gradient Broadcast Approach

In case of mobile sink, there is another approach called GRAB[5].In GRAB, mobility of sink and source can be managed by setting a new target source each time a new node has been relocated and inserts into network area.

Gradient Broadcast, is a similar querying service, but the main difference between DD is the "cost advertisement packets". Before data dissemination begins, sink sends cost value packets, and the intermediate nodes to forward data to source choose the least cost path. This is a technique to add control in forwarding mesh propagation[3].

Chapter 1 Introduction 20

C. Passive Clustering

PC mechanism is based on the independence of using specific control packets. In the conventional clustering there are many information messages exchanging in order to find neighborhood of each node. In [4] authors claim that if k is the minimum node degree then to collect data and form clusters in a network has O(k) complexity. As WSN are large networks with many sensor nodes , in certain time periods a subset of sensors has to collect and propagate data. Without clustering mechanism , the cost of communication is much more larger. In a wireless ad hoc network the number of data messages exchanging is much more reduced if there is cluster topology. Considering the number of sensors, without grouping nodes, data may have more possibility to have collisions and redundant transmissions will occur.

PC is a mechanism which can be described as an **on-demand** technique. This means that there is no need to form clusters and maintain these groups with large information message exchanging in the begging of a routing protocol. So, there is avoidance of time period of set up period and redundant advertisement messages between nodes. For example, when a Cluster Head must be elected many algorithms, are based to many message exchanging depending on position and connectivity with nodes.

In PC, this procedure can be conducted after flooding query from sink to source(Interest). While query has travelled through all nodes, every intermediate node will have in cache memory the Event that can sense. Forming clusters after this phase, will be more effective.

The reasons are:

1. Clusters between nodes will have same Event.
2. Messages after initial state of network will be reduced because more nodes will play the role of source node transmitters (CHs).
3. Having the same nodes on the optimal path used for a long time period is not wise. Gradients in each node after clusters will be updated and include alternate paths to reach source. Using more frequently same nodes in same backbone of the network is on contrary with limited battery life sensor have.

In other words, using for a long time the same reinforcement path will cause to nodes included in it , energy exhaustion and limited network robustness. In case of network failure or single-point failure on this route, there is more difficult to recover and many messages will be lost.

To conclude, PC can be combined with a data-centric routing protocol like DD, because DD can aggregate and reduce data from multiple sources to one destination and in combination with PC , can limit "channel contention" as scientists refer in [4]. Using a more distributed solution, we can maintain size of network with less communication cost, there is less energy consumption and energy techniques like sleep and awake periods can make sensors prolong their lifetime as a future perspective.Of course, network performance is directly and deeply related with energy consumption in WSN due to architecture of sensors have like limited size, limited processing power and limited memory.

1.3.2 Analysis and Investigation

In this section we will discuss about "weak" and "strong" points on previous techniques. This explains the main idea of our approach and the open issues that this dissertation tries to solve.

5.2.1. Direct Diffusion

Strong & Weak points: DD is a technique where we have large system robustness. Although data flooding may increase data redundancy and network overhead, there are many routes for data messages to travel (limited data loss probability) and system can recover quickly when network and node failures occur[3]. Moreover, data aggregation and cache memory allow improvements in performance than multicast-tree structures in WSN.

On the other hand, data flooding in network will be increased proportional with increasing number of nodes. Moreover, frequent updates of gradient values on each node and usage of same reinforcement path will cause energy exhaustion on nodes in the specific path.

To conclude, another weak point of DD is the static target of source and sink, no scalability characteristic.

5.2.2. Gradient Broadcast Approach

Strong & Weak points: Here data choose intermediate nodes to reach destination of a packet. Additional cost advertisement packets contributes to raising energy cost.

Moreover, this system tends to set a node with low cost to forward larger number of messages than nodes with higher cost which do not forward lots of traffic load. The elected nodes tend to waste more energy in less time.

5.2.3. Passive Clustering

Strong & Weak points: PC is a reactive method implemented here. Clustering is done after Interest flooding of network and it is based on the specific event, the specific query that has inserted into the WSN.

On the other hand, PC is done by additional headers which increase message length. Filtering of messages adds more latency especially in case where a great amount of messages have to be filtered. Finally, election of CHs is based to a *first declaration wins* announcement letting other nodes, that may are low-cost in transmitting data, unselected and consequently unused in an efficient way.

1.3.3. Prototyping

A prototype is built on four basic concepts:

1. More balanced energy consumption by setting location of node as parameter in clustering process.
2. Election of CHs farther from Sink to avoid Hot-Spot Problem.
3. Reduction of Duplicated messages. Flooding is a costly process, Interest propagation happens in the beginning of routing but afterwards, following Interest propagations happens after cluster formation.
4. Prolong of network lifetime by changing the role of CH to CM when it is required.

Our approach has the following phases:

Phase 1: Transmission of Interests-Initialization and Preparing Network

During this phase a query (an Interest message) floods the network. Source floods network and every node transmit this query to all its neighbors. Event tables on cache memory of each node are filled. Information of topology is initialized for next phase.

Phase 2: Forming Clusters

Cluster Heads and Cluster Members are set. Election is done by three factors, residual energy, distance to sink and node degree.

Phase 3: Set up Gradients-Start Communication

Node set their gradients, pointing to the previous node from which they accepted Interest.

Phase 4: Communication Phase-Updating Gradients and Making exploratory data When all gradients have been set up, exploratory data start travel until they reach Sink.

Phase 5: Path Reinforcement-Finding Reverse Path

When Sink gets exploratory data for a specific event, can start the reverse path to communicate with Source. In this phase optimal path is the path from Sink to Source through cluster heads and ordinary nodes.

In next chapter there is an in-depth description on processing and data communication during these phases.

1.3.3 System Evaluation

The DDRA is based on PC and DD. Our approach borrows advantages from DD and Clustering method. DD is used as an on-demand request routing to gather data on application layer. On the other hand, significance of PC is to build energy-efficient paths in network communication.

In DDRA there are advantages on delay metric. As our solution refer to static sink, we managed to get better end-to-end delay compared with DD. In addition, as the network size increase, the energy consumption tends to decrease similarly. More modules participate in clusters, more paths are created. This means that data transmission does not follow a centralized path, which will spent energy of involved modules. There are many routes for data propagation.

To add, traffic patterns as multiple sinks and sources contribute to a more balanced energy consumption. When the number of sinks and sources increase, "hot spot" problem does not occur in two specific points (near single sink and single source). There are more locations and modules of network topology that accept greater amount of data and network overload is shared. This is a distributed solution that eliminates energy wastage.

Although many results reveal that energy consumption ranges in high level in the smallest network size case scenario . Modules of system tend to have unbalanced remaining energy level and more specifically, a set of nodes happens to be "unfair" used. This leads to nodes loss very quickly a significant amount of energy but other nodes maintain at their high energy resources in the end of simulation. Another issue is the network lifetime. Unfortunately, energy wastage contributes to have a link-failure and disconnection between source and sink. This leads to unsuccessful maintenance of system and connectivity between source and sink is not reliable and scalable.

1.4 Novel Features of The Thesis

Generally the DDRA for WSN is an effort to combine methods for energy-efficient routing.

Our novelty features are the according:

Combination of three approaches on WSN architecture

- "Energy Balancing and Hierarchical Clustering based Routing algorithm" (EBHCR) protocol on CH election scheme
- Passive Clustering
- Direct Diffusion

Our main novelty features of related work can be summarized as following:

1. Source floods network with Interest and not sink as in DD occurs .
2. There is no filter on messages and additional headers for messages.
3. Cluster Head election is a hybrid process and not proactive process.
4. There is no "*first declaration wins*" for CH. Clusters are formed based on three factors, energy level, network topology and node degree of each sensor.
5. Nodes play the role of CHs but when energy level drops of a threshold, cluster members compete each other (based on election scheme) to announce themselves as CHs.
6. There are no gateways, only clusters.
7. Usage of distance between sensors and final destination (sink) helps to avoid bottlenecks near crucial points like sink and source.
8. Setting maximum number of hops in a cluster, helps to avoid energy wastage for remote nodes. There is no varying cluster size. All clusters have same fixed maximum hops.

1.5 Outline of the Thesis

The structure of thesis is the following:

Chapter 1: Introduction. This part is about the general idea of this approach. This is an overview about concept of DDRA for WSN. Motivation refers to the main idea that inspired this research and research methodology and research methodology includes previous approaches on routing algorithms for WSN. Moreover, there is analysis about the prototype which has build, overview of simulation system and general idea about system evaluation.

Finally, we discuss in summary the novelty of our approach.

Chapter 2: Literature Review. This chapter is dedicated to related work on routing algorithms for WSN. This is the background that DDRA is based and there is a discussion of main problems we tried to solve.

Chapter 3: Proposed System. This part is about our implementation. There is the theoretical approach of algorithm. Comparison between other approaches, advantages and disadvantages of previous related work are discussed and there is a presentation of the phases of DDRA.

Chapter 4: Description of Proposed System-Energy model and Channel Attributes. This part is about energy consumption of our system, attributes of channel and description of messages we use. This part precedes the next chapter that describes the methodology choice of our simulation system.

Chapter 5: Methodology Choice. Our simulation model is presented in this section. Regarding we describe the network with 12 sensors and we suppose network with 32 and network with 64 nodes are developed in same fashion. There is an extensive discussion about techniques we use and generally, is the detailed presentation of developed simulation model. Discussion for each procedure is included. References to Appendix A show the methods and procedure we used in our systems.

Chapter 6: Analysis of Simulation Model. Analyzing the proposed system is done in many ways. We analyze the network performance, the communication phase according graphical interface and justify the functionality of model.

Chapter 7: Result Discussion. Here we analyze the evaluation criteria on network performance.

Chapter 8: Conclusions. This is the main part for describing the entire network performance and present conclusions on design of DDRA. Evaluation of system is the main point of this chapter. Here there is a presentation of Future Work about more research on our topic and potential usage of this thesis. In addition, ethical issues are included stemmed by applications based on DDRA.

Chapter 2

Literature Review

This chapter refers to the related work and solutions based on Direct Diffusion and Clustering Method.

2.1 Introduction

As described above many routing approaches are based either on location on nodes or their energy resources.WSNs are although networks consist of limited capability devices and consequently the most researches and routing approaches has as main goal the energy consumption. Our approach has the same objective.

In this part there is a presentation of traditional and recent routing proposals about WSN.

Table 2.1 shows a comparison between these approaches.

2.2 Overview on Clustering Algorithms

Sensors are devices limited in computing and storage capability. Also, especially, in large scale measurement of environments it is infeasible to be recharged. Clustering is a technique which offers scalability ,high energy efficiency and extension of battery life of sensors. As Basilis Mamalis et al. discuss in [6], clusters lead to a two-level hierarchy where Cluster Heads consist the higher level. The lower level is the nodes that are cluster members. The leaders of clusters, the CHs are the responsible nodes for data fusion and aggregation. Also periodically transmission of messages leads to significant energy savings. The main reason for using hierarchical routing protocols is *"scalability, load balancing and efficient routing utilization"*, as authors in [6]mention about clusters in WSNs.

The main objectives of using hierarchy and data fusion in clusters are:

Load Balancing

When a CH is selected among other sensors, and the device is the same with the other sensors, so they have same energy and processing power, it is more wise to balance the load of data exchanging. A CH has the duty of intra-cluster communication, but also has to forward results to the base station. An idea is to balance the number of sensors in each cluster and so, all clusters will have same size. This helps us implement a more "fair" implementation and an equal load sharing. This leads to synchronization between the CHs of the system[7].

 Fault Tolerance

In many applications, WSN operate in harsh environments. In the case that a CH fails, there must be re-clustering of network. This maybe is expensive for overall operation. To prevent such loss , there can be a backup CH. This CH can be in the radio range of the initial CH. In techniques where the role of CH is rotating among the nodes, there is a fault-tolerance and failures can easily overcome by quick replacement[7].

 Increased connectivity and reduced delay

A basic characteristic in the cluster is the connectivity. The main goal about connection of all components is either every path can reach from the CHs to the base station

or to reduce the length of those paths. We want to limit the path to the sensors. In other words, we want to set a constraint on the upper limit of hops. In contrast, when we are interested in the data latency we set limits in the intra-cluster communication (communication between nodes in the same cluster). We can set a maximum k hops to communicate a CH with a member of a cluster[7].

Minimal cluster count

Obviously, we do not want a system with large number of CH, because this is in contrast with the distributed design of WSN. As we want CH be rich-resource devices, we try to limit the number of clusters. Less CHs will communicate more effectively with base station, but we must not choose a large number of nodes in a cluster. By this way, the CH will soon have overhead and their energy will be exhausted at short time[6]. There must be a balance between number of clusters to optimize communication between CHs and communication between members of a cluster.

Maximum network longevity

A node can be set as CH if it has enough energy resources to communicate with intra-cluster and inter-cluster section of network. Moreover, the required energy to reach long distance sensors in a cluster is more than communicate with closest entities. We can combine low-energy techniques and effective location schemes for CH, to achieve longer duration of lifetime of these devices. For example, we can use "awake" and "sleep" mode for signal transmission in CHs and we can choose a CH that is more close to others sensors of cluster to reduce energy consume to reach them. (in the center of cluster[7]).

In figure 2.1 we can see the architecture of cluster system [6]. There is a description about inter-cluster communication and intra-cluster communication, one of the basic function in a routing protocol forming two-tier architecture.
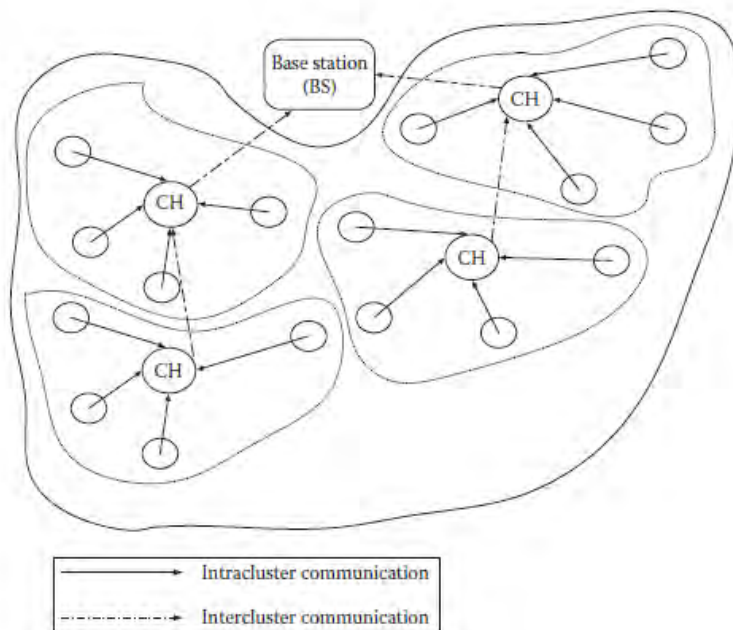
Figure 2.1 Inter –cluster /Intra- cluster communication

2.3 Clustering Algorithms in Wireless Sensor Networks

The cluster algorithms can be distinguished into two main categories according convergence time[7]. This is about O(n), n is the number of nodes. Variable convergence time are algorithms that the time to finish depends on the network size and constant convergence time refers to the category of algorithms that have certain iterations and number of nodes is an independent factor.

In [7], "A *survey on clustering algorithms for wireless sensor networks*" by *Ameer Ahmed Abbasi et al.* authors present the basic cluster algorithms for wireless sensor networks.

2.3.1 Variable Convergence Time Algorithms

1. Linked Cluster Algorithm (LCA) :D.J. Baker and A. Ephremides developed early an approach in the wireless networks(1984)[8]. Here they focused on a topology that can handle mobility of devices. CHs perform a backbone where the members can connect any time even when they are on the move. The main idea is to make CHs connect directly with all members of cluster. LCA is based in maximizing the network connectivity. In the first stage, every node broadcast in a time slot its ID. It listens to all other nodes. In the next round every node knows 1-hop and 2-hop neighbors. The node with highest ID (with more neighbors) become a cluster head.

Open Issues: The LCA approach results to large number of clusters and this is not energy efficient solution. Moreover, the cluster stability is moderate. This can be translated as the number of clusters that algorithm produce may vary according sample data and clustering parameters.

2. Adaptive clustering: Lin and Gerla in [9]proposed another suggestion to support multimedia applications. Here they managed to decrease data delivery delay by clustering. As authors claim " a distinct code is assigned to the cluster". Here we have too, a single-hop intra-cluster topology, as in the previous algorithm (LCA). By reusing the channels, we have small number of clusters. In addition, they reduce data delivery delay by avoiding inter-cluster routing. This means that they use large size clusters and the CH chooses arbitrary the communication code with neighbor CHs. Time Division Medium Access is used for intra-cluster communication[7].

Open Issues: Adaptive clustering is based on distinct code. In every cluster there is a reuse of channels. This is repeated often because the cluster size is small and we have many clusters in the implementation.

3. Random Competition based Clustering (RCC): In [10] the proposed algorithm is for mobile ad hoc networks but can be adopted by WSN. The main idea here is to maintain clusters stable to support mobile nodes. Here there is the *First Declaration Wins* rule. Any node that has the other nodes in its range it becomes the CH if it is the first one declares that. So the first node broadcasts its role of being CH. The other neighbor nodes, hear this claim and join the cluster. The others will not nominate themselves to be CHs and they know the CH of their cluster. Neighbor nodes can broadcast at the same time CH claim packet and collision can be created. To avoid previous situation, every node has a random time value. This value decreases every time the node wants to broadcast claim packet. If time value is not zero and has received a claim packet from another node, it stops the time value, consequently stops the effort to transmit this type of packet to others.

Random time value is not a complete solution for conflicts, so there is an ID for each node. If two nodes try to broadcast concurrently the same time this packet, the node

with lowest ID will become CH. The main difference from above approaches is that in adaptive clustering a CH gives the role of CH and become itself CM to another with lower ID. In this algorithm (RCC) CH gives up its role when another CH comes near to it[7].

Open Issues: In this proposal the main disadvantage is the node mobility. The role of the CH is changed frequently as it easily abandons its role when a new node join cluster. This is critical point for a WSN system, especially when mobility is high.

4. CLUBS: Nagpal and Coore in [11] proposed a different approach. Here the clusters are formed by local broadcast and convergence. The time period to form these clusters is analogous to the density of the nodes. Three main features dominate here:

- Every node must be a part of a cluster.
- Maximum diameter of all clusters must be equal for all clusters. Diameter is the maximum distance of any two points in a cluster.
- Clusters should support the intra-cluster communication (any node in the cluster can communicate with all other nodes in the cluster).

The methodology which they used had the following steps:

1) There are maximum 2-hops in each cluster.

2) Every node picks randomly a number from a fixed integer range.

3) Silently counts down until it reaches zero.

4) If a node can reach zero and has not received a recruit message, becomes a CH and broadcast the "recruit" message to the neighbors with maximum 2-hops.

5) When a node receives such a message, stops counting and accepts "recruit" and joins the cluster.

6) A node which just joined the cluster is called "follower" and can not compete anymore to be CH.

Open Issues: Advantages of CLUBS are that can be adopted in asynchronous environments, with no knowledge about topology. Major problem, although is when CHs are 1-hop range away from one to another. In this case, CHs must restart the election of cluster head and many clusters will collapse[6].

5. Hierarchical control clustering: Here we have a multi-tier hierarchy in clusters by Banerjee and Khuller. In the [12] based on some attributes of clusters like cluster size, degree of overlap and hierarchy, groups are made. In the proposed scheme, any node can start the process of clustering formation. The node with the minimum ID will be first (will be given priority)if there are many nodes starting this at the same time. We have two phases: Tree Discovery and Cluster Formation.

Tree Discovery: A distributed formation of Breadth-First-Search (BFS) tree. Each node ,u, broadcast every p units of time, message about its shortest hop-distance to the root, r. A node, c, will become child of neighbor node u (and node u will become its parent) if the route to the node r is shortest. After this process, the BFS is no longer so important, and the major role has moved to the information about the cluster information [7]. This technique is suitable for dynamic environments with mobile nodes.

Open Issues: This approach is suitable for mobile nodes but we have an extra overhead since any node can start formation of clusters and the data delivery delay are increased in Tree Discovery phase. There are too many packets transmitted in order to construct the tree.

6. GS³: In [13] Zhang and Arora, think that we must measure and take account the geographical boundary of clusters. They managed to give a solution that creates a wireless network into a cellular hexagon structure. To define the measure of the geo-

metric size they have a radius of the circle. If we have a large radius, this increases energy consumption for intra-cluster communication. Besides that, we have two types of nodes, big and small. The big ones initiate the process of forming clusters and they introduce the small nodes to the others and can be middle points for communication between them. This hexagon structure is virtual and the area is divided into cells equal of radius R. One of the big nodes starts by being the CH and selects the CHs of the neighboring cells. The neighbor CHs select their neighbor CHs and so on. The nodes that are not selected, become members of cluster. This happens until no more "unselected" node exists. GS^3 differs significantly from other approaches because here the CH relocate their center to form a hexagon and uses geographical elements like radius. It can be characterized as self-healing and though it is working in static and dynamic networks [6]. In figure 2.2 we can see the structure of the system in GS^3 algorithm [7] .
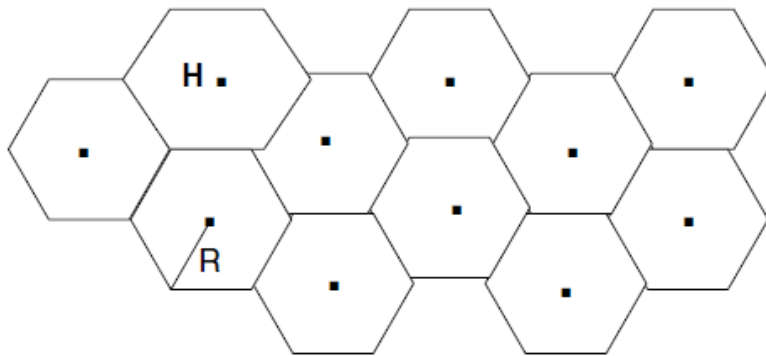
Figure 2.2 Cellular hexagon structure of wireless network (GS3)

Open Issues: One major problem here is that the location of the nodes must be known and this algorithm is not so adaptive in frequent changes like mobility and failure of a node. Moreover, since here authors use geographical radius, this may cause large intra-cluster links, which may decrease the propagation speed of messages.

7. Energy Efficient Hierarchical Clustering (EEHC): This is the most interesting algorithm because here we have a combination of energy and hierarchy for clusters. Our development will be based in energy saving and hierarchy is an attribute which adds distribution character and minimized overhead in data exchanging. We will present the propose of *Bandyopadhyay* and *Coyle* which took place in 2003 at the *22th Annual Conference of IEEE Communication Society.*

First of all, this approach is for WSN. The objective is to maximize network lifetime. This is crucial for sensors since they are small devices and at least in our implementation their energy resource will be batteries or any other energy source but not plugged-in devices. Furthermore, our system is about random deployed sensors so the design limits our implementation to set in proactive manner clustering information[14].

Here the CHs of all clusters read the message of every member of the cluster and send a report to the base-station. Now there are two stages: initial and extended. In initial stage nodes can announce themselves as CH with p probability and a node that receives this announcement becomes immediately member of the closest cluster.

On the other hand, in extended stage, we have multi-level clustering. Like Hierarchical Control Clustering, the formation of clusters is repeated until CHs form an entire block . Here, also we have h-hop connectivity between the CHs and the base-station, as h is the highest distance we can have in the system [7].

The energy consumption depends on probability p and k hops of maximum member distance from CHs of algorithm. Energy will be reduced for functions like sensor data collection and aggregated transmission to base station. The authors use mathematical expressions and involve stochastic geometry to estimate communication energy. With simulation results they prove that by using optimal parameter values for p and k, there is a significant reduction in energy consumption[7].

Open Issues: In this algorithm have a complexity of time O(k1+k2+k3+…Kn), feature which gives gain in comparison with previous ones. The energy consumption depends on the probability of selecting a CH and the k number of hops, maximum number of hops in a cluster. Although the energy is reduced significantly only when the p and k parameters are chosen in optimal way, something which makes approach not so flexible for many types of networks.

2.3.2 Constant Convergence time algorithms

The principal of constant convergence time is that they have a point of convergence after a certain number of iterations. This is true despite the number of nodes. Here each node execute the algorithm independent from all others and takes its own decision about being a member of a cluster or CH according its state and neighbors state[7].

In this part we will present some basic algorithms that belong in constant convergence time category.

1. (Low Energy Adaptive Clustering Hierarchy) LEACH: [15] Very popular algorithm for WSN. The formation of clusters depends on the signal strength and the CHs play the role of routers to the base station. Each cluster is responsible for data processing.

Open Issues: The main disadvantages of this approach is that even nodes having low energy power can be elected as CH, therefore if a sensor "dies", a CH "dies", the cluster can not operate with the others. In other words, the system does not work efficiently and is not fault-tolerant [7].

The CH must reach base-station. Although in dynamic environments, the CH can be far away from base-station. This can cause transmission errors (e.g. by obstacles) or additional overhead because we have a multi-hop communication between base-station and all CHs of system.

This solution is ideal for nodes that can communicate directly with CH of their group. But, what happens if members are farther that 1-hop from the CH? Another solution must be invistigated to include this case. Case which is common in mobile networks[16].

2. Hybrid Energy-Efficient Distributed Clustering (HEED): [17]In this algorithm the CH is elected by the other nodes. Scientists consider an energy and communication cost for the selection process of CH. On the contrary of LEACH, here CH are not random nodes, but they choose the nodes having high residual energy. Three features can be highlighted:

1. Two nodes that are close in location, consequently are in the same transmission range have less probability to become CHs[7].
2. Energy level on each sensor is not equal, so the "rich" sensors, in terms of power, are treated in different way and have different roles that others (more "poor" in energy resources).
3. The probability of sensor to become CH, can be adjusted based on its transmission range to ensure communication for all members in the cluster (intra-cluster communication).Each node is mapped to one cluster and can directly communicate with the CH.

Huang and Wu in [18] changed slightly the operation of HEED algorithm. They introduced the term of "orphaned nodes". These are nodes that didn't hear any announcement from any other node. In the HEED algorithm, at Finalization stage they become CH themselves. Here another option is to re-execute the steps and resulted in having less CHs in the system. This means that the routing trees are limited, so the communication cost and the data transfer latency can be reduced[18].

Open Issues: In HEED we have to re-execute the algorithm for the nodes that didn't hear about the announcement of any CH. Moreover, we would like to reduce the route

cost (number of hops in the routing tree) and so it would be more effective to minimize the hops in the inter-cluster communication. Those two proposals were made by Huang and Wu in [18].

3. Distributed Weight-Based Energy Efficient Hierarchical Clustering (DWEHC) : Another interesting algorithm, based on the idea of HEED is DWEHC. Ding et al. in [19] achieved to balance the size of clusters and construct a better " intra-cluster topology"[6].The process contains the following steps:

2.3.1 Each sensor calculates the weight it has, after the knowledge of neighbor nodes position. The weight is proportional to sensor's energy reserve and the distance between the neighbors. So the topology is like a neighborhood. Close nodes calculate their weight. The largest weight node can become a CH and remaining nodes can be members. First-level members are the nodes that have direct link with CH. The nodes must adjust their membership to consume less energy for communication.

2.3.2 The node checks about non-neighbors CHs to find out the minimal cost for reaching a CH. Here the nodes must decide if it is better to be first-level members or second-level members. In other words a node finds the least cost path to a direct link to CH or a 2-hop link to this CH.

2.3.3 The process of counting minimal cost on routing for the CH stops when all nodes have the most "energy –efficient intra-cluster topology". We do not want a multi-level path, so authors suggested a range for each cluster to "recruit" member nodes[6].

Open Issues: Both in DWEHC and HEED the most important criteria for selecting CH is the energy reserve but it is important to count on the size of the network. Actually, the size affects directly delay and throughput, the data collection latency and the overall energy consumption of the system.

2.3.3. Hierarchal Algorithms in WSN
In hierarchical protocols one of the major design attribute is scalability. In a flat connection of nodes the gateway will have extra overload as the number of sensors increases. This may cause latency in communication and inadequate tracking of events. To allow new connections and coverage of larger areas, clustering and hierarchy can introduced. In the section above we will present algorithms based on multi-level hierarchy.

1. PEGASIS
Power-Efficient GAthering in Sensor Information Systems (PEGASIS) is an improvement of LEACH protocol[20]. The main idea here is not have many clusters but chain of nodes. Every node transmits to its neighbor and receives packets from a neighbor too. Only one node can transmit finally to the sink. The data are aggregated to every node-"station" and a chain in communication is performed. The main difference from LEACH is multi-hop routing by chaining and transmission by only one node[21].
PEGASIS worked successfully to many different network sizes and this is achieved because overhead of performing multiple clusters is absent.

Open Issues: It is obvious that for very long chains there will be much delay in packets transmissions. In addition, the single node may be a bottleneck, as all traffic is merged in this point.

2. Hierarchical-PEGASIS

This extension of PEGASIS try to solve the problem of delay in transmission from node to base station. Takes account the energy and the delay metrics of network. The main idea is to transmit simultaneously either by CDMA or spatial nodes are allowed to transmit [22].

In the CDMA approach there is a tree like hierarchy and each level consists of a chain of nodes. At each level one single node can transmit to upper level so we can have simultaneous transmissions.

In the spatial node communication approach there is a three-level hierarchy. Collisions and interference are avoided by scheduling packet delivery[21].

Open Issues: This solution may improve PEGASIS performance but sensor's energy is not tracked so each node can not be informed about the status of the neighboring nodes. Consequently, the routing sometimes can not be established. Eventually, overhead is introduced during the communication.

3. TEEN

Threshold sensitive Energy Efficient sensor Network protocol (TEEN) is a hierarchical protocol designed to be sensitive to changes of the environment attributes that are sensed by sensors, for example the temperature[23]. WSN must be responsive to changes and they act in reactive mode. The sensor network architecture has clusters and this continues to second level and upper levels until the base station can be reached.

After the cluster formation, the CHs broadcast two thresholds to the nodes, the "hard" and the "soft" threshold. "Hard" is the minimum value that can trigger an action in the system and the "soft" is the maximum value of the sensed attribute. Data are transmitted only when the value is equal or greater than the hard threshold or transmit if the the value is equal or lower than the soft threshold. Using this mechanism the transmissions are reduced and sensors are activated for a certain range of interest. Redundancy of data is also eliminated.

Open Issues: This is not proper for systems where periodic reports must be done. If the values are not in the range, maybe we can have no transmissions at all[21].

4. APTEEN

Adaptive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) [24]is a variation of previous algorithm. Here we have a combination data periodic reports and time-critical events. The base station forms the clusters. Then cluster heads broadcast the attributes, the threshold values and the transmission schedule for all the nodes. There are three types of queries: historical, where there is analysis of past values, one-time, where we have a snapshot view of network and persistent to observe an event for a duration of time.

APTEEN and TEEN excess in performance the LEACH algorithm.

Open Issues: Although there is an important elimination of packets transmissions and delays created by them, there is much complexity considering the effort to form clusters in multiple levels, having threshold-based functions and queries relative to attribute-naming

5. Energy-aware routing for cluster-based sensor networks

Younis et al. in [25] proposed a three-tier architecture. The clusters are formed before network communication. The CHs act like gateways are more powerful than ordinary nodes of the system. The CHs set up multi-hop routes for collecting data of sensors. The sink can be connected only with the cluster heads.

Here each node has to be informed by its cluster head. There are certain time slots when a sensor can listen to other's transmissions and slots when the sensor can transmit its signals.

The sensor can have two power modes, the active one and the stand-by mode. Moreover, there are four states about each sensor. Sensing only, where sensor probe environment and generate data at constant rate. Relaying only, where can only transmit from other active sensors .No sense data. Finally, "save- energy" mode. In Sensing and Relaying state, sensor can both sense and relay data. In the fourth state, sensor is in inactive state where turns off its transmitter and receiver.

This proposal achieves expansion of network lifetime as energy-based metrics are considered. Also, time slots are combined with states of sensors, improving throughput and end-to-end delays.

Open Issues: Simple propagation model can fit in such conditions. In contrast, many gateways may be required in order to cover large distance networks and high density of sensors.

6. Self-Organizing protocol

Another effort on hierarchy in WSN comes from Subramanian and Katz in [26], on which self organization and taxonomy of applications are the critical points. There are two kinds of sensors, mobile and stationary. The stationary nodes are the "routers" and perform a backbone on the system. The others, collect and transmit data to the routers. There are four phases to execute and build routing tables in this algorithm:

Discovery Phase: nodes in proximity are discovered.

Organization Phase: There are groups and hierarchy. Each node has an address according the position in the hierarchy. For every node we have a routing table.

Maintenance Phase: It is necessary to update the routing tables and the energy level of nodes and it is done during this phase. The node sends its status to the neighbor node and the broadcast trees are reconstructed.

Self-reorganization Phase: When a node fails or depletes, reorganization reconstructs the network system[21].

This approach keeps all sensors as set and a limited subset of nodes are used. This allows energy saving. The addressing of individual nodes allows communication with a single node. In addition, maintaining routing tables and keeping hierarchy is balanced due to the backbone and addressing scheme.

Open Issues: In the organization phase, which is not on-demand, data exchanging is performed before updating routing tables. Many messages may be forwarded by not optimal route.

2.4 Hierarchical Ad-Hoc Routing
In this section we will review approaches on ad-hoc WSN.

Hierarchical protocols introduce a new technique. Here there is no flat architecture of nodes but we set levels on network communication. We build a hierarchy by separate nodes into levels and typically with clustering techniques we improve scalability and efficiency of routing[27].

1. **Small Minimum Energy Communication Network (MENC-SMECN)**

In [28] authors present the construction of an energy-efficient subnetwork by using low power GPS. The basic motivation of this approach is that in every node, authors, set a relay region containing nodes that it is more efficient to transmit, rather transmit in direct way to neighbors. The enclosure graph of a node p is the union of all regions that node p can reach. Assuming a pair of nodes that can communicate in more efficient way, the objective here is to create a subnet with less number of nodes and less required power energy to transmit between pairs of nodes in the system. By checking the relay regions of all nodes, authors have as result global minimum power paths. To do so, localized search is done on any node [29].

The privileges of this implementation is that can easily reconfigured and self-organized. This system can cope from node failure and join of new devices.

The SMECN is the extension of Minimum Energy Communication MECN in a smaller subgraph. In the latter we assume that each node can transmit to any other node, something impossible under certain circumstances (obstacles, link failures, e.t.c.). In SMECN, the subgraph constructed is smaller, having less edges, but we assume this subnetwork fully connected too.

Considering that G is the graph of all sensors, there are two standards for subnetwork establishment G': 1) G' has less edges than the edges included in the G graph, 2) the energy required between nodes on G' is less than the energy required for all nodes presented in graph G of network [29].

Open Issues: Actually, this is a solution not for the minimum energy path but a usage of a subnetwork providing a less energy cost solution for communication. To emphasize in network performance limitations, the subgraph G' adds more overhead to be constructed in the initial phase, fact that adds delay and time consumption.

2. **Hierarchical Power-aware Routing (HPAR)**

In [30] scientists divide the network into groups. The group of sensors are grouped based on their location, they consist zones and every zone is like an entity. To forward every packet from one zone to another, sensors choose hierarchical routing and choose the route that offers maximum battery life for nodes. The messages are routed along a path which has the maximum power of all minimum remaining power. This is done because if we use nodes having maximum energy remaining might be much more expensive by using nodes that perform transmissions with minimal power consumption. The general motivation for the authors was to present an algorithm in which if they could minimize total power consumption and could maximize total energy remaining in the devices of over all system.

Firstly, by Dijkstra's algorithm we find minimum power consumption path. Secondly, we find a path which maximizes the minimum residual energy and the final path is a combination of these two paths [30].

Open Issues: The algorithm introduces a new idea, the idea of finding routes that may include nodes with largest of minimum energy remaining but will cost less in energy consumption. The basic features of WSN although are node's mobility and the exhaustion of the grouped nodes. Is this solution feasible for a long time period or nodes may easy depleted if they are chosen frequently? The question brings up aspects like lifetime sensor and scalability in network which are not highlighted.

3. **Two-Tier Data Dissemination (TTDD)**

Another point of view is the one presented in the [28]. Here we exploit data delivery to multiple mobile base-stations. Each sensor firstly builds a grid to mobile base stations that are stationary and location-aware. Afterwards, sensors are assumed stationary but sinks can change their position. When a change or event happens, one of the sensor plays the role of the generator and forwards reports to the sinks. To form a grid, the source chooses itself as the starting point and sends announcements to the other four neighbors, in order to have a geographical region. When the message is transmitted to crossing point (message has this information), the message stops to propagate and a piece of grid has been created into this area. These types of messages are propagated until the last message reaches the last crossing point of network, the geographical limits.

In the next phase a query is flooding all the network to destination (sinks) and through all included sensors. As the base stations can move into a new sensor field, the routing tables are updated and defined by the reverse path that query followed previously from sensors to sinks [29].

Open Issues: Setting up the grid seems not an optimal energy-aware approach, besides the forwarding path of query is much more longer than shortest path given by other proposals. In addition, authors may insist on scalability on WSN but the path proposed can not be characterized as optimal either in energy terms or in number of hops. Also authors in [29] indicate there are important issues on flooding method.

Flaws of flooding protocol can be summarized as following:

   I. <u>Implosion</u>: duplicated data sent to same node
   II. <u>Overlap</u>: two nodes in same area, sense and will send similar packets to same neighbor
   III. <u>Blind use of resources</u>: Consume large amount of energy without considering energy resources

Finally, delays are increased from message and query propagations and mobility of sensors is not supported in this situation, something that contributes into a very limited and inflexible solution provided by scientists.

4. Direct Diffusion and Passive Clustering (PCDD)

In [2] Vlado Handziski et al. propose a routing algorithm that combines DD and PC. They try to investigate "how it is feasible to execute directed diffusion on top of a sensor network where the topology is implicitly constructed by passive clustering?"

In their research they support that depending the scenario, PC can contribute to great reduction of energy consuming and delivery time on messages exchanging in the network.

Umesh Rawat et al. in [31] discuss about the technique of DD. DD is a data-centric approach and all sensors are application-aware. The main characteristic is that sink initiates communication by flooding network with a message, called interest. The interest can be query such as "Are there any mobile cars in your region?". This request is transmitted to the next node, the node will transmit it to its neighbors and this task will repeat until query arrives to end point, the sources.

In this section there is a brief description about DD and PC
as authors in [2] describe.

In the initial phase, as in DD, we have flooding routing. This is the Interest that has the role of a query. The sink of the network starts propagate the Interest to all of the neighbors. Then the neighbors propagate this kind of query to their neighbors, too. It is remarkable to highlight that this creates an extra overhead of energy and time [31].

As the interest is flooding on the network, nodes keep track of query. They keep in their cache the type of interest received and the source node, the previous node that propagate this interest. Obviously, some nodes may receive redundant data and collisions may occur.

In the next phase, gradients are set up. Here the sensors will keep in their cache an attribute pair containing a description of interest and a direction. Description is about the type of data requested and direction is the node from which "heard" the interest. In the case it heard the same interest from many neighbors, will set up interest for each of them [2]. So every node will set up gradients to communicate with neighbors having the gradients.

In this phase the nodes will create between close nodes clusters. So this is the stage of cluster formation. The nodes will become cluster heads, if they haven't heard an announcement by other nodes and will be CHs of the cluster. The "listeners nodes" will become cluster members of the first cluster head announcement they heard. This is the "*first declaration wins*" mechanism that writers use in the corresponding research article.

Other solutions suggested was to use hexagon diameter in a constant topology. In this idea sensors will transmit data in hexagonal diameter and change main routes to conserve required energy resources [32].

In PCEEC ("*Passive Clustering for Efficient Energy Conservation in Wireless Sensor Network*") [33] the authors extended more the idea of Vlado Handziski and Andreas K¨opke, and introduce the notion of using alternative cluster heads when the energy level of cluster heads are reduced significantly.

In the network there are three types of nodes, cluster members, cluster heads and gateways. Gateways are the nodes that heard two Cluster Head announcement and also, if there is no other Gateway announcement for these two particular CHs , this node can play the role of gateway.

In order to achieve clustering in the topology, there is a filter on Direct Diffusion Core. This filter has as incoming messages before gradient set up and take out messages from gradients to network. This filter actually adds a header and sets up the state that each node will have (ordinary node, cluster head, gateway).

After this procedure, the sensor will sense the environment and send that data to the neighbors having matching gradients. In this phase, those data are characterized as exploratory data. Intermediate nodes forward this packet until it reaches the sink. In this phase a route has been established from source to sink. Actually, this is the phase of making gradients to the sink. Sink will receive the exploratory data and will reply with a reinforcement message but choosing only the one neighbor that heard first this exploratory data message. In this phase, we have a gradient structure in the network. Source will accept this exploratory data (reinforcement path) and starts sending data to sensors more frequently at certain time periods (e.g. every 2 seconds).

Finally, the reinforcement path begins from source to sink and follows the optimal path indicated by two parameters of the network topology, gradients which are kept

for a certain time period in each node and structure of clusters for inter-communication between sensors and sink.

Open Issues: In this approach obviously, there is an open area subject on how we can elect cluster heads and gateways in a more efficient way. The *first declaration wins* technique could be further replaced by an energy computation mechanism. Another idea is to follow an energy-saving scheme where nodes could alternate between sleep and wake up modes.

In the following chapters there will be an in-depth investigation on how this DD and PC algorithm can be improved by implementing another election of cluster heads based mainly on energy and topology factors.

2.5 Comparison

In the following pages there is a comparison on the previous algorithms presented.

	Protocol	Node Mobility	Cluster Overlapping	Location Awareness	Energy Efficient	Failure Recovery	Balanced Clustering
1.	LCA	Possible	No	Required	No	Yes	OK
2.	Adaptive Clustering	Yes	No	Required	N/A	Yes	OK
3.	RCC	Yes	No	Required	N/A	Yes	Good
4.	GS^3	Possible	Low	Required	N/A	Yes	Good
5.	CLUBS	Possible	High	Not required	N/A	Yes	OK
6.	Hierarchical Control Clustering	Possible	Low	Not required	N/A	Yes	Good
7.	EEHC	No	No	Required	Yes	N/A	OK
8.	LEACH	Fixed BS	No	Not required	No	Yes	OK
9.	HEED	Stationary	No	Not required	Yes	N/A	Good
10.	DWEHC	Stationary	No	Required	Yes	N/A	Very Good
11.	PEGASIS	Fixed BS	No	Required	No	N/A	OK
12.	Hierarchical-PEGASIS	Fixed BS	No	Required	No	N/A	OK
13.	TEEN	Yes	No	Not required	Yes	N/A	OK

	Protocol	Node Mobility	Cluster Overlapping	Location Awareness	Energy Efficient	Failure Recovery	Balanced Clustering
14.	APTEEN	Yes	No	Not required	Yes	N/A	OK
15.	Energy-Aware routing for cluster-based sensor networks	Fixed BS	High	Required	Yes	Yes	Good
16.	Self-Organizing protocol	Stationary and mobile	No	Required	Yes	Yes	Very Good
17.	MECN	No	No	Not Required	Yes	N/A	Good
18.	SMECN	No	No	Not Required	Yes	N/A	Good
19.	HPAR	No	Moderate	Not Required	Yes	N/A	Good
20.	TTDD	Yes	No	Required	Yes	Yes	Very Good
21.	DD &PC	Yes	No	Not Required	Yes	Yes	Good

Table   2.1 Classification of Algorithms based on Clustering Attributes

Chapter 3

Description of DDRA for WSN

This chapter contains the description of DDRA protocol, the main idea that this protocol is based and the phases of this algorithm.

3.1.   Introduction

This chapter analyzes the method and the features of DDRA. In this study we present a network protocol based on DD and PC. This research work is based on PC methodology and reviews on how DD can be improved. Improvements are based in cluster election, message propagations and decrease of duplicated data due to flooding method.

3.2.   State of Art

Here we will discuss two main approaches of PC and DD .The "PCDD protocol" and "Energy Conserving PC for Efficient Routing Protocol (ECPC)". The latter is an approach for reducing the number of Gateways in a system to avoid additional overhead. Moreover, other approaches on DD and PC are presented.

3.2.1. Review of "Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering "-Routing Algorithm for WSN

Vlado Handziski , Andreas K¨opke, Holger Karl, Christian Frank, and Witold Drytkiewicz presented a routing approach while working in the Technology University of Berlin about setting together the idea of DD and PC. This paper is included in the book of title "Wireless Sensor Networks", and having subtitle "First European Workshop, EWSN 2004, Berlin, Germany, January 19-21, 2004".

This research is the main motivation about the algorithm proposed. Although 10 years have passed from this publication, this paper has motivated many others research papers like "Passive Clustering for Efficient Energy Conservation in Wireless Sensor Network (PCEEC)", of Abderrahim MAIZATE et al. presented in 2013 and "Energy Conserving Passive Clustering for Efficient Routing in Wireless Sensor Network" published in 2007 [3]at "The 9th International Conference on Advanced Communication Technology" and written by Md. Mamun-or-Rashid, Muhammad Mahbub Alam and Choong Seon Hong, Kyung Hee University of Korea.

Another idea by Arash Nasiri Eghbali et. al in [34] is to avoid massive data transmissions of control messages (Exploratory Data) and use a virtual sink. This virtual sink will send local Interest messages and afterwards data packets will be routed to destination.

As "Springer" publications refer about this scientist paper is that the contribution of this work is "How it is feasible to execute directed diffusion on top of a sensor network where the topology is implicitly constructed by passive clustering?". In addition, also "Springer" declares that this paper is "A simulation-based comparison between plain directed diffusion and one based on PC shows that, depending on the scenario,

PC can significantly reduce the required energy while maintaining and even improving the delay and the delivery rate." [35]

The first paper describes the concept of DD in the first part. Authors underline that if clustering can be imposed as second parameter in the behavior of DD, with respect to this process they could manage reduce energy consumption, delay of delivery and delivery rate of packets.

The idea to use PC instead of traditional clustering schemes was inspired by the dynamic formation of clusters. Unlike proactive clustering, here the clusters are made by the first data message, and authors achieved to reduce the initial-long period of setting clusters at the beginning of clustering method. After a small number of messages forwarding rounds, we have a distributed network, separated into clusters and sharing the traffic from critical nodes in the reinforcement path.
Authors in [2] use the method of first declaration wins to choose a cluster head and cover a radio range.
As the main duty of clustering is to optimize the exchange of flooding messages and avoid the large amount of controlling messages scientists choose PC to minimize firstly, the waste of energy that occurs when no traffic is on network and waste of energy to maintain the clusters. Dynamically clusters made only on-demand and sensors that only sense and relay data discard some type of messages after clustering formation. More specific, when exploratory data and interest participated at first place to set paths and clusters, there is no need be transmitted by final destination nodes.

In our implementation we choose the next neighbor of cluster head that has the best energy level remaining. Although, this restrict us to use multi-hop communication, manages to extend the network lifetime, the lifetime of cluster.
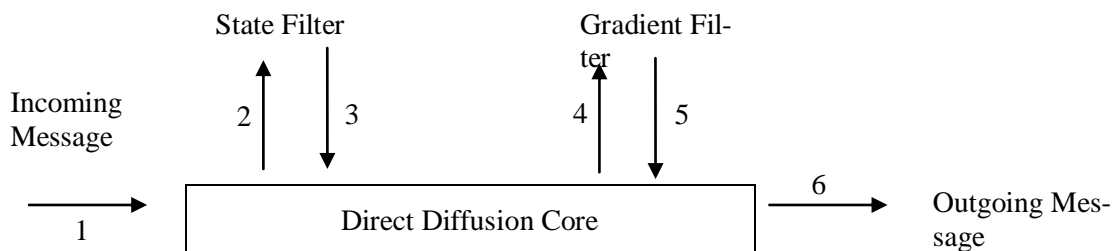
Figure 3. 1 The Direct Diffusion Core

Finally, we conclude that in our proposed system we try too, to use sub-optimal clusters and gradients saved in cache on the nodes, as in PCDD design. This is done to combine sharing load of network and reduce number of messages transmitted to reserve energy and expand network lifetime.

In the Table 3.1 we can see a comparison of state of the each node and the action will be taken according the type of message it receives. This is the result of using Direct Diffusion Core.

State of Node	Interest	Exploratory Data	Local Application
Cluster Head	Forward	Forward	Forward
Gateway	Forward	Forward	Forward
Sink	Generate this	Propagate	Propagate
Ordinary Node	Discard	Forward	Accept
Header Added	No	Yes (if state changed to CH or G)	No
State Changed	No	Yes	No

Table 3.1 State of nodes and Actions in PCDD approach

More about PCDD approach is on headers added through filtering.Below there is a description of Header that will be added in the case that a node accepts a message from network. Only if this message managed to change the external state in the corresponding node, then the following header will be added. If the new state of node or the state in this time is Ordinary Node then the message will be discarded. This technique is introduce to minimize the PC overhead[2].

The fields CH1_ID and CH2_ID are filled only if the state of node is Gateway, so the field G will be updated too. These fields represent the previous and next cluster heads in the backbone of the system.

0 31	34	35	67	99
Node_ID	State	G	CH1_ID	CH2_ID

Identify the node	State of node	Gateway or not	The first cluster head in backbone	The second Cluster Head in backbone

Table 3.2 Passive Clustering Header

Actually the header attaching has a cost of 4 byte and diffusion message ends up with 17 bytes at maximum.

Now we will present the filter in the Direct Diffusion Core that accepts the two types of messages: Network (Interest, Exploratory Data) or Local Application. The types of messages arriving in the Direct Diffusion Core are two: Network and Local Application. The network messages can be Interests (generated from sink and propagated to sources) and the local application messages concern sensing a region and transmitting sensing data to the sink or activate-trigger a sensor [2].

In the outgoing message either a header will be added (network message and state of node changed) or it would be just forwarded to next hop if the message came from local application and has as destination an Ordinary Node.

So we can summarize to the following steps on Filtering and Message Exchange:
1: Reception of message {interest, exploratory data, local application}

2:If message is exploratory data, add header and change state (CH or G)
    If message is Interest or Local application, do not change state
3: Store locally the message and forward to Direct Diffusion Core
4: Send message to Gradient Filter and look for matching neighbor
5: Forward the message to the neighbor having the same Interest
6: Send message to network or Ordinary Node

3.2.2 Structure of the Network in "Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering (PCDD)"

Phase 1: Initial Topology	Phase 2: Cluster topology
Phase 3: Gradients after clustered interest propagated by sink t (Gray circles: Cluster Heads Gray diamonds: Gateways) 	Phase 4: Reinforced paths from sources s to sink t
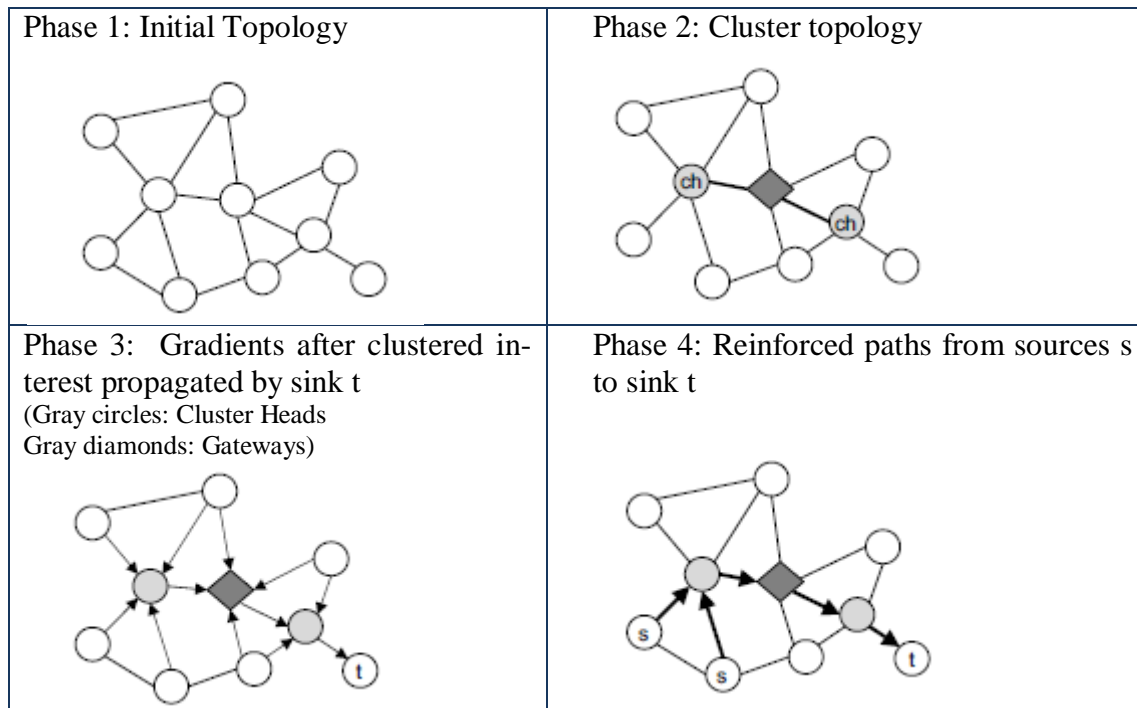
Figure 3. 2 Structure of Network in PCDD protocol

3.3. Evaluation and Discussion

The general idea to use PC instead of forming clusters in the initial state of the network came from extra overhead that classical clustering techniques have. In traditional clusters many messages are exchanged to control clusters and maintain the connectivity in the network. In our solution we combine DD and then we set PC to minimize data exchanging. The local application messages will be transmitted in the clusters and not all nodes have to exchange and forward these types of messages. The backbone introduced will distribute information for the remote cluster members. Additionally, DD tries to maintain gradients on each node, using cache memory but reinforce optimal paths to be followed from source to sink.

In DDRA approach we use an algorithm according the cost of routes to elect the Cluster Heads based on CH probability value, motivated by other election CH schemes. The cost refers to the network metrics about distance from sink and the number of neighbors that a node has in its transmission range. We set clusters at maximum 2 hops from "candidate cluster heads". Moreover, based on unequal clusters theory, we replace a Cluster Head with the first neighbor having greatest energy resources and priority given by topology factors. Obviously, this will cause multi-hop communication. The "hot spot" problem is solved with locally exchanging of messages and there is no need to flood the network with information for updating routing tables, actually change gradients values. The nearest cluster heads will forward and control message exchanging. We expect such a mechanism to reduce the flooding update messages (exploratory data on neighborhood not whole structure of network).

In addition, we have set the ordinary nodes not to propagate interests and exploratory data and clustering method used in not proactive only or reactive only fashion. We use

Chapter 3 Description of DDRA for WSN 48

a hybrid way that operates firstly to assigned to each node a "weight", a probability value to become CH. Later, we have on-demand election scheme, when energy levels on CHs are reduced. This is selected to create paths in flexible way. The idea is to set clusters not only when are needed and not only have a initial heavyweight creation and maintain of clusters, which are independent of network dynamics.

In conclusion, basically we will combine two techniques, DD and PC to implement a new algorithm. The layers that will be involved is the network layer and the application layer of OSI architecture. Network layer due to setting optimal paths and Application layer due to inserting Interests from sink to whole network. In addition, control messages will help to route discovery, route establishment and communication in intra and inter cluster procedures.

3.4. Open Issues of PCDD algorithm

- The first disadvantage of using those mechanisms is that the flooding routing in the initial state of network (forwarding the interest from sink), will cause poor performance in the beginning of network operation.
- As the interest propagates from node to next node, many nodes will accept the same message from neighbors and collisions will reduce throughput.
- In addition, in the phase of constructing the clusters, additional overhead by message exchanging will reduce energy level on each node.
- Furthermore, the backbone of cluster heads and gateways will transmit a large amount of data, causing energy exhaustion in a short time.
- Setting gradients for all neighbors that Interest came, adds unnecessary cost.

3.5. Identification of open problems in Direct Diffusion and Passive Clustering

In this part, there is a discussion about open problems about DD and introduction on several approaches and acts we can do to overcome possible drawbacks and problems in this routing method.

First of all, DD involve flooding of Interest messages in initial state of network. Apparently flooding technique will cause great message redundancy and waste of battery resources in each node. As Sink starts propagate the first Interest message, all receivers of this particular message will start propagate this message to next neighbors connected to them and so on. Finally, Interest message will reach destination, the Source module.

The difficulties in this method are:

- Redundancy of transmitting same message. To be more specific, many identical messages (having same information) will be exchanged even if it is not necessary.
- Acceptance of same message from many preceding nodes, for example if node 4 is connected with node 3 and node 2, both nodes will transmit the message to their neighbor, node 4.

According the number of nodes, this will cause increase of duration of this procedure.

Another point to discuss is the following question "What must be done if the last or middle node is unconnected until message reaches Source?". In the simulation model for example a node is unconnected. This is more important if we have wireless communication and mobility in the network attributes.

Secondly, we must consider the types of messages that will travel in the network.

After transmitting Interest, which is a message injected in network by Sink, nodes will set up Gradients. Gradients will be stored for limited amount of time in cache memory of nodes. When a node hears the same Interest from two or more neighbors, node will set up gradients for each node.

Moreover, gradients must be updated. When a node gets a new Interest, a new Gradient must be set up. In addition, arrivals of new nodes and termination of exhausted nodes force new messages exchanging.

This action will lead to nodes having many neighbors , must set up more gradients , concluding to have more cache memory, update more times gradients and waste more energy resources in these certain nodes.

To add, the Event Table that nodes keep in their cache must be updated too. When a new Interest reaches a node, node tries to match the Event ID of Interest, with the Event ID of its entries in the table. If there is a match ,*Exploratory Data* , begin transmitted depending on the Gradients.

Thirdly, there is a discussion for Exploratory Data. Exploratory Data are messages that are forwarded from nodes to sink. The intermediate nodes forward those messages according their gradients. Eventually, Exploratory Data reaches Sink.

After this, Sink propagate reinforcement path message to last neighbor, from which got Exploratory Data. In same manner, source will get reinforcement path message and this is the event for source to start periodically transmission of data (e.g. every 2 msec).

In this point, there is an increasing amount of data traffic in network. One major question is *"The cost of increased amount of data can compensate the network performance in the final state of network?"* or *"How we can minimize this traffic overhead?"*

One of the approaches, is obvious PC that authors in [2] introduce. They distinguish nodes after Interest flooding into three types: Cluster Head, Gateways and Ordinary Nodes (No-Gateways, No-Cluster Heads).

In [2] there is following theory:

"Ordinary(non-cluster head, non-gateway) nodes do not forward the Interest and exploratory data messages that they receive".

This fact can decrease interaction between nodes and apparently save network communication from large same information exchanging.

Another significant point that authors in [2] propose is that using a *"Diffusion Routing Core"*, they set the state of each node. By having a pre-filter and a post-filter, in incoming and outgoing messages , this core can define the external state of node: Cluster Head, Gateway or Ordinary. To implement this method, there is a PC header.

In contrast, to have an additional header (there is a 4 byte overhead), this policy increases the amount of data.

Another point of view is that gateways will become soon "hot spots" in the network model. As messages will be forwarded through these components, the energy level is expected to decrease very soon.

On the contrary, scientists in [2] declare that using the previous theory of discarding messages of ordinary nodes, can reduce the data redundancy in system.

3.6. Review on " Energy Conserving Passive Clustering for Efficient Routing Protocol (ECPC)"

In Energy Conserving PC for Efficient Routing Protocol (ECPC)[3] authors manage to combine timing before nodes announce themselves as CHs or Gateways. This waiting time depends on distance that node received query by sink and the energy a node has until this time. The nodes that have large amount of energy remaining and short distance from sink will have the chance to wait a short time period and be the first to declare themselves as CHs. Moreover, Gateways will declare themselves in almost the same way, but here waiting time is calculated by average distance a node has from two or more cluster heads in the same transmission range that can hear.

The advantage of this technique is that we avoid to make a large number of gateways, so the redundant transmissions will be reduced and so the network lifetime will be expanded.

The most important aspect on this algorithm is the equation 3.1 that authors propose for the Gateways. In order to have minimum number of candidate Gateways, a node can not declare itself as Gateway if the following equation is true:

$$N(ni(Cid)) = \bigcup_{j=0}^{n} N(nj(N(Cid)) \qquad (3.1)$$

Where ni is node i
$N(ni(Cid))$ are the neighbor Cluster Head IDs of node ni
nj is the neighbor gateways of node i
$nj (N(Cid))$ are the neighbor Cluster Head IDs of ni's neighbor gateways

In other words, if the set of Cluster Heads that a node can listen to and they are responsible for him is the same set with the set of the Cluster Heads that neighbor gateways communicate with this node, there is no need for this node become a Gateway, because basically covers the same region that the Cluster Heads do. Here we want to select a Gateway which connects two clusters that are not overlapping and they are distributed located and do not connect clusters with the same cluster members.

Intra-cluster Communication

Authors in [32] too, set up a mechanism to conserve energy on cluster members by go on active time interval and sleep time interval. Obviously, when a sensor has to disseminate data is on active period. When a CH wants to forward a query (received from sink) has to wait until sensors are active. The sleep time period must be not too small because radio on and off's of sensor's (cluster member) will waste energy and not too long because the query propagation will be delayed.

Inter-cluster Communication

This type of message exchanging works on backbone created by PC and resulting into a network connected via cluster heads and gateways. Those nodes apparently must be "awake" for data dissemination and query propagation.

Another point to discuss, is that to choose a cluster head in our idea we use an equation(equation 3.4) that considers the cost of the candidate node (number of hops from sink) and the number of neighborhood nodes has. In this way we try to form less clus-

ters, near to sink (to avoid energy consumption of remotely located sensors) and by using a chain-formed, a backbone based structure , data will be propagated to destination. In the backbone we will have cluster heads. No Gateways will be set to simplify and decrease complexity of algorithm. Setting CHs will help us reduce the load traffic presented on data communication.

Moreover, in our implementation we will us only no filter in the Direct Diffusion core, which its role is to set up the nodes do an action, become clusters heads, gateways or ordinary nodes. In other words, set their state. Also, authors in [31] mention that DD has two filters, one Energy filter and one Gradient Filter (Figure 3.1) . In our approach, clustering process will simply change state of each node. There are 4 states, Ordinary, Cluster Member, Cluster Head and Dead node.

3.7. Other Approaches on Energy Efficiency Algorithms for Improving DD and Passive Clustering-Open Issues

This part of this analysis is about solutions and alternative ways scientists propose in order to minimize disadvantages described above.

First of all, in [33] "*Passive Clustering for Efficient Energy Conservation in Wireless Sensor Network* " (PCEEC) authors use the advantages of GRIDS algorithm. This approach tends to set alternate CHs to succeed balanced energy consumption and keep longer the clusters and the structure of network to avoid many frequent changes and many "dead node" announcements [33] .

Although here there is "*a dynamic balancing of the role of clusterhead to the alternate when leaving or failure*", in other words, there is more stable environment for keeping the sections of network and avoid energy exhaustion in specific nodes by alternate their roles. On the other hand, in this research paper authors do not discuss about how we can maximize more efficiently selection of CHs depending on other factors like, energy level remaining, position of node, how far or how close is from other nodes and generally select roles depending on topology structure.

To have a more distributed solution, we must consider those metrics and combine them to achieve better performance.

Another approach about the main disadvantage of DD, which is the flooding in the initial state has been introduced by *Miresmaeil Mirnabibaboli et al.* in [32]*,"Improving the Directed Diffusion in Order to Reduce the Average of Energy Consumption in Wireless Sensor Networks"*. Authors here claim that we can avoid flooding if we classify interest and divide the network geographically. In detail, the network is divided into segments where each segment has its own variable. Every time a source has been found in that segment, the variable of segment is reported to the sink.

This variable has main role about selecting appropriate filter selection after network segmentation. There are two types of Diffusion filter, the first one called Flooding and the latter called Rumor. To decide which filter must be imposed , the sink firstly checks the value of its variables. If one value is bigger than the others, it uses Flooding phase. If the variable has value lowest than the others uses the Rumor Filter.

In a network where we have a large number of queries and very few events, it is more wise to use flood diffusion [32]. On the other hand, if there are few queries there is no need to flood the whole network with information and unnecessary updates of Event-Table. When a node finds a new event, will create an "agent" and add this new event in its local Event-Table. By this way, rumor routing, actually the node that got this new event, produces a packet that "lives" much longer than the others, this is the attribute of agent. Agent now will forward this new event to the further nodes. Concurrently, updates on Event-Table of nodes are made. Finally, the route about this event is indicated by the Event-Table of nodes.

So, when a query is injected in the network, there will be a known path and the query will be transmitted not to all nodes, but only to those that have reference of this query in their Event Table.

Here the authors manage to decrease the large number of messages produced by flooding but, if there is an increasing in the network size and sensors, then the source may not be reached. In addition, if there many areas the delivery rate will be decreased and obviously the network performance will get worse[32].

One of the most important issue in Clustering Algorithms is the selection of Cluster Heads. In our system this mathematical equation has been inspired by the research paper of *Roghaiyeh Gachpaz Hamed* and *Jaber Karimpour,* called *"(EBHCR) Energy Balancing and Hierarchical Clustering BasedRouting algorithm for Wireless Sensor Networks"* [36]. As authors mention, they have invented a dynamic algorithm that selects the best node for cluster head based on three parameters. The energy remaining level, the number of neighbors and the distance between nodes. This leads to optimization in network lifetime by 45 percentage compared with LEACH algorithm.

- The assumptions that authors use is that if a cluster head is closer to the base station, than another cluster head further than base station, then the closest one has more possibilities to become cluster head.
- Another assumption in this approach is that if an ordinary node is closer to the cluster head than another ordinary node further than the same cluster head, then the first ordinary node has more possibilities to become cluster member of this CH.

In [37] *"Energy-Efficient Distributed Unequal Clustering Protocol for Wireless Sensor Networks"*-EEDUC there is an approach considering many variables like intra-clustering communication and processing, inter-cluster traffic and distributed clustering.

In more details, here every node set a waiting time before getting to clusters or become cluster head. This time depends on residual energy and number of neighborhood nodes.

In this approach residual energy can be used to determine a CH and authors emphasize that clusters near base station must have smaller size. This provides smaller energy consumption for communication in the cluster and reserve more energy for communication with all other clusters. Clusters further than base station may have bigger size.

In [38] authors indicate that GRIDS compared to PC has the advantage that select the critical nodes (CH, GW) be the ones that have higher energy level that others. PCEEC uses the same policy with PC for constructing and maintaining the clusters. The main difference is that here we have 6 possible states of nodes. Besides the well-known(initial, dead, ordinary, cluster head, gateway) there is clusterhead-alternate

and clusterhead –ready state. When a node must announce itself as cluster head in the clusterhead-ready mode can become either gateway, either clusterhead-alternate, or Ordinary.

Scientists in this research paper are based on the theory explained in [39] by Heinzelman et al. Heinzelman uses both open spaces (energy dissipation d2) and multi-path (dissipation d4) channels to find the distance between transmitter and receiver. In [36] authors use the following equation to determine which node must announce itself as a CH.

$$\text{Competence\_Value}_{CH} = a\left(\frac{E_r}{E_m}\right) + \beta\left(\frac{N_{elt}}{N_n}\right) + \gamma\left(\frac{D_{ref} - D_{CH\_BS}}{D_{ref}}\right) \qquad (3.2)$$

As α, β and γ are coefficient in front of derivatives, the other measurements are:
E_r=Energy remaining in the node
E_m=initial energy in the node (maximum energy)
N_{elt}=number of neighbors nodes with residual energy less than a certain threshold
N_n=number of all nodes in network
D_{ref}=Longest Distance from Base station (network diameter)
D_{CH_BS}=Distance from corresponding node and Base Station

The following equation(equation 3.3) is about the ordinary nodes to calculate the merit value about the cluster heads that are in their range (so they can hear their announcement). They choose the cluster head that has the greatest value in this.

$$\text{Merit\_Value}_{CH} = \frac{E_n + E_{CH}}{\left(D_{n\_CH} + D_{CH\_BS}\right)^2} \qquad (3.3)$$

Where E_n=remaining energy of node
E_{CH}=remaining energy of cluster head
D_{n_CH}=distance between ordinary node and cluster head
D_{CH_BS}=distance between cluster head and Base Station

These two equations (equation 3.2 and 3.3) lead authors to prove that if we consider not only energy level but the distance between nodes and how many nodes are in the transmission range of a candidate cluster head, we can have a better balanced energy consumption, in other words, avoidance of quick exhaustion of gateways and cluster heads (nodes that tend to accumulate traffic and high delivery rate) and let energy consumption be divided between closer areas, between neighborhoods having almost similar attributes and energy needs.

3.8. Novelty Idea of *" DDRA for WSN"*

In our approach we present a slightly different equation about cluster head election.
In the cluster formation state of network, each node calculate its probability to become CH given by:

$$\text{Probabilty}_{\text{ch}} = \frac{Eres}{Emax} + \frac{Nneigh}{Nnet} + Dref - \frac{Dch\_s - 1}{Dmaxh\_in\_cluster} \qquad (3.4)$$

$Eres$ =energy residual on the node
$Emax$ =initial energy on the node(maximum energy)
$Nneigh$ =number of nodes connected with node in 2-hops at maximum
$Nnet$ =number of total nodes in the network
$Dref$ = The longest distance between sink and node, (in our example, we suppose source is the furthest node)-Graph Diameter
Dch_s =Distance between candidate cluster head and source
$Dmaxh\_in\_cluster$ =maximum number of hops in the cluster (set as default to 2-hops)

The first parameter is about determination of the energy level left on the node. A node with higher energy level will have a greater value in this part.

The second parameter is about the number of nodes that are connected with the corresponding node. It is a measurement for neighborhood in two-hops area. It is more preferable to select a node that can reach and have in transmission range more nodes than select a node that is connected to few neighbors. It is crucial to mention that we consider 2-hops and not one because we would like to give more chances to a node that might be 2-hops far from sink and not be very close to sink (having the problem to give great value to nodes close sink). Furthermore, this approach leads to balance the energy consumption and avoid having many cluster heads near sink . We set cluster heads nodes that are connected to more nodes but not gathered near base, near sink, because that will cause "hot spots" problem.

The third parameter is about the distance between the node and the sink. The biggest distance in the network determine that a node must be "almost" in a appropriate position. This means as close to Sink to deliver with minimum energy packets but as far as it cans to be distributed and away from Sink, to be sparse in the network. For this reason, se set graph Diameter and then we subtract the metric relevant Distance of node. Relevant Distance can be specified by having Distance of node to Sink minus one (this is a constant value) and divided by maximum number of hops we want to set to our network.

The distance D_{CH_S} represents how far the node is from Sink. This distance is related with the energy needed to deliver a packet from Sink to CH. If this distance is long, more energy is needed. If this distance is close, many cluster heads will be created near sink, so this leads to an unbalanced energy consumption solution. We set the Distance from node to sink be subtracted by one, in order to minimize this parameter and let nodes which are close to Sink, get a smallest probability value.

To continue, this Distance is divided by maximum hops we would like to have in a cluster. By setting this measurement to one, makes Distance of node to Sink be larger $(\frac{Dch\_s-1}{Dmaxh\_in\_cluster} = Dch\_s - 1)$ so, a node placed farther from the Sink than a closer one will get lower probability (is substracted by Graph Diameter $-Dref$) than a node closer to sink. This is something we want to avoid. The election of cluster head would be better be distributed, like an intermediate solution in energy consumption and reachability of sink.

Actually, what is more preferable is to have certain number of hops maximum in a cluster, so in the simulation model this has been set to 2-hops.

To conclude, node which is far from sink and setting small cluster (for example 1 hop in a cluster) will get lower probability value. On the contrary, node which is close to sink and setting small cluster (for example 1 hop in a cluster) will have greater probability value.
By setting maximum hops in a cluster to 2, the distance between node and sink, the distance between transmitter and receiver is divided by two, so this measurement tends to choose cluster heads approximately in the middle of the distance between source and sink, intermediate nodes as cluster heads. Apparently max hops in a cluster parameter, depends on network size. Here we consider a small rather network.

Another reason to set maximum hops in a cluster to 2 is because , if we want a cluster with hops up to 5 (a large one), this means that cluster heads between them, will have greater energy requirements to communicate. Cluster Heads which are very sparse and in long distance to each other, will waste a great amount of energy to communicate each other. Indeed, in a wireless system where nodes leave and join very often, this is a major disadvantage.

To summarize, DDRA approach is a combination of two methods. We try to implement the DD method improving with PC approach. Here we set an equation on cluster head election scheme. This equation is inspired by EBHCR protocol [36]. The main goal of our approach is to create clusters in a more energy efficient way according energy resources, node location and network topology. We aim to reduce energy consumption by improving actually PCDD approach by adding not a random selection of Cluster Heads but selection of "advanced" nodes to play the role of Cluster Heads. "Advanced " node can be characterized a node which a) has sufficient energy level to transmit data to cluster members b) can communicate with sufficient number of neighbors so that many nodes can reach in two-hop communication c) have Relevant Distance of Sink , close enough to transmit with minimum energy but distributed and farther from Sink to avoid have many clusters near Sink(this will lead to energy hole if many CHs are near this location).

3.9.   Objectives of *"DDRA for WSN"*
To sum up the following points can describe the open issues in DD and PC Approach:
- Data Redundancy
- "Hot Spots" about Gateway and secondly Cluster Heads
- Large scale of network depending the size
- Creation of duplicated messages, having same information
- Multiple states until path reinforcement
- Increasing energy consuming on initial message exchange and not for data transmission

On the contrary the objectives in DDRA for WSN can be summarized to the following points:
3.1.Reduce Data Redundancy, by decreasing number of messages. Interest messages could be reduced and Gradients would be created with minimized energy cost.

3.2. Avoid "Hot Spots" problem by setting cluster heads on greater energy resources.

3.3. Decrease energy consumption by changing inter-cluster and intra-cluster communication.

3.4. Avoid Additional Headers and Filters on messages, to maintain messages having limited size.

3.5. Allow alternate nodes play the role of cluster heads when CHs are out of energy to sustain communication and connectivity.

3.10. Methodology Choice

Recent researches described above, try to solve problems created by flooding interest in the initial phase of DD and by combining PC introduce a distributed and energy-aware solution. In the DDRA, there is a combination between DD and Formation of Clusters based on energy and location parameters. As DD is a data-centric routing can be combined with data aggregation, which minimize data redundancy and is a method to find optimal paths in a query-based routing. Generally is a routing algorithm of having an optimal path from multiple sources to a single destination. The dissemination leads to creation of gradients and the data matching with interest propagation allows a structure of network where nodes can sense different environmental parameters (for example) and propagate measurements of different events in the same network using routing paths. Events recorded in the cache memory, can be updated, can be deleted and describe the best route data must follow.

3.11. Phases of *"DDRA for WSN"*

In this chapter there is a description and a presentation about the main phases of our approach and the communication between sensors in our network. The routing algorithm has five phases. In each of these phases we have developed different operations on how functions prepare ,control and coordinate modules in the network system and the way they communicate each other and relative to Sink module and Source module.

Phase 1: Transmission of Interests-Initialization and Preparing Network
In this phase, which is the initial phase, four main operations exist.

Interest Propagation: The source starts to flood the network with Interest Message. This messages is transmitted to next neighbors. The nodes will accept the Interest Message and will transmit the same Interest Message to their neighbors. Finally this Interest Message will be transmitted to Source.

Initialize Event tables: Every node has an event table that describes which data can sense. There are 5 different kind of Events, described as constant integer values by 1 to 5. This structure involves the cache memory of nodes.

Initial Energy Level: The nodes are charged initially with 1000 energy units, 20.000 in milli Joule (mJ) or else 20 Joule (J) and they are ready to exchange data.

Network Topology Information: Each node calculates the number of out links and in links it has, actually the number of neighbors in 2-hop neighborhood, the number of nodes exist in 2-hops neighborhood and the number of hops a node is far from Sink.

These are important parameters and calculations to do because in next phase we set the clustering mechanism.

Phase 2: Forming Clusters

In this section we have information about network (location of nodes, connections, graph diameter and energy resources). As mention above by using equation 3.4 each module gets a value for cluster head election. This is the probability cluster head value (Probability $_{CH}$). First we calculate this value for every sensor node and then we calculate the average value of this probabilities. If a node has probability CH value greater than average probability value then it is the first elected as Cluster Head. If a sensor node has a probability CH value lower than average value it gets the Cluster Member role. Cluster Members are connected in 1-hop neighborhood, directly connected with Cluster Heads. In more details there are four states describing each node's status. Ordinary node status, this is default for all sensors in network in initial state, Cluster Head status and Cluster Member status, status given after Forming Clusters state and Dead Battery status, role which is given according energy resources of node, obviously when energy of node is exhausted.

Phase 3: Set up Gradients-Start Communication

In this phase there is one basic function, the setting up of gradients. As referred in [2] by *V. Handziski, A. Kopke, H. Karl, C. Frank, and W. Drytkiewicz* , gradients are important messages that actually inform each node what was the source of Interest Message accepted and having the same Event id in Event table. It is like nodes have a direction on which node Interest came, but having the same Event too. Gradient is a message that is created only if there is a match between Event ID of Interest came and Event ID of node keeps in the cache memory. This ID describes the kind of data a node can sense for example sense the movement of vehicles in region with Cell Id 2. Only nodes that have event id 2 can create gradients for this request. In addition, gradients are kept in cache memory for a certain time period.

Phase 4: Communication Phase-Updating Gradients and Making exploratory data

In this phase each node can exchange the following message types , gradient messages, exploratory data messages and optimal path messages. When a node has no longer gradients in its memory or there is an update in Event Table of node, this node must receive new Interest Messages and create again its new gradients. Concurrently, by the time a node has a gradient (after having a match on kind of data requested - *"event id"*), the sensor node starts to propagate to its neighbors exploratory data. Exploratory Data are messages that follow the path created by gradients(path created by gradients has as starting point Sink and ends up to Source) .Modules start exchange exploratory data messages. Only nodes that are capable to exchange exploratory data can transmit those messages (having a match on event, so pointed by gradients). Exploratory Data are forwarded and follow paths created by clusters. Clusters already have been as described above.

Phase 5: Path Reinforcement-Finding Reverse Path

When Sink accepts an exploratory data, then can start reinforcement path. This is the reverse path ,Sink will start to propagate another type of message . In our model this is indicated by changing Sink icon as Antenna Tower. This new type of message is Optimal Path message. This message shows what is the best path to follow between

Sink and Source, to communicate. Eventually, optimal path messages will have as destination the Source. Those messages record the path they follow (source of message and destination of message). By this way, source will have information for the best optimal path for network communication between sensor nodes.

3.12. Phases of DDRA for WSN
Phase 1: Transmit Interest Message
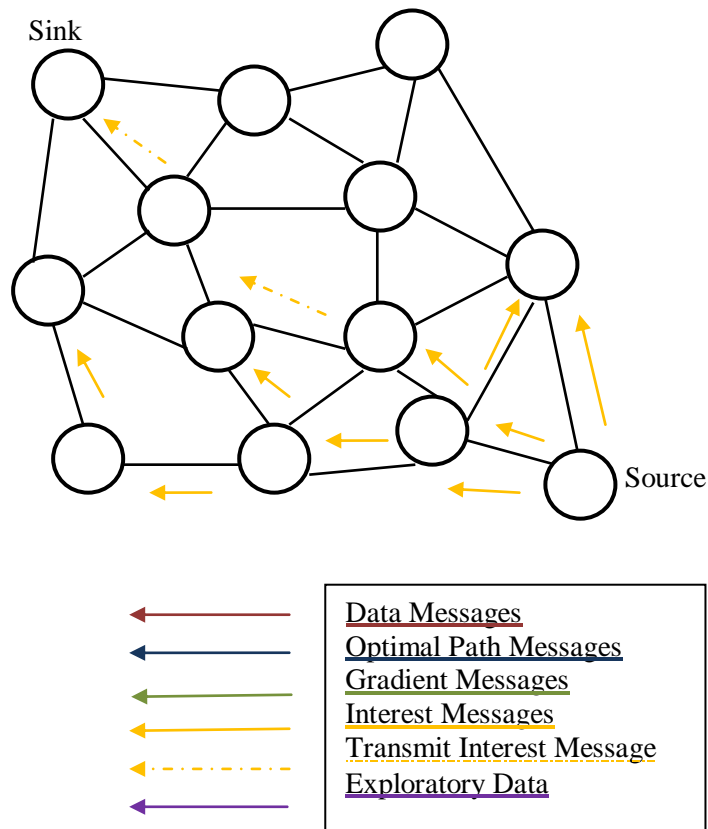
Figure 3.3 Phase 1 of DDRA for WSN

Phase 2: Forming Clusters
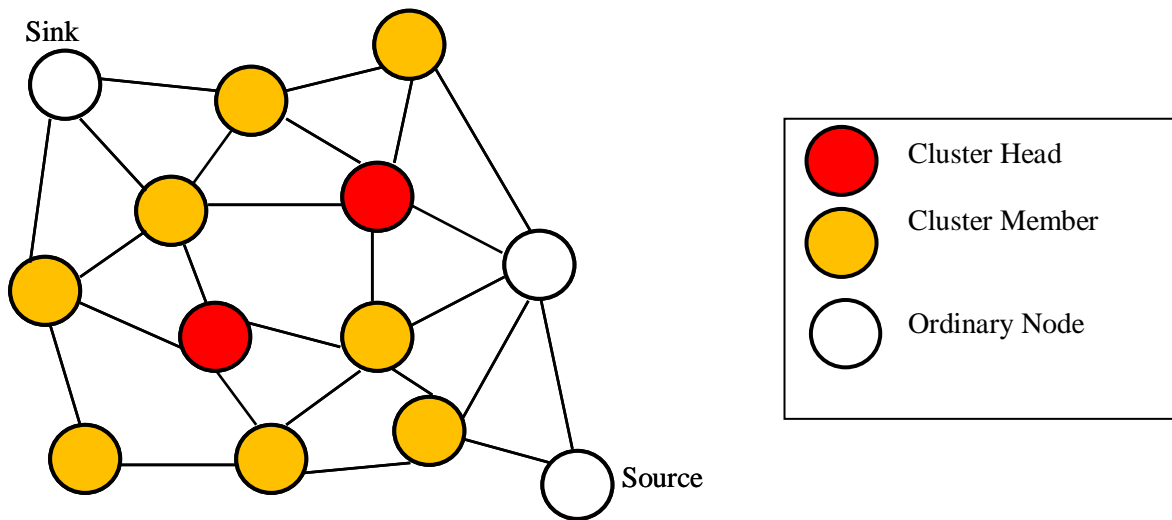
Figure 3.4 Phase 2 of DDRA for WSN

Phase 3: Setting Gradients
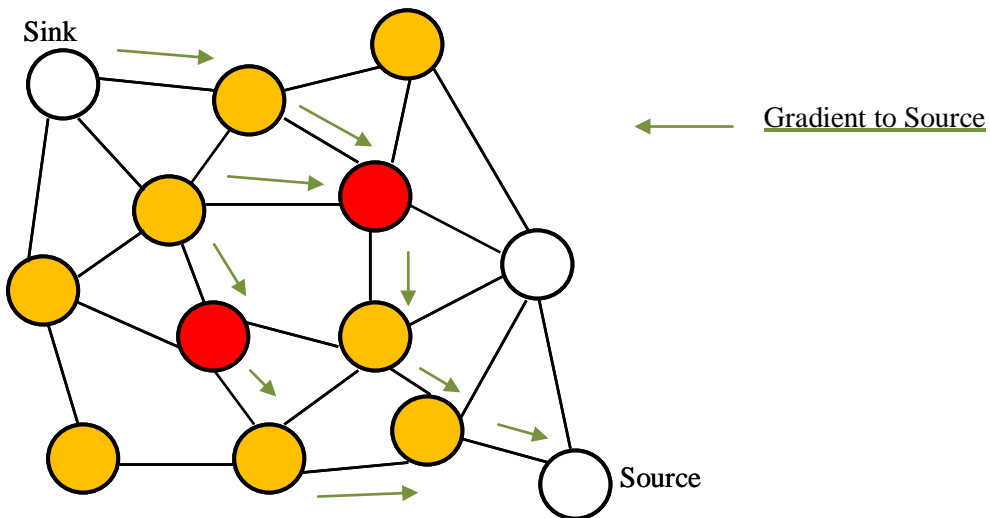
Figure 3.5 Phase 3 of DDRA for WSN

Phase 4: Exploratory Data Propagation
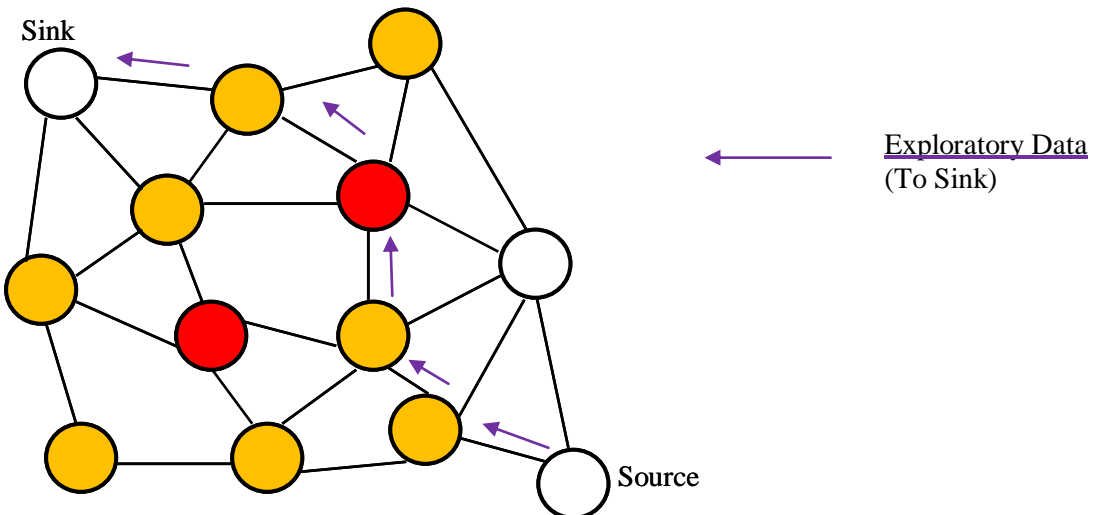
Figure 3.6 Phase 4 of DDRA for WSN
Chapter 3 Description of DDRA for WSN   60

Phase 5: Path Reinforcement – Start reverse optimal path
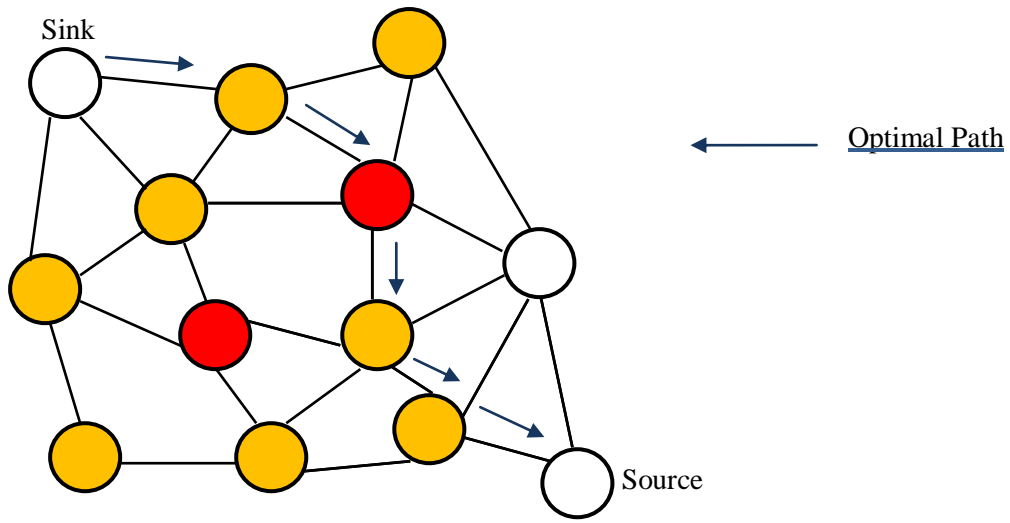
Figure 3.7 Phase 5 of DDRA for WSN

3.12. Event Diagram

Here we present an event diagram on DDRA protocol procedures.

Static Source	Cluster Head Node	Regular Node	Static Sink
1. Source start to flood network with Interest			
	2. Each Node calculate the Probability value to become CH		
	3. Cluster Head Node announce itself as CH and sets Cluster Members		
		4. Every Node creates Gradients Condition (match of Event ID and Event of Interest)	
	5. Propagation of Exploratory Data (through clustering scheme)		
			6. Sink accepts Exploratory Data and start path reinforcement
7. Source accepts information for reverse path. **8. Data packets are forwarded through optimal path and clustering**			

Table 3.3 Event Diagram for interaction between sensors, sink and source during simulation

63

Chapter 4

Description of Proposed System

This chapter refers to channel and energy model attributes. This configuration is about the simulation model of DDRA.

Energy model and Wireless Channel

4.1 About Energy Consumption-Energy Model

Authors in "Sensor Node Lifetime: An Experimental Study" [40] present a study about measuring sensor node lifetime with plenty of commercial batteries. They focus on TelosB platform which is running on TinyOS. Based on this research they inform about power consumption for TelosB radio transmission.

In this project, the nomimal power consumption is 1.8mJ is Active mode and 5.1 μJ in Sleep Mode. In addition, RF transceiver consumes 19.7 mJ when there is a receipt, 17.4 mJ when transmits a message and 0.01 mJ when is in sleep mode.

By considering this information, in our project we set the following power consumption about message propagation:

Type of Event	Energy Consumption	Measurement
Data Message Forwarding	4.50 mJ	In mJ (milli Joule)
Record certain particular message-Forwarding	4.0 mJ	As Energy
Setting clusters- Message Exchanging	4.0 mJ	Consumption
Interest Forwarding	4.0 mJ	
Gradient Generation and Forwarding	2.0 mJ	
Exploratory Data Generation and Forwarding	2.0 mJ	
Optimal Path Generation and Forwarding	2.0 mJ	
Total energy cost	22.5 mJ	

Table 4.1 Energy Consumption of Message Propagation

To calculate the total energy a sensor requires while operates in "awake state" we consider two types of energy, energy required for transmitting messages and energy required for receiving messages.

Cui et al. in [41] show that the energy consumption of a node is separated into two factors:

Power consumption of amplifiers Ppa and power consumption of circuit blocks Pc. So the power consumption can be described as:

$$Pt = Ppa + Pc \tag{4.1}$$

Also in [42] authors indicate relay region. Here if we set sender node S and R the relay node, nodes in relay region can be reached with least energy E. In our approach we set an amount of energy, given by (4.3) to consider energy cost of transmission of a message.

Here we suppose and develop a simple energy model for local interactions. In addition in our protocol there is no mobility or wireless communication.

The authors highlight in [43] below also, that the total energy consumption per bit is given by:

$$Et = \frac{Ppa + Pc}{Rc}$$ (4.2)

where *Rc* is transmission bit rate, *Ppa* is power consumption of amplifiers and Pc is power consumption of circuit blocks.

In our energy model we consider following two equations on energy consumption, that refers the first one to transmit a message (equation 4.3) and the second about receiving a message (equation 4.7) .

The total energy of transmitting a message is:

$$Esnd\_msg = VxNxEbt$$ (4.3)

Where *V* is number of bytes,
N is number of messages
Ebt is the energy consumption per byte of a sensor

In our model we set that most messages will have fixed length, we will calculate each transmission energy (for one message, N=1). In following section, there is a detailed description for energy units of each message.

Energy Consumption Equation=	Sending Message to Random Gate	+	Sending specific message	+	Forming clusters-Local communication	+	Interest Messages Propagation	+	Gradient generation & Forwarding	+	Exploratory Data Forwarding	+	Optimal Path Message Forwarding
milliJoule	4.5 mJ		4.0 mJ		4.0 mJ		4.0 mJ		2.0 mJ		2.0 mJ		2.0 mJ

Figure 4.1 Energy Consumption Equation

According the authors in [40] to calculate the amount of energy that a sensor consumes, two parameters are crucial, the number of bits (N) transmitted in unit of time and the distance between the two corresponding nodes. In our approach messages have different packet size according the type of message and the amount of data they carry. We have set all messages have a header of 10 bytes. Interest and Optimal path message have 50 bytes length. Exploratory data messages and Gradients have 25 bytes length. Data messages are the largest messages exchanging in network and they consist of 100 bytes. The distance d between two nodes is considered as minimal in order to set node i can reach in its transmission range next node j.
The energy consumption formula of transmitting (equation 4.4) and receiving data between two nodes i and j is as above :

$$Etx(i,j) = E_{snd\_msg-rnd} + E_{snd\_msg} + E_{cl} + N \times \left(E_{snd\_Int} + E_{opt\_msg} \right) + \left(E_{exp\_msg} + E_{grad\_msg} \right) \qquad (4.4)$$

Where $E_{snd\_msg-rnd}$ is Energy consumed to send a message to a random gate (next neighbor)
E_{snd_msg} is Energy consumed to send a message and track the path
E_{cl} is the Energy consumed by a node while forming clusters
N is the number of bytes of messages
E_{exp_msg} is Energy consumed to send exploratory data message to next node
E_{grad_msg} is Energy consumed to send gradient message to next node

Let's consider that a simple message for a random next node consumes the same energy with a message a node sends to track path , this energy is the same with the amount required to transmit exploratory data and gradient message to next node.
So (4.4) can be written as:

$$Etx(i,j) = E_{snd\_msg-rnd} + E_{snd\_msg} + E_{cl} + 2 \times \left(E_{snd\_Int} + E_{opt\_msg} \right) + \left(E_{exp\_msg} + E_{grad\_msg} \right) \qquad (4.5)$$

$$\Leftrightarrow$$

$$Etx(i,j) = 4 \times (E_{snd\_msg-tr}) + E_{cl} + \frac{\left(E_{snd\_Int} + E_{opt\_msg} \right)}{2} \qquad (4.6)$$

where $Esnd\_msg\_tr$ is the energy consumed for sending messages scheduled, tracking path, exploratory data and gradient message from i node to next node j,
Ecl is the energy consumed by a node to send data for clustering formation and
$Esnd\_Int$ is energy consumed for sending an Interest message and,
$Eopt\_msg$ is the energy consumed for sending an optimal path message.

In this part we will explain the concept of energy consumption of receiving a message. This can be given by type:

$$E_{RCV} = V x N x Et + Ec, \qquad (4.7)$$

Where V is number of bytes,
N is number of messages,
Et is total amount of energy accept a message and
Ec is a constant variable of a sensor while there is no data collection.

The measurement of energy constant has been provided by authors in [42] below.
In our energy model we set that E_{RCV} of a message is equal to 4.0 mJ

4.2 Wireless Channel Properties
In this part there is a discussion about the channel that is used for connection between nodes. Although the channel has a predefined delay of 100 milliseconds in the initial-

ization phase of algorithm, there is a mechanism about mobility and the power of signal required to permit two nodes to receive and send messages. This part was developed in Castalia Model version 3.0.

Castalia is a simulator for Wireless Sensor Networks and Body Area networks created by

NICTA Australia (Australia's Information Communications Technology (ICT) Research Centre of Excellence). As indicated in omnetpp.org web site [43] below Castalia is a new version that "implements the Baseline MAC for Body Area Networks" and is a simulator for networks consist of low-power equipments.

Path loss generally is the decrease of a signal's power as it is transmitted in free space. In [44] belowthere is a analysis about the wireless channel NICTA group used. In wireless networks, implementing wireless channel is difficult when you must consider mobility of nodes. Scientists here divided the network into cell. They used functions to locate the cell of each node and then they calculate signal strength and the connectivity between nodes. One important issue here is to estimate path loss between mobile nodes. To address this problem, the team group have been based on lognormal shadowing model below[45] below.To calculate the average path loss in dB according distance, the following formula was used:

$$PL(d_0) = PL(d_0) + 10 \times n \times \log\left(\frac{d}{d_0}\right) + X\sigma \tag{4.8}$$

Where PL(d) is path loss at distance d
$PL(d_0)$ is the known path loss, when we have a known distance d_0
n is the path loss exponent and,
Xσ is a Gaussian zero-mean random variable with deviation σ

In wireless channel authors had to calculate path loss at certain time periods. To achieve this, they created a path loss map in first place. Later they set the $PL(d_0)$ parameter as

$$PL(d_0) = (T \times PowerUsed\_dBm - max \times (receiverSensitivity, noisefloor + 5dBm)) - 10 \times pathLossExponent \times log(50) \tag{4.9}$$

supposing that there is a 50 meters unit area we consider.

If the signal's power is above signal delivery threshold (this value shows how many nodes are affected when a node transmits and if a signal is not strong enough, must not be transmitted) , then we add this element into array.

Signal Delivery threshold has the default value of -100 dB, which means that only signals above -100dB must be transmitted.

Generally, in our model a two-dimensions array is used to save the cell id of each node, the first field is about "last observed difference between path loss measured in this simulation time and the average path loss" and the second field is about the simulation time this was measured. A pathloss map has been set considering we have 4 cells with IDs 1,2,3 and 4 respectively. The path loss element is updated with the new path loss measurement. Supposing that in the cell there are nodes that their signal power is strong enough to communicate to each other and the path loss to communi-

cate with further cells is increasing. The increased pathloss value prevents elements in pathloss array communicate between them.

Although this is a crucial subject about WSN, this approach focus on routing layer so no further discussion will be exist on the parameters of a wireless channel. Obviously, here we consider the following:

- Each cell can affect 1 cell in the region.
- The nodes are static.
- All nodes are connected with a channel with 100 ms delay.

Chapter 5

Methodology Choice

This chapter is about methodology choice of our implementation. Also there is a justification for every mechanism was used.

5.1 Introduction

Chapter 5 is dedicated to the methods and the operation of our simulation model. Here there is an analysis about how we developed our system, how this system works and why we use these techniques. This analysis refers to the system consists of 12 nodes, one sink and one source. Similarly, the same methodology is chosen for larger systems.

5.2 Analysis of *"DDRA for WSN"*-Methodology Choice

The main parts of *DDRA for WSN* based on the simulation model has the following parts:

    A. Initialization of the network
    B. Management of Message Communication

5.2.1 Initialization of the network

    A. Initialization of the network

In this part we set initial state of nodes about energy resources, we set initial state of each node as an Ordinary Node, we get information about topology of network, for example Graph Diameter, number of neighbors on each node, number of hops from sink and determine the cell each node belongs. We have divided area into 4 cells equal in size.

Another important issue in this phase is to define parameters that we will examine and compare in the final state of communication. These are number of messages arrived on each node, number of messages transmitted on each node, number of hops (hop count) that a message did until reaches final destination and the simulation time required for transmitting, delivering and receiving particular messages.

In addition, in this state we set nodes begin message delivery by creating some self-messages, at simulation time 0.0, while we boot the process.

Energy resources
    1. Initialize the energy resources of nodes, given a random number between 800 mJ and 1000 mJ.[1]

Network Details
    2. Number of Nodes: Source, Sink and 12 Nodes: total number of nodes 14.

Justification:
We expect that nodes must have slightly different energy amount to calculate approximately at first stage Cluster Heads and Cluster Members. For that reason we have

---

[1] See also Appendix A (Section I Part 1)

some nodes take a lower energy level and some others a higher energy level. During communication energy level will be reduced obviously.

Initial State of Nodes-Clustering Information
3. Initialize state of nodes. Every node has four possible states. Obviously those states change by formation of clusters and these are:

1. CL_HEAD=1,
2. CL_MEMBER=2,
3. ORDINARY=3,
4. DEAD=4

The nodes take the state Ordinary Node at initial phase by having the value of 3 (enum statement).[2]

Topology of Network
4. Printing information of every node about theirs links and their position.
   Here we set several information about:
   - how nodes are located in the network
   - how many neighbors they have
   - how many neighbors are in two hops neighborhood
   - how many hops a node is from longest distance, we consider sink

Justification:

In our simulation model we suppose that nodes are static and there is no mobility.
This measurements are important and required to calculate the chance that every node has to become Cluster Head or Cluster Member.[3]
Also in this phase we set as initial energy level of every node to 1000. Additional information about:
   - Maximum Distance from Source to Sink (Graph Diameter)
   - Number of nodes that each node has in maximum 2-hops
   - Energy remaining in node
   - Maximum hops we set for a cluster (here is maximum 2-hops)
   - Calculate the probability each node has to become Cluster Head by using mathematical type:

$$\text{Probability}_{\text{ch\_election}} = \frac{(Energy\ left)}{(Energy\ maximum)} + \frac{(Neighbors\ in\ \max 2-hops)}{(total\ nodes)} +$$
$$Max\_Distance - \left(\frac{(Distance\ from\ Sink\ in\ Hops - 1)}{maximum\ number\ of\ hops\ we\ want\ for\ clusters}\right) \quad (5.1)$$

Then we calculate the average probability each node has to become cluster head and this value is required in the comparison and exchanging message between nodes to set CH and CM.

Justification:

---

[2] See also Appendix A (Section I Part 2)
[3] See also Appendix A (Section I Part 1)

Chapter 5 Methodology Choice

71

This value gives the opportunity to more powerful and distributed located nodes to have more value of getting the role of CH. CHs must be distributed, consume less energy and not located all near sink or source to avoid "hot-spot" problem.

Observation and Network Statistics
5. Set parameters for watching :
a. number of messages sent
b. number of messages received
c. number of hops messages have
d. end-to end delay in message exchanging
e. energy consumption

Justification:

These parameters are required to prove and observe about energy efficiency of algorithm. Although these parameters have dependency between them. Energy consumption is connected directly with number of messages received and number of messages send. Number of hops, to add, is also connected with energy decrease. While a message is transmitted and forwarded by a large amount of nodes, energy level of each transmitter falls and the energy consumption of overall system decreases too.

Another subject here is that we want to see how much time it takes a message to reach final destination in two cases: Firstly, when there are no clusters and gradients set up in network and secondly, when optimal path reinforcement starts and messages are delivered based on clustering and path establishment. This fact will help us determine if this is an energy efficient approach and can give us better results and optimize network communication.[4]

Initial Energy
6. Set the energy that every node has spent in simulation time 0, to 0. This will change and certain constant values (representing energy cost) will be added in each generation of message, message forwarding and other procedures (e.g. clustering, local interactions).[5]

Justification:

This will determine energy level of each node. This process will help us calculate the energy spent on each node after transmitting a message. The energy cost is required to update energy level on each node and is the most important parameter in simulation for result discussion as indicates the energy-efficiency design of our model.

Types of Messages
7. Start sending the first messages to start communication, having initial values.
The node indexed 0, which is the Source node which starts exchanging messages of three types:

NodeMsg –Message Having just data about source and destination

InterestMessage-Message having an event id, this will later update Event Table of nodes and determines which nodes can sense data and send them to Source.

Gradient Message-Message on every node having the direction of the previous node that Interest Message came from. These messages are created by nodes only if there is a match on the event id of the Event Table of node holds in cache memory and the

---

[4] See also Appendix A (Section I Part 5)
[5] See also Appendix A (Section I Part 1)

event id the Interest Message describes. Here we set Event table for each node and we suppose we want to search for event id that equals 2. Event Ids are from integer value 1 and continue until 4.

MessageEnergy-Control Message that set all the nodes to Initial Energy. There are four energy states that will be updated depending on energy left on each node:

a. FULLY_CHARGED=1,
b. STRONG_BATTERY=2,
c. WEAK_BATTERY=3,
d. DEAD_BATTERY=4

Exploratory Data-Message that is exchanged after gradients set up phase. Exploratory Data message is initiated by Source and as final Destination has Sink. This type of message helps to discover optimal path, and chooses more beneficial paths in terms of energy consumption.

Optimal Path-this type of message is the one that shows which is the optimal path created in the system.

Justification:

All these messages expect the control message for Energy (Message Energy) have initial values, set to 0, and are used just to make communication available between nodes. These messages are important for setting and configure the network. Each one of them plays certain role about structure of network and data dissemination. Apparently, they describe how nodes communicate and how they can exchange information for all procedures created and occur on the network.

5.2.2. Management of Message Communication
    A.   Management of Message Communication

In this part there is the network communication and there is an analysis about:
- How nodes communicate
- How messages are updated
- How clusters are formed
- How energy level changes
- How roles are set after clustering
- How nodes maintain tables in cache memory
- How gradients are created and updated
- How optimal path is constructed at the final state

The following chapter is about the technique we used to implement DDRA approach for WSN . Every step of approach is described in ascending order, starting from initial phase after node's initialization and complete in the end of communication where energy resources are eliminated and nodes can not continue transmitting or receive.

Part 1: Setting Events on each Node (Saved in cache memory)
    1.   Set the initial Event table of each node. Every node has an event id as random number between value 1 and 4.

Justification:

This corresponds to data that each node can sense and update gradients have to forwarded the data that must reach node to sink. In our approach we set random values about events in the begging. The optimal path created in the final states is connected with event table of each node. Obviously, we can not have an optimal path where the node has not maintain a gradient for its neighbor. Events describe in other words, what data can sense, an attribute about application layer in communication.

Part 2: Creation and Delivery of Messages

2. The messages are categorized into 4 types. According the message type there are different functions that nodes must follow:

NodeMsg Type:

I. Get the state of node. If the state is Cluster Head or Ordinary, then if the message has arrived to source that first generated it (generator node), forward it to network, inform with a Display String and count the following:

- Total messages of this type received by specific node
- The hops that this message did
- The simulation time this message needed to come from sender to receiver (in this case we suppose that must be the same module).

Then delete this message and do not forward this to another module. NodeMsg messages have red color.

I. If the node has state of Cluster Member or Dead will delete this message after measure the three parameters described above.

II. Update the energy level in the node and set the energy lost by previous actions.

III. If the message has not arrived to the final destination (the generator), then send this message to random neighbor and set the energy lost. Energy cost is calculated here as 4.50 energy units.

Justification:

These messages are data messages and will help measure the network performance. In addition, having enough number of messages will show how network performs and how the other type of messages are delayed or not. The "NodeMsg" messages are the initial messages exchanged. They start from any random node and are propagated via forwarding to any random neighbor. They start as self-messages on initialization. Self-message in OmNET++ simulator is a message that is used by simulation kernel, it was scheduled (scheduleAt()) or sent (send()) by a module in network. Usually self-messages represent event intervals. In our simulation model, NodeMsg messages represent a) the amount of data arriving in a node, b) the energy cost we get by transmitting such messages and c) the delay we get having messages transmitting while other types of messages are exchanged. To be more specific, messages that represent interest, gradient and Optimal path information have to travel and set concurrent data messages exchanging in model. This will show how these transmissions affect delivery rate, delay and energy consumption. Each data message will cause energy reduction.

Moreover, in this section there is a particular action when a node is Ordinary or Dead. Dead is the node having energy equals zero, so node can not either accept or forward any type of message. The message will be discarded.

In addition, there is a value about transmitting this data message. Energy cost in each data transmission of this type message is 4.50 energy units. This value has been set by practical way. The reduction of energy resources on each node has been set in a manner to avoid very quick exhaustion of device (for example by setting as energy cost 30.0 energy units, a node forwarding 20 messages in 200 milliseconds would be dead in very short time). On the other hand, this energy cost mustn't have a tiny value as 0.45 mA for example. By this method, a large amount of nodes would have the possibility to become Cluster Heads. Having a large amount of residual energy after many transmissions, is not realistic and adds disadvantages like network load from many messages exchanged of becoming a cluster head. The fact of having plenty of candidates is translated as many competitions about leadership of clustering and consequently many redundant message exchange.

Part 3: Cluster Formation

3.   Set the cluster members and the cluster heads

Here we use the average value of all nodes on probabilities to become cluster heads. The probability to become CH is given by mathematical equation 5.1 and depends on remaining energy, distance and size of cluster and Graph Diameter of network. This value is updated and in combination will residual energy a node will be cluster head, cluster member or dead.[6]

CH color=Red

If a node has *probabilityCH* greater than *averageCHprobability* and energy level is above 800.0 energy units then is Cluster Head, change its state to 1 and forward the message to a random neighbor. Battery State=Fully_Charged. (indicated with green icon)

If a node has *probabilityCH* greater than *averageCHprobability* and energy level is between 500.0 energy units and 800.0 energy units is still a Cluster Head, change its state to 1 and forward the message to a random neighbor but now it has Battery State=Strong_Battery.(indicated with yellow icon)

If a node has residual energy below 2.0 energy units then it becomes Dead, it can not forward nothing and changes its state to 4. Battery State=Dead_Battery. (indicated with red-stop icon)

CM color=yellow

If a node has *probabilityCH* lower than *averageCHprobability* and energy level is between 499.0 energy units and 100.0 energy units it becomes a Cluster Member, change its state to 2 and will not forward the message, just accept it. Now it has Battery State=Weak_Battery.(indicated with red icon)

EXHAUSTED color=white

If a node is a Cluster Member and the energy level remaining is below 2.0 energy units then it is a DEAD node, its state changes to DEAD and Battery state is DEAD.(indicated with red-stop icon). It will not accept no further messages and will not respond. This node is no longer capable to participate in communication. Must be recharged or replaced.

---

[6] See also Appendix (Section I Part 2)

Chapter 5 Methodology Choice   75

Justification:

This is a very important function to set the roles of nodes in network communication. The idea of using this approach was introduced by authors in [2]. As mentioned before, it is a simple technique about clustering and cluster head announcements. We use the average cluster head value to decrease number of nodes having similar possibility to be come CHs. This is done to avoid many competitions between many nodes and re-clustering.

The general idea of electing CHs is like assign to each node a "weight". This "weight" is the probability value of CH and the three network parameters are : 1) <u>energy remaining level</u>, to keep "strong" nodes deliver the future messages and handle future events. 2) <u>position of node</u>, to have not many CHs near sink and in contrast not many CHs remote and far away from sink and source. 3) <u>number of neighbors</u>, to have nodes that can "serve" and communicate with sufficient number of cluster members and to avoid have cluster heads that are connected to small number of neighbors. In this way, the factor of in and out links, gives the opportunity to more distributed and centralized nodes to be elected. This is a semantic approach and does not count only energy level and network lifetime by battery resources.

The categorization into four states of battery node is important because the major issue in a routing approach based on energy consumption is the energy cost of data transmissions and definitely those states are necessary to show and define how many nodes can compete in a cluster to get Cluster Head role. To continue, battery state will determine when a node is adequate to become cluster member and when is dead. Through this method, the evaluation of energy cost metric will show how long a node can be a Cluster Head and generally, how long the network communication can be characterized as energy efficient. Apparently, after a long period, when crucial nodes will be exhausted, the network performance will get worse and communication without updates and replacement of devices will not be able to exist.

Part 4: Data Transmission before Path Reinforcement and Clustering

    4. Get information about a certain message. This display the module that a certain message received. Helps to understand the route the message follow, the modules that received this certain message and how the duplicated messages change their ids. The tree id of a message although is unique and help us understand for every message which module created it and which forwarded it. A message can not be received by many nodes concurrently so it was necessary to create a large number of duplicated messages to keep communication in our model.

This function costs to each node 4.0 energy units as a energy level value.

Then Update the energy of nodes. This is done frequently in the network operation, because we have to get the most recent values about energy remaining on each node.[7]

Justification:

This operation has been set to show the path that messages follow. In addition, this contributes to minimization of energy level at nodes as more messages are created and forwarded. In this function energy cost has been set to 4.0 energy units. This function helps us observe the hops that an data message does during network communication. The results are that many data messages are frequently exchanged between same sender and receiver, (many back and forth propagation between two nodes). This is maybe caused by many random forwarding of messages. This has a very significant

---

[7] See also Appendix A (Section I Part 4)

Chapter 5 Methodology Choice 76

role in energy consumption of a node and represents a great amount of energy lost in overall network system.

Although, this is a costly in energy process, recording message paths is required to see how messages are forwarded, what is the total energy cost of this propagation and how we could optimize and update drawbacks of this data transmission.

Part 5: Propagation of Interest Message

5. Interest Message. Interest messages has 4 fields:[8]

6. Source generated Interest Message. Firstly Source generates Interest and tries to forward this towards Sink

- Destination: this field changes and is updated by every node that accept this message

- Hops Counts: To measure how many hops did the message to get from source to sink

- Sink Id: This is set to 15, to compare the Sink id and the id of each module that received message. If those values are equal, message has arrived to Sink. Sink turns to green color.

- Event Id: This is set to 2. This indicates that Interest message concerns nodes that have event id 2 in their event table.

7. After Initialization of Interest this Interest is forwarded to all neighbors of Source. This is also a attribute that is discussed in [2] about data dissemination. If next node is Cluster Head or Ordinary node this message is forwarded. On the contrary, if next node is dead or cluster member then this message is deleted. At this message we keep tracking of message to count the hops and record the path this message follows. Interest messages have yellow color.The transmission of this message consumes 3.0 energy units.

8. If Interest reaches Source then will be deleted and hops will be displayed, if Interest reaches to Sink, there is a display of Information about Interest and then again the message will be deleted, also.

Justification:

Interest message is a message that starts updating event tables of nodes and basically represent a request on the kind of data we want to sense. Actually the result of algorithm is to find optimal paths but in reverse order the Interest travels, for example Interest travels from Source to Sink. On the contrary, the optimal path we have to find at the final state of approach refers on how messages must be forwarded from Sink to Source. In the simulation model, source is in the lower position and sink is located in the upper position. We must mention that the distance between source and sink considers the graph diameter of network, the maximum hops we have between remote nodes.

The main difference between the approach presented in [2] below, is that although in the initial state, sink starts propagation of Interest too, the sink is located in the lower position. In our implementation the sink is on the opposite location.

---

[8] See also Appendix A (Section II and Section III)

The choice of changing this attribute of network topology, is not significant, it is done only to be more easy to understand how data are transmitted in initial state. The way and the technique communication is conducted, is based in the same principles.

Part 6: Creation and Propagation of Gradients[9]

9. Gradient Message: This message is created by each node and actually shows where a node got an Interest that match with its Event table values. In other words, nodes create gradients to each node that Interest came (previous node) , having a match. At initial state match id is set to 0. If node again is capable (not Dead or not Cluster Member state) will update the gradients and forward message. Gradient messages have green color.

10. Gradient has 2 fields:

- Node id: This is the node id that has the gradient

- Source Int: This is the node id that Interest came from. Actually, this is done by record the previous module, and record the gate id that Interest message passed through.

11. When a node receives a gradient, it updates the values and will send this to all its neighbors. Then there is a significant process based on Event id that a node has in the Event table and the sender module of Interest message. There is a comparison between a) the gradient values a node holds and b) the event id we search in our model. If the node has gradient for a neighbor with Event id equals 2, then this node generates another type of message Exploratory data. Exploratory data are forwarded to the final destination, the Sink. When Sink accepts Exploratory Data, its icon turns into antenna tower icon.

The energy consumption of both gradient and exploratory data can be measured to 4.0 energy units (2.0 energy units for gradient setting, 2.0 energy units for exploratory data).

12. Energy Model

In our simulation model we have set nodes to have as initial energy 1000 energy units. We consider that each energy unit corresponds to 20 mJ. Similarly, 4.0 energy units means actually 4.0 x 20 mJ=80 mJ energy consumption and 2.0 energy units is 40 mJ energy consumption.

This consideration is based on [48] research paper about EEQR protocol for clustered WSN. In this research project, authors measured that a system with a static sink has as average energy consumption 0,0861330 J per packet. Although this is energy consumption of proposed EEQR routing protocol ("*Energy efficient and QoS aware routing protocol for Clustered Wireless Sensor Network*") of this research project, we will use this energy model as basic in our model which has similar attributes as clustering architecture and energy efficient attribute.[10]

---

[9] See also Appendix A (Section I Part 3)
[10] See also Appendix A (Section I Part 1)

13. Here we set Event id equals 2. If the node has a gradient for a neighbor having Event id 2 then Exploratory data can be created by node and forwarded to destination. Destination of Exploratory Data is Sink.

Justification:

As in "Improving the Energy Efficiency of Direct Diffusion Using Passive Clustering", authors present the significance of gradients. A gradient is a type of message that shows the sender of Interest. Here there is a comparison between Event ids that a node holds in cache memory, and the event id Interest has. If there is a match on this field, then node can create a gradient to sender. It shows from which node accepted Interest, an Interest that refers to a specific event. So, finally, all nodes that have same Event id and accepted same Interest with this Event id, will create a gradient to this "previous" sender. The DD is a data-centric routing on application layer. By this method, if we examine all gradients that every node has for a specific event id, we will result in best path from source to sink (reverse path of Interest propagation from sink to source) that refers to this event id.

The PC is used to prevent long paths in the network and construct paths including reduced number of nodes and reduced number of hops to reach final destination. Here the gradient to a node that a module holds is not just any of the 14 modules in the network. The gradients refer to Cluster Heads, devices to propagate data in a energy-efficient way.Gradients are important to determine final path reinforcement.

Part 7: Creation and Propagation of Exploratory Data

14. Exploratory data are simple data messages showing what kind of data a node can sense and send to Source. [11]

Justification:

The Exploratory data are data messages and finally must follow path indicated by gradients. Although Optimal path is not already constructed, Exploratory data must be created by nodes that have a match on Event Id of gradient they have and the Event id is searched. Where a node accepts a Interest and creates a gradient, to the "previous node", if there is a match on Event Id's, then only this node can create the Exploratory Data, representing this event, this type of sensing data.

Exploratory Data have not particular parameters and fields that have to be updated, although they play an important role in the overall procedure. They are data messages but they are created only by nodes having gradients and searching the same Event id.

Exploratory data can represent an environment parameter for example the temperature or the humidity on a certain criteria described by an integer value.

To sum up, gradients are created by propagation of Interest from Source to Sink. Gradients show the path from Sink to Source and Exploratory Data follow reverse path which is from Source to Sink.

Part 8: Creation and Propagation of Optimal Path Messages

15. Optimal Path Messages: As Exploratory Data reaches Sink, then by matching again event ids of getting value of gradient in each node and event id we look

---

[11] See also Appendix 1 (Section I Part 4)

for (here is 2), then Optimal Path messages are created. Optimal Path messages have 6 fields:

- Src ID: The id of the node that has this kind of message (index of module)
- Event ID: The event ID that optimal path message will describe. This is set by call of reference.
- Source ID: The id of Source (set to 15)
- Sink ID: The id of Sink (set to 14)
- Target ID: The id of the node that must be the next, the one to forward Exploratory data, the module that has match on event id and this node has a gradient.
- Target index: The index of target, of next module. This value helps to control values and record optimal path. Communication of nodes is more easy based on indexes of nodes than ids, so this field is set for easy functionality.
- The transmission power consumption concerns this message is set to 2.0 mA.

16. Optimal path message holds the information for the target the messages must follow. After generation of Optimal path message, which have blue color, every node displays which next module is the best, the optimal and forward the message to it. Optimal path follow the path from Sink to Source.

Justification:

Optimal path messages represent the path reinforcement. Actually, after generation of Exploratory data and propagation from Source to Sink, the data follow the path including Cluster Heads (using integer event id value).Optimal Path message shows the best path. The field about target is about the next "optimal" neighbor to forward the data messages. The target module, as it is described, must be a Cluster Head, must have a gradient for next target, must have generated Exploratory data and eventually, will forward "optimal path" message to Source.

This type of message does not exist in other approaches but here it is created for two main reasons:

- To show how the optimal path in network
- To gather information after events and data transmission
- To record optimal path and keep nodes that must be used for transmission

By forwarding to next target, on display screen we can record path and show the best optimal path according cluster role state and matching criteria.

17. To keep Optimal Path clean from complicated methods there is a dedicated gate on each node for receiving optimal path messages, which are DirectIn and DirectOut gates for input messages and output messages respectively.
18. Optimal Path messages are inserted into a queue. This queue contains the next target, actually the next node that algorithm must choose. After printing this information, we have the best routing path.

Justification:

Here this is a presentation of final state of network communication. The propagation of Exploratory data is not based on random gates and random neighbors. Here the knowledge of best next node, best target module is declared. To force nodes, (cluster heads), forward messages to specific nodes, there is a requirement to maintain dedicated gates for this data exchanging.

Although, all other messages use the usual input and output gates, here "optimal path" messages use only dedicated gates. Sending a message through these gates helps, at first, keep messages arrived on other gates do not be delayed and not be confused or

having collisions with other messages and secondly, make communication and propagation of these messages faster and in a more simple way.

Part 9: Packet Loss Ratio

19. In our model we have set Data packets have 100 bytes length. The Packet Loss can be defined as the failure of one or more messages arrive to final destination. In [46] authors claim that the usage of balanced energy among nodes can improve QoS parameters like delay and packet loss. In this simulation model, we consider that packets are created after optimal path establishment. Each node can send a data packet to next module, but not any module just connected to source. The node sends data packets to next best target for routing. A simple outgoing gate sends direct this packet. There are packets that are transmitted successfully and packets that can not be transmitted. Nodes that are dead or they do not belong in the optimal path, can not transmit those data packets. Some nodes have a percentage of 100% packet loss ratio and some nodes (best for routing) have zero packet loss.

20. Finally, we record the number of successfully transmissions on each node and the average number of packet loss ratio characterize our network performance.

Justification:

To describe and evaluate results of this approach we must consider performance metrics like delay, throughput and packet loss. In our implementation, we consider data packet exchanging after best path establishment. The disadvantage of this consideration is that data messages that nodes exchange during communication are not added in network load, until final phase of algorithm. On the other hand, this approach allows us to measure the data bytes exchanged in final phase and to record the average packet loss ratio.

Part 10: End of Simulation
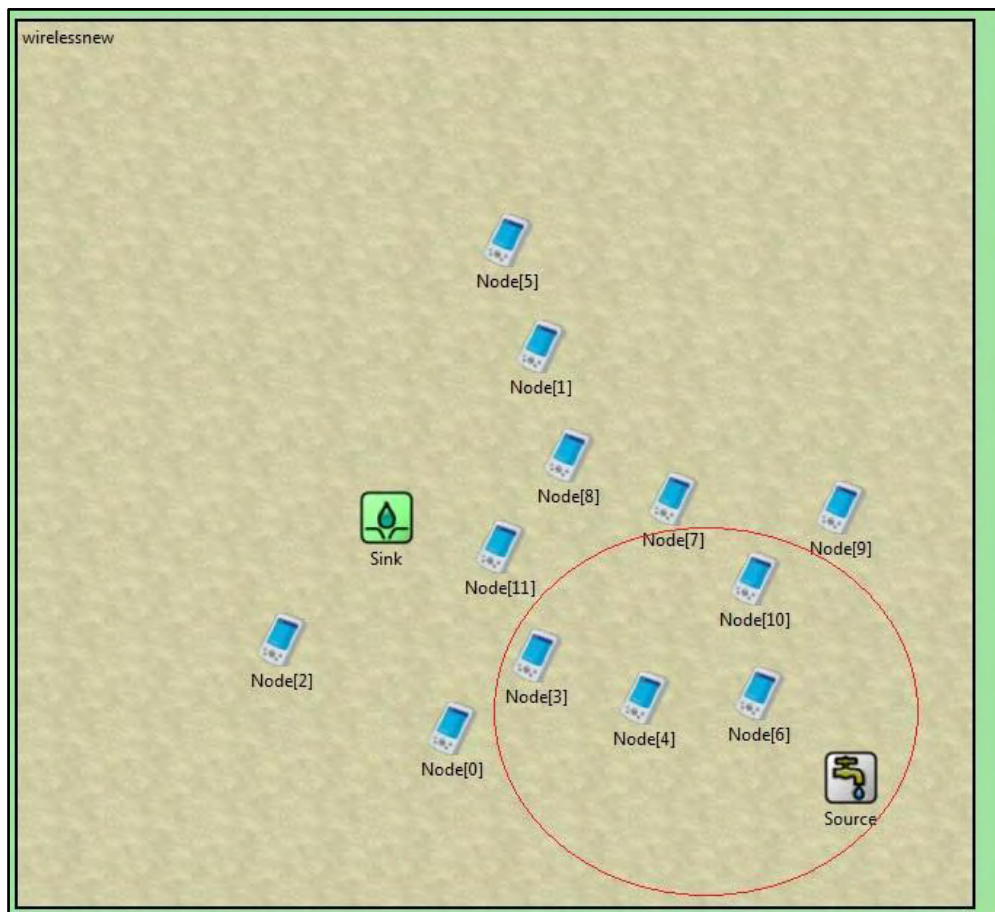
Figure 5.1 Simulation Model-Nodes near Source module

21. The simulation model stops after creating the optimal path, obviously. As the nodes keep exchanging many types of messages and after many recalculations of clustering formation the total energy consumption increases. Nodes keep changing status from cluster heads to cluster members as energy resources decrease, node status becomes dead when battery level equals to zero energy units. To stop simulation model, we have to consider that communication between sink and source must be infeasible. When the nodes create their gradients, the end condition affects nodes around source. In detail, in our model nodes with index 4, 6, 3 and 10 is close to Source. Node 6 is directly connected with Source. When those nodes are dead, there is no capability to create gradient for source. In this phase, communication stops. By calling finish() function all measurements are recorded and histograms for analysis of results created by OmNET++.12

22. Until this condition is true, the gradients have already been set are, for example from Node 4 to Node 6 and from Node 3 to Node 4. When end condition is true, no more data messages are capable to reach final destination, which is Source. (Figure 6.11)

---

12 See also Appendix A (Section I Part 5)

Chapter 5 Methodology Choice   82

23. There is no standard time that this model will stop. Changing parameters like energy consumption, node location and connectivity of nodes the simulation model will stop at different time (in milliseconds).

Justification:

This routing algorithm is about discovery of best optimal path between source and sink. When nodes are either directly connected with one of these entities or neighbors of them are exhausted and they can not relay any more messages, path does no more exist. Based on previous methodology description, we have recorded until this phase best path and main goal of this research seems to be completed.

5.3. Pseudo-Code of DDRA-Communication Phase

Pseudo-Code of DDRA for WSN

Input: Number of neighbors, Number of Nodes, Energy resources, Max hops in clusters, Distance (Graph Diameter), Distance Node from sink, Event ID table in cache memory

Output: Clusters, Optimal Path reinforcement

Begin
1. Transmit to all nodes Interest Message (Having an Event ID)
2. Nodes start making clusters
3. **If** (Node!=Dead**) then**
4.   Find probability to become Cluster Head
5.   **Energy resources + Number of Neighbors +Graph Diameter –Hops from Destination**
6.   Find *average $Prop_{CH}$* of all nodes
7. **endif**
8. **If** (*$Prop_{CH}$>Average $Prop_{CH}$*) && (*Energy resources>minimum Energy Threshold*) **then**
9.   Candidate Node=Cluster Head
10. **else If** (*$Prop_{CH}$<Average $Prop_{CH}$*) **then**
11.   Candidate Node= Cluster Member
12. **else if** (*Energy Resources==0.0*)
13.   Node=Dead
14. **endif**
15. Create **Gradients** pointing to previous nodes (sources of Interest)
16. **If** (*EventID==Interest's Event ID*) **then**
17.   Make *Exploratory Data*
18.   Update *Gradients* to match *Event ID*
19.   Update *Energy Resources*
20. **If** (*Exploratory Data* reaches Destination) **then**
21.   Follow updated *Gradients*
22.   Start *reverse path reinforcement*
23.   Record *Optimal Path*
24.   Transmit Data Packets through *Optimal Path*
25. **endif** *//Exploratory Data*
26. **endif** *//match on Event ID*
End

5.4. Conclusion

The methodology of this routing approach can be described as operations between modules (local interactions mostly) and can be divide into seven main steps. These steps indicate the way this network operates, the phases before discovery of best optimal path and the condition to stop and gather results on network performance. The seven steps are:

1. Preparation of network and gathering of network topology
2. Clustering according energy level of module, location and node density
3. Propagation of Interest and Data Messages
4. Creation of Gradients showing what groups of nodes can communicate having a match on Events that keep in cache memory
5. Propagation of Exploratory Data based on gradients set-up
6. When Sink accepts Exploratory Data, can start path reinforcement, a reverse path showing best route from Sink to Source
7. End of Simulation when communication of Source and Destination is no more feasible. Link-failures in path between Source and Sink.

Chapter 6

Analysis of Simulation Model

This chapter describes the simulation model of DDRA, clustering and generally network performance.

6.1 Introduction

In this section, there is a discussion on simulation model. In the first part, there is a presentation of the simulation parameters and the attributes of the WSN. In the second part, there is a description about clustering. Several figures show how this routing protocol behaves and how clusters are set up in network. In the third part, there is a discussion and analysis about the phases of simulation model, phases of DDRA that described in Chapter 3.

6.2 Simulation scenario

The first simulation model is a network consists of 14 nodes. There are Nodes identified by name Node[0], Node[1],….Node[11]. There are also Sink module and Source module. We have developed two other simulation models, consist of 32 and 64 nodes respectively.

Communication channel

There are connections between those 14 nodes described in the Network Description (.ned) file of system. Although this algorithm is about wireless communication, the implementation of this scenario is based on a wired channel having a 100 milliseconds delay. This is because the focus of this dissertation is about Network Layer and does include further information about communication with MAC layer and Transport layer. Moreover, about channel and connection between nodes there are gates about messages exchange and gates dedicated only to exchange data about optimal path discovery and information.

Although we set a wired channel, as referred in previous chapter, there is a path loss map, according Castalia model that shows how strong the signal is of each node and what is the transmission range of them. Nodes are distinguished too, in different cells. Although our design suppose that network is divided into four cells. In each cell, nodes directly connected with a centralized node (one CH) belong in same cell.

Network Topology

In figure 6.1 we can see how the nodes are located in the environment. Nodes are placed randomly and this localization appear similar to a tree-like structure. The area of network has 700 meter of width and 650 meter of height. (700 x 650) m^2 . The region that nodes are deployed in much smaller in the middle of basic entities, Source and Sink.Each cell of network considers as an area of 5 x 5 m^2.

Energy Resources-Battery Level

We set each module have at maximum 1000 energy units, which means 20.000 in milli Joule. (every energy unit equals 20 mJ).This is the initial energy each module has.

Energy Consumption

The energy consumption is calculated by number of messages exchanged, functions like clustering formation ,transmission of data messages , creation of gradients and

generally the communication between entities of system. Generally, nodes that forward greater number of messages and tend to be elected as Cluster Heads, apparently consume greater amount of energy resources. The condition about energy that exists here is that 22.5 milli Joule are spent in total of each procedure involved (Table 4.1). Obviously, these phases (functions of communications and calculation of parameters) occur many times until optimal path discovery and multiple times in many nodes. This leads to a significant energy loss as entities in network communicate.

Packet Size

In our model we suppose that we have five types of messages. Interest and Optimal path message have size of 50 bytes, because they carry more information. Data messages, has been set to have 100 bytes. Data messages are double the size of Interest message. Exploratory Data messages and Gradients have 25 bytes length. We consider that each packet has a 10 bytes header. Exploratory Data and Gradients are half the size of Interest message and of Optimal path message due to their information. They are control packets and Interest and Optimal path messages have greater priority as type of packets in overall transmission.

Packet Size of Messages

Type of Message	Length in Bytes	Packet Size in Bits
Data Message	100 bytes	800 bits
Interest Message	50 bytes	400 bits
Exploratory Data Message	25 bytes	200 bits
Gradient Message	25 bytes	200 bits
Optimal Path Message	50 bytes	400 bits

Table 6.1 Size of messages (in bits/bytes)

Figure 6.1 Network Topology

Chapter 6 Analysis of Simulation Model 88

In Table 6.2 we can see the parameters of network.

Parameter	Value	Unit	Information
Area	600 x 750	m^2	(600 x 750)
Number of Nodes	14		
Initial Energy	1000	Energy Units	20.000 mJ (or 20 J)
Broadcast Message	50	bytes	10 bytes Header
Packet Size	100	bytes	8000 bits
Control Message	25	bytes	200 bits
E$_{msg}$	4.5	Energy Units	90 mJ/p (mJ/packet)
E$_{elec}$	4.0	Energy Units	80 mJ/p (mJ/packet)
E$_{rec\_path}$	4.0	Energy Units	80 mJ/p (mJ/packet)
E$_{grad}$	2.0	Energy Units	40 mJ/p (mJ/packet)
E$_{msg}$	4.0	Energy Units	80 mJ/p (mJ/packet)

Table 6.2 Simulation Parameters

6.3 Clustering election scheme overview

In this part we will demonstrate the setting of clusters in the simulation model, distinguished by phases of DDRA and simulation time.

6.4 Analysis of Clustering Process

6.4.1 Clustering Process

Methodology: Hybrid method. First every sensor has a probability value to become CH. After some rounds completed, CH probability is recalculated based on energy resources.

Objective of node grouping: Groups help avoid energy holes near sink and source and prevent additional network overload on CHs in a centralized backbone of data traffic.

CH selection: Based on probability by mathematical type. (equation 3.4)

Algorithmic complexity: Depend on network size. Complexity is given by O(n), where n is the number of nodes (variable convergence time algorithm).

6.4.2 Clustering in simulation model

In this chapter there is a description on cluster formation. DDRA sets up clusters and communication is forwarded through advanced nodes-Cluster Heads.

In Figure 6.2 we can see clusters at 300 millisecond. In this phase optimal path messages (indicated with blue color) travel on network and follow next best node. The best node depends on role in cluster (CH) and availability to sense same data for setting a route about sensing same variable (for example in same geographical area). In Figure 6.2 we can see that we have three clusters created after 300 milliseconds. Ac-

tually, the third cluster will be formed after 500 milliseconds (Figure 6.3). Nodes keep change their status and energy. Role of Cluster Head apparently is not permanent. So we have a "dynamic" CH role, which changes to "weak" or "dead" status node during simulation. Besides, as energy decreases nodes could be "Dead".
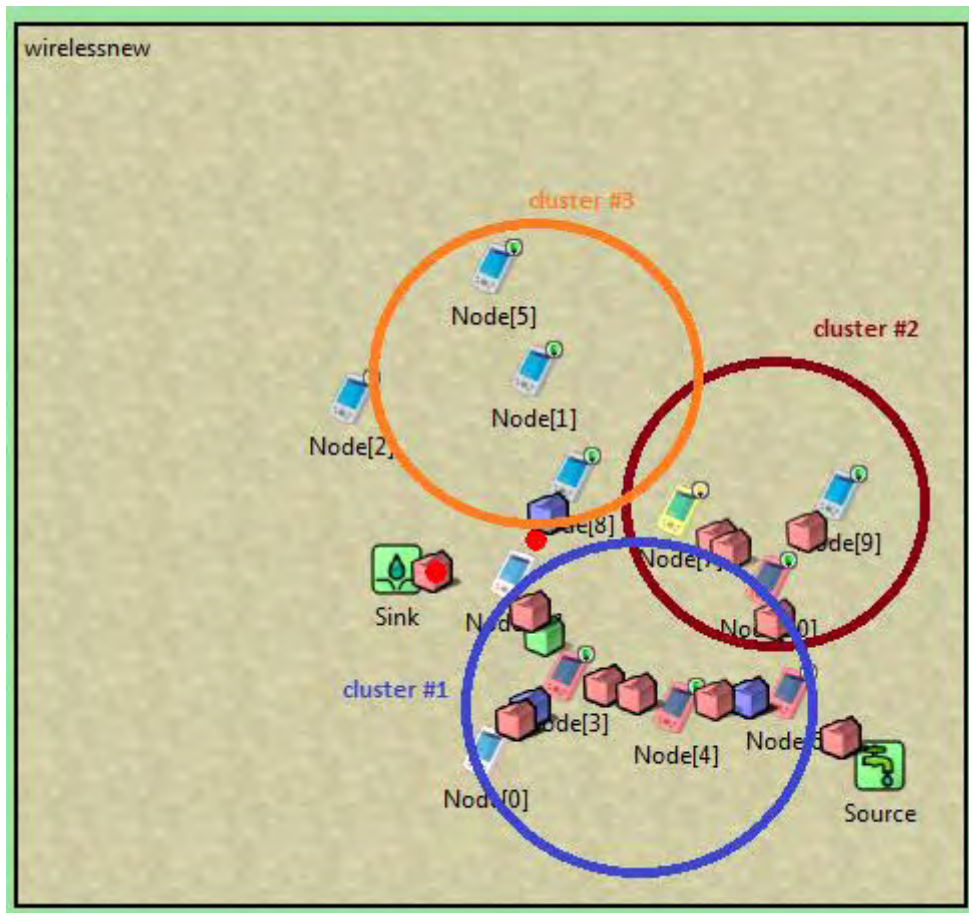
Figure 6.2 Clusters at 300 msec

As more messages are generated and energy resources tend to minimize, as referred above, there are many nodes that change status. This fact causes new cluster, new Cluster Head announcements and new hierarchical level of network.
In Figure 6.3 there are new clusters after 500 milliseconds. Optimal path messages travel from Sink to Source now. Here the difference is that Nodes 8 and 1 are no longer ordinary nodes. They are cluster members, indicated by yellow color. Node 11 plays the role of Cluster Head.
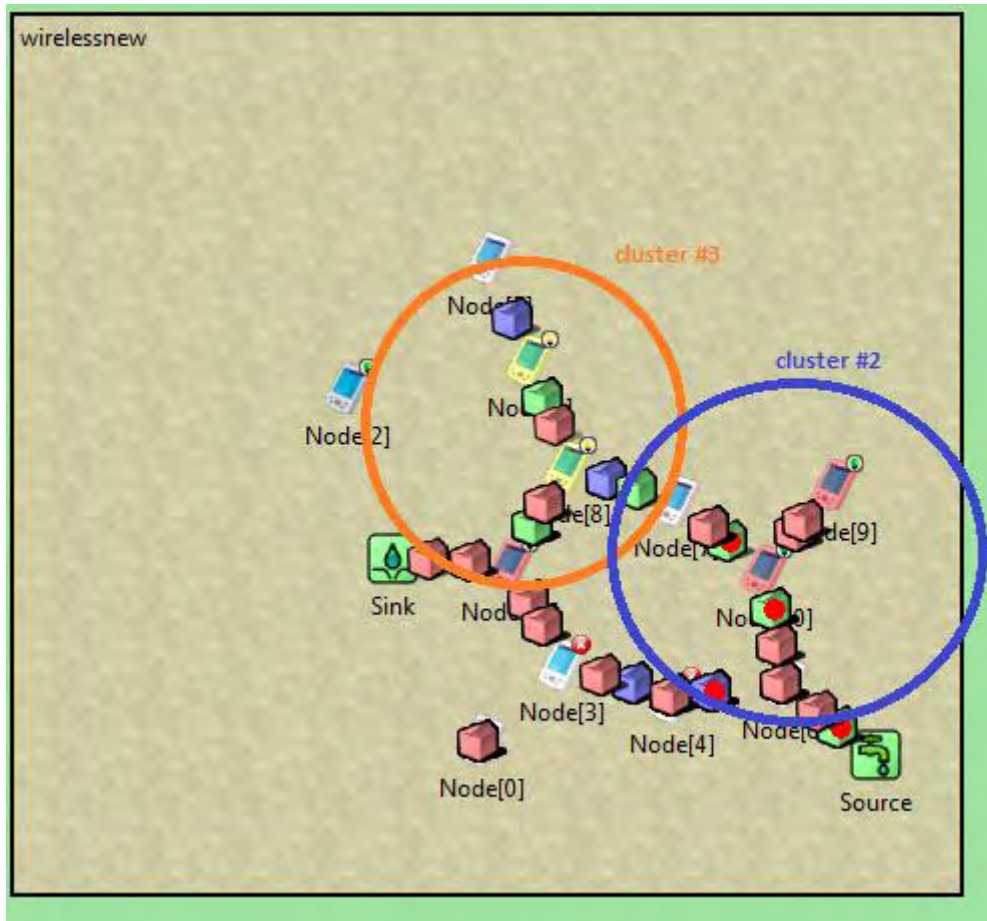
Figure 6.3 Clusters at 500 msec.

6.4.3 Conclusion and Results in Clustering Process

Comparison between previous figures show that many nodes play role of cluster heads (for example in cluster #1 all nodes are cluster heads. To add, Nodes which are farther from Sink, having more hops from Sink, do not participate yet into clustering. This obviously is a negative effect, because all nodes must simultaneously exchange data in distributed way among cluster heads.

After 500 milliseconds we can see that Nodes 8 and 1 participate in a cluster, cluster #3. Here cluster head is the Node 11. Node 11 is directly connected with Sink.

Figure 6.4 Network Design

6.5.   Analysis of Network Performance - Phases of Simulation Model

In this part we will present the stages of network after certain time periods. We record simulation model every 100 milliseconds and we will compare results with general behavior of network performance. In the first part of this analysis there is a presentation of simulation steps. There is a presentation of simulation model in seven steps-seven figures, representing simulation time every 100 milliseconds. We start from 100 milliseconds and after 500 milliseconds there is an overview of ending of algorithm. Every step is explained according graphical icons and interface of model. In Figure 6.4 we can see the network layout of network with 14 modules.

Chapter 6 Analysis of Simulation Model   92

Figure 6.5 Network performance after 100 milliseconds

6.5.1 Initial phase-Network Performance at 100 milliseconds

In this stage we can see that all nodes have full energy battery. This is indicated by green light icon on nodes.(Figure 6.5)

After 100 milliseconds, in simulation time 0,2 seconds algorithm starts to compute the possibility of each node to become cluster heads. In this stage, as shown in Figure 6.6 we can see that nodes 4, 6 and 10 are chosen as cluster heads (in red color). These nodes have

greater probability to become cluster heads because they are connected with greater number of nodes (serve more nodes as cluster heads than others) and they are close to source, which means that they will fast relay data from sink to source (consider source as the final destination). In more details, the factor about distance from destination in our *probabilityCH* value in equation 3.4 has greater weight setting probabilities to become CHs.
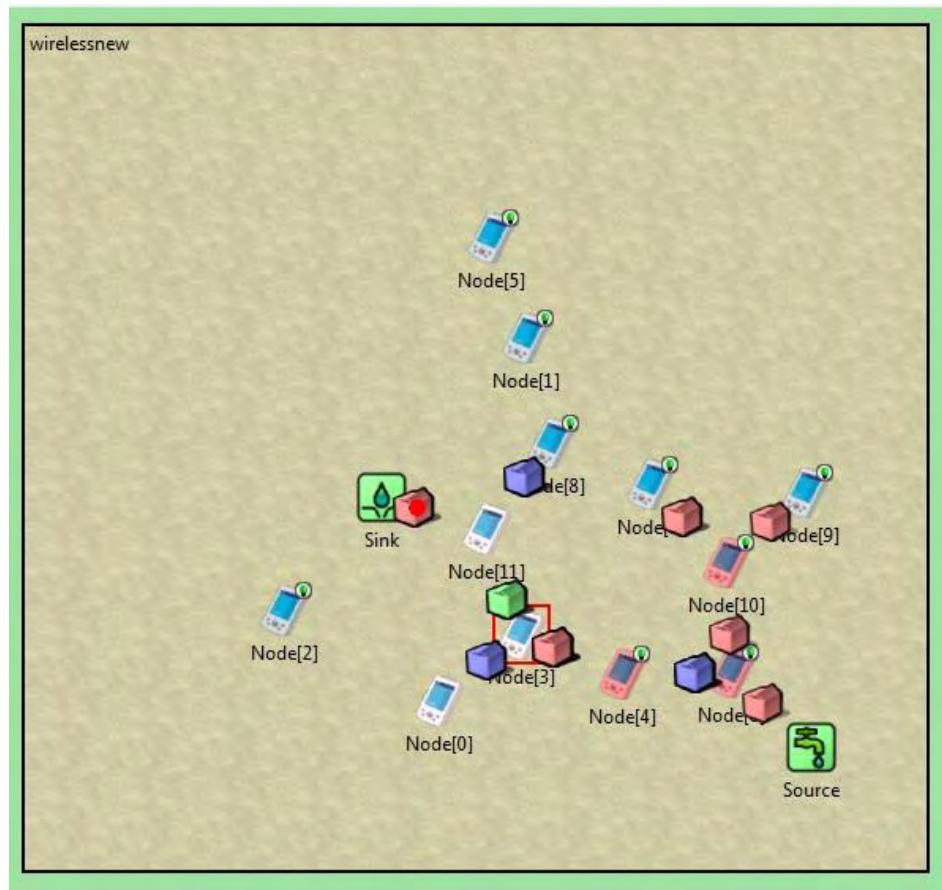
Figure 6.6 Network Performance after 200 milliseconds

6.5.2 Network Performance at 300 milliseconds

In simulation time 300 milliseconds we can see that there is a better distribution of cluster heads. There are now two main cluster head nodes, node 4 and node 10. The node 4 serves nodes of bottom part , specifically nodes 6,3, 0 and 11. On the other hand, node 10 is a cluster head and serves nodes in upper location like nodes 9,7,8,1 and 5. Node 7 is a cluster member which this is indicated with yellow color (Figure 6.7).

So in this part, there is a distributed solution of having actually two clusters in our network, the former about propagation in upper level of location and the latter in the bottom location. Obviously, those clusters are made according distance from final destination, the sink.
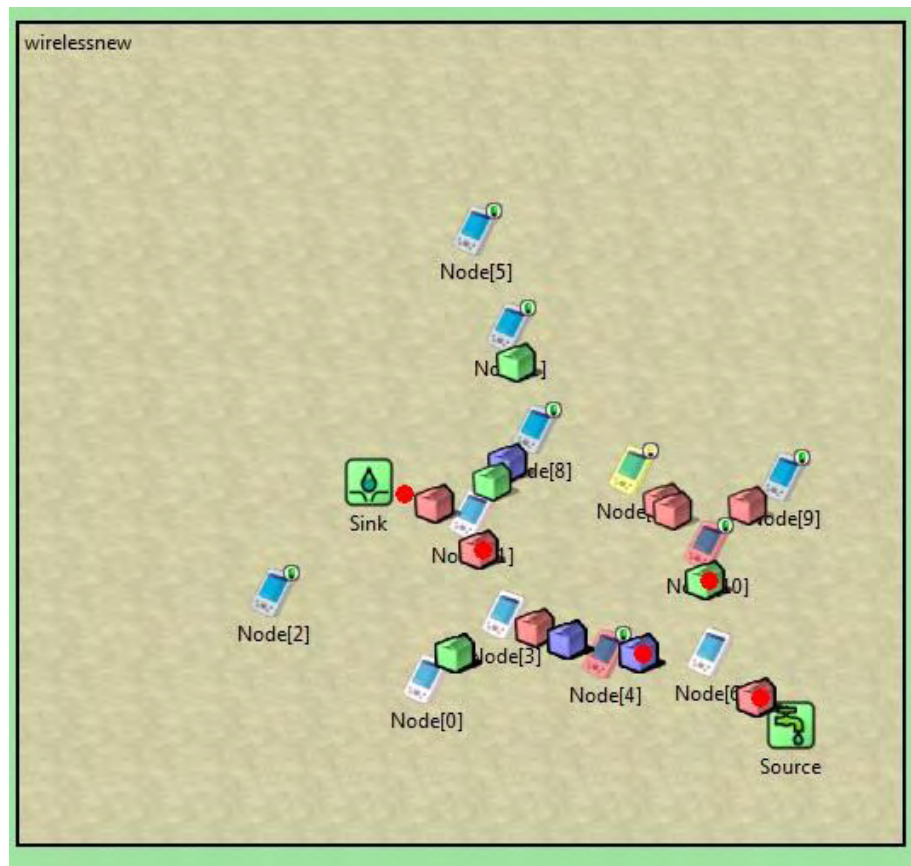
Figure 6.7 Network Performance in 300 milliseconds

6.5.3 Network Performance at 400 milliseconds

In Figure 6.8 we can see the network after 400 milliseconds. As the number of messages increase, nodes calculate again their probability of having the role of CHs. In this phase, we can mention that we have an increased number of nodes become CHs. We can see that in path from Source to Sink, Node 6-Node 3-Node 4- Node 10- Sink, all nodes except Node 6 which is exhausted, are cluster heads.

Although this is unnecessary, because we would expect only one of these nodes play the role of transmitter to all others (cluster head), this happens because now the system tries to find the best optimal path. In this phase, we can see that gradients has been set from the previous state (green packets) and now Sink has accepted Exploratory Data. This is indicated by turning the icon of Sink into Antenna Tower. Exploratory Data are messages that show that we have a path matching a certain event (consider event as a request we have in the network), and nodes that have a match on events can start exploratory data. Those messages will finally reach destination (sink) and after this fact, sink can start reinforcement path by choosing the best path.

The blue packet transmitted from Node 4 to Node 6 shows that the discovery of optimal path has started.

Chapter 6 Analysis of Simulation Model 95
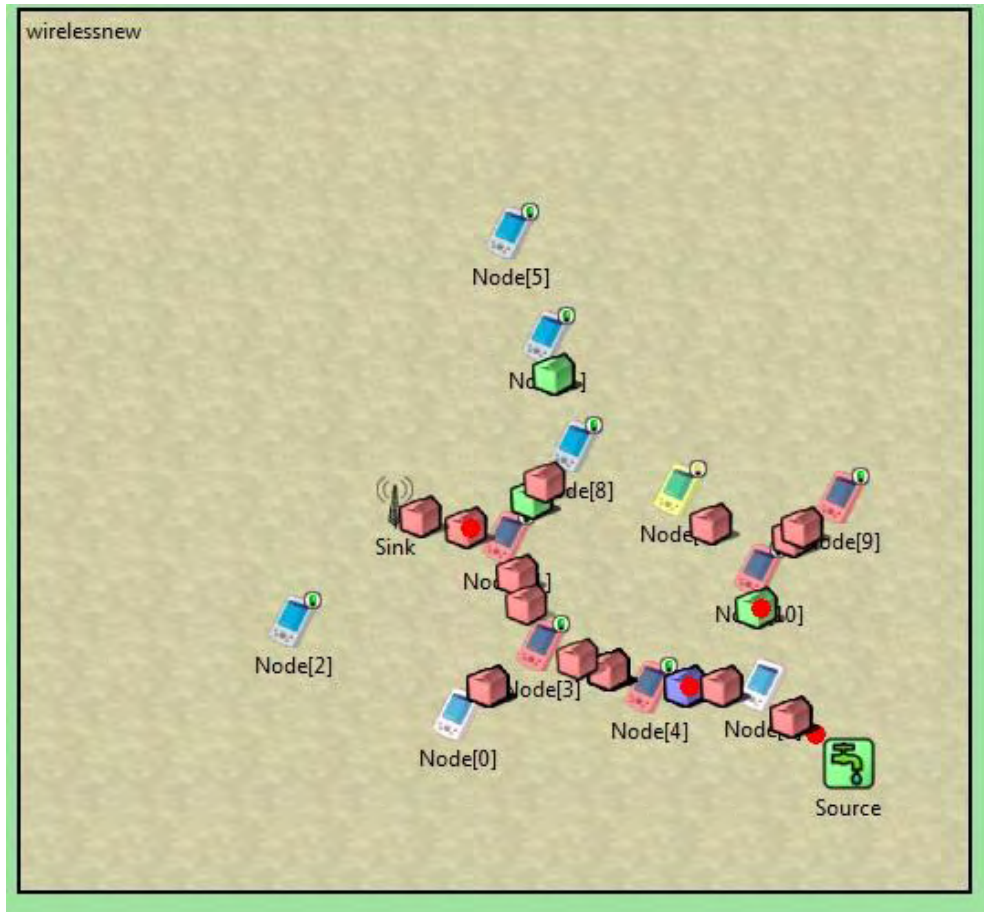
**Figure 6.8 Network Performance in 400 milliseconds**

6.5.4 Network Performance at 500 milliseconds

In figure 6.9 we can see the different types of packets are exchanged in network. Red packets represent data, green packets are gradients to the source of Interest messages came and blue packets are showing the best path. In this figure we can see that there is not yet set the best path, because blue packets travel in the upper part of network, as in the lower level. But, we can claim that the lower part of network (having path of nodes 6, 4, 3, 11) transmit and receive more blue packets. This shows that may this path would be chosen as a more efficient solution for data forwarding.

Concurrently, more nodes lose great amount of energy as Node 6, Node 3, Node 0 and Node 7.
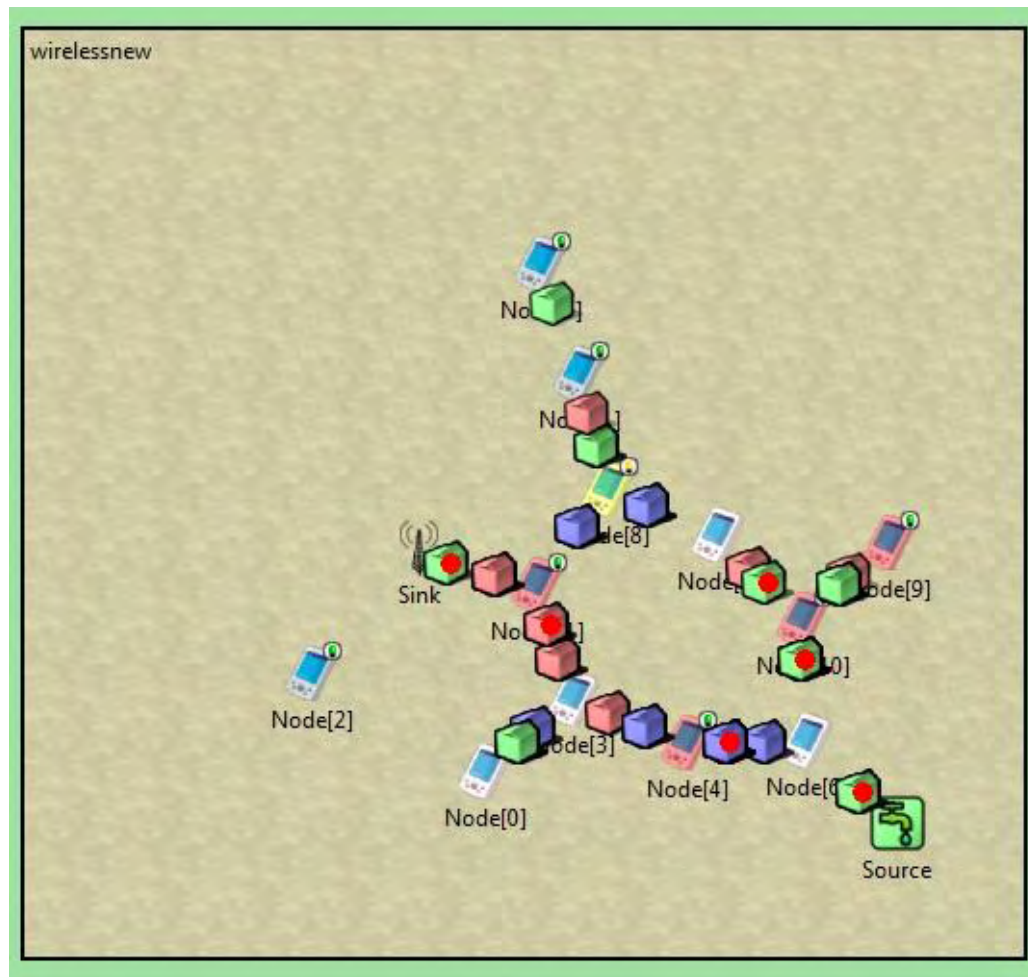
Figure 6.9 Network Performance after 500 milliseconds

6.5.5 Network Performance at final phase-End of Simulation

The two following figures (Figure 6.10 and Figure 6.11) represent the network at the final state, at simulation time of 600 milliseconds. Meanwhile, now the path of Nodes 6, 4, 3 and 11 exchange more blue packets (as optimal path chosen). This shows that optimal path has been established now.

On the contrary, in the first figure we can see that Node 6 and Node 3 are exhausted (white color of Node and red icon of stop operation). This means that the Nodes can not be used anymore. In contrast, other nodes keep exchanging data.

Chapter 6 Analysis of Simulation Model 97
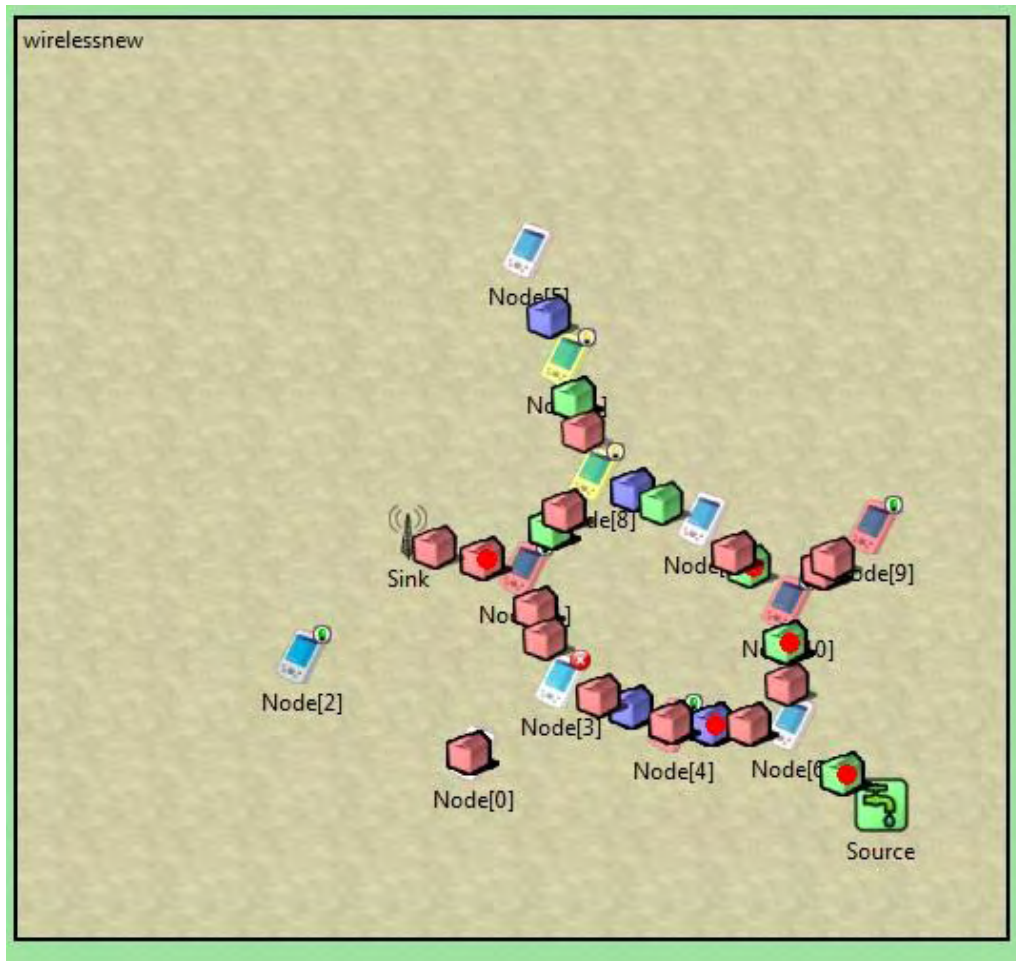
Figure 6.10 Network Performance at 600 milliseconds

Eventually, in the last Figure (Figure 6.11) we can see that all nodes in optimal path are "Dead". No more forwarding of data can be achieved so the algorithm stops and presents the corresponding measurements about energy consumption and number of generated and exchanged messages. Red arrows show that Nodes 6, 3, 4 and 11 are exhausted.
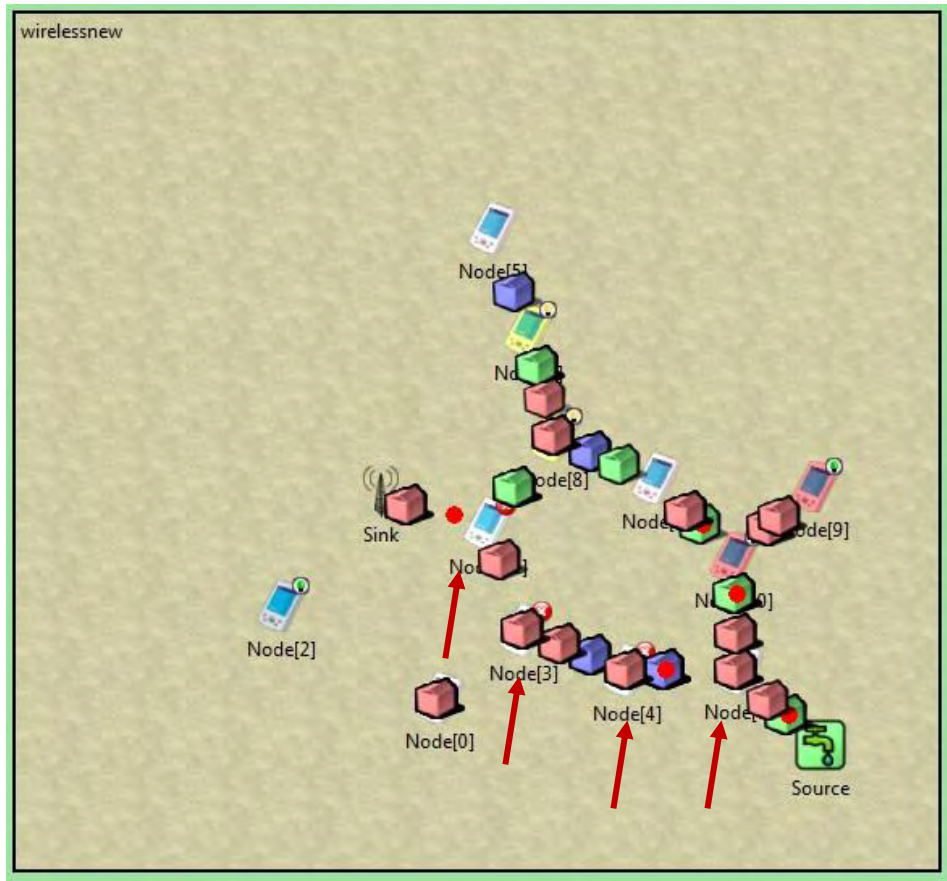
Figure 6.11 Network Performance at the End of Simulation

6.5.6 Discovery of optimal path in communication between source and sink

One of the most important results of this method is to examine and compare the results about optimal path. In our model there are two directions about which is the best nodes to follow in order to:

- Source reaches Sink
- Communication between nodes

The nodes that are presented to are advanced and play crucial role in network communication are:

1. Node 6-Node 4-Node 3-Node 11
2. Node 8-Node 1

The first route shows how data can reach destination (source) and the second shows what Nodes are best in the upper location of network for message propagation. The DDRA design chooses the <u>first route as optimal</u>.

In this part of simulation, the best path has been discovered and is indicated in the graphical runtime Interface of Omnet++. In the following Figure 6.12 there is a mark showing the node chosen by algorithm as best for forwarding data. In Figure 6.13 there is a mark showing best next node, similar as previous Figure 6.12.

Chapter 6 Analysis of Simulation Model 99

Figure 6.12 Tkenv Environment of Omnet++

Figure 6.13 Tkenv-Optimal Path

Chapter 6 Analysis of Simulation Model   100

6.6 Conclusion

In this section we described the network performance. After message exchange, optimal path messages show the least cost path in energy terms. This path also is created considering topology of network, node degree and based on data dissemination. Data are specific, are injected as requests and have fields to discover only sensors that can "answer" and "sense" this kind of data.

Unfortunately, the selected optimal routing path turns into path consist of "Dead" nodes rather quickly, due to large amount of control messages and continuously re-clustering procedures.

Chapter 7

Result Discussion

This chapter is about results analysis based on QoS and mainly on energy efficiency of simulation model.

7.1 Introduction-Performance Metrics

In this part we will analyze and present the experimental results of our simulation model. The data collected by OMNeT++ version 4.3 . Statistical significance was analyzed using the Analysis Editor of OMNet++ and Microsoft Excel 2007.

All sample data are categorized into four parts, depending on their meaning.

The performance of DDRA for WSN can be evaluated using metrics such as energy efficiency, throughput and QoS (packet loss ratio, end –to-end delay, delivery packet ratio). DDRA for WSN is a routing protocol that should provide end-to-end QoS in an energy-efficient manner.

In this section we will show and compare the results we have, working on our simulation model. We will analyze and compare simulation results according:

- throughput of network
- number of messages that modules exchange
- energy consumption of each module and total of our network
- percentage of dead nodes after simulation
- delay and delivery packet ratio

To understand and compare the results we have to examine the location of each node in the network. Nodes near Sink, and positioned in central location, tend to have more changes to play the role of Cluster Heads. To clarify, nodes near Sink tend to receive more messages than other nodes and have a better throughput compared with nodes which are located in further distance, and which are connected with small number of neighbors. On the contrary, energy consumption is increased in those entities.

7.2 Energy Efficiency

7.2.1 Energy Consumption

Energy Consumption of network model concerns two aspects: Energy Consumption of each Module and Energy Consumption of overall system. Firstly, to identify performance of network in energy cost of modules we have set energy cost at each message transmission of every message nodes receive or forward. In methodology choice, the Equation 4.6 give us the description about energy reduction during communication between nodes i and j. Secondly, to evaluate the overall energy consumption of system we consider the average value of energy consumption of all nodes. This will reveal the behavior of system and the success of the approach as an energy efficient solution.

Energy consumption in the system varies extremely between nodes. Figure 7.1 shows the energy cost of each module after 300 millisecond. This corresponds to the middle of simulation performance.
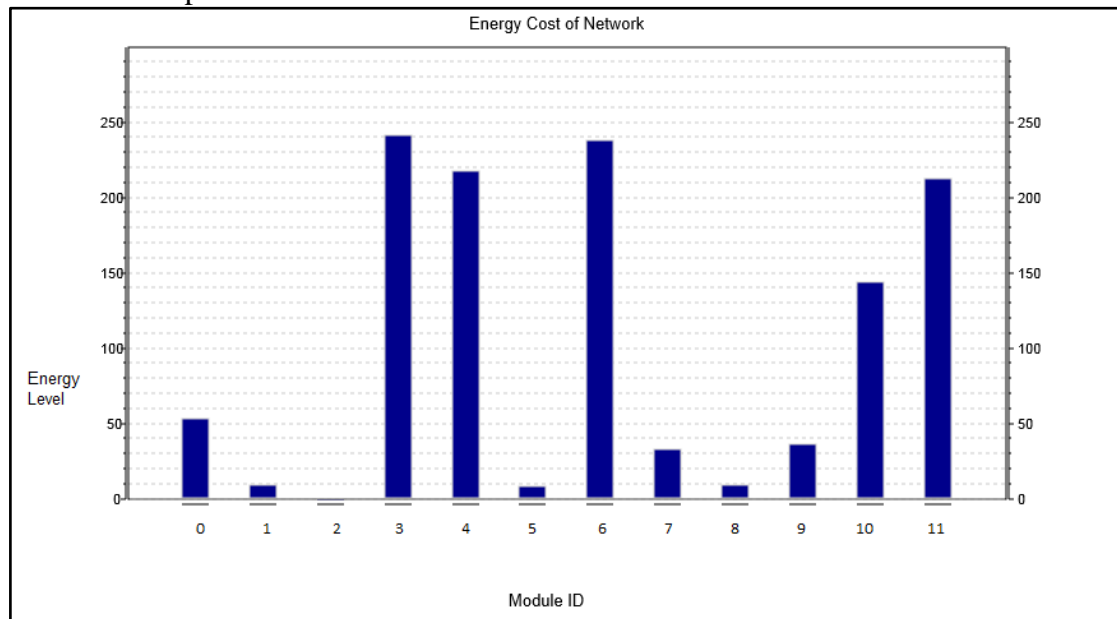
Figure 7. 1 Energy cost of Each Module

As Figure 7.1 shows Nodes 3, 4,6 and 11 spend great values of energy units. These values are above 200 energy units(this corresponds to 4000 mJ). Node 10 spends also high energy but much lower than those four nodes. All other nodes spent lower energy units and Node 2 spends no energy at all. Node 2 for network is an unconnected node and receives or transmits no messages.

Changes in energy consumption reveal that Nodes 3,4,6 and 11 spent four times greater energy units than other nodes. Actually, those nodes belong to the optimal path that algorithm produce. In addition, these nodes are the first to play role of cluster heads.

This finding validates the assumption of extreme usage of cluster head nodes. However, this significant discrepancy between nodes in optimal path and other nodes would affect negatively the network performance. In small time period, these nodes will be exhausted and alternative method is required to change routing from Source to Sink.

7.2.2 Energy Consumption of Each module
The following Figures (Figure 7.2 and Figure 7.3) show the residual energy resources on each module. The former is about energy level left after 300 millisec of simulation and the latter refers to energy at end of simulation.
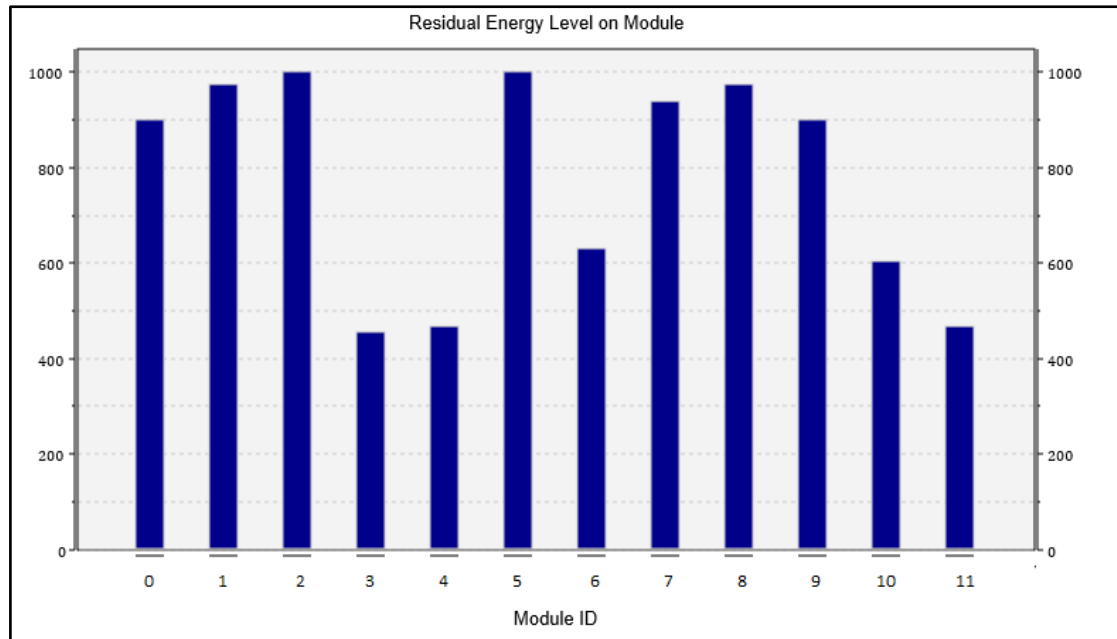
Figure 7.2 Residual Energy after 300 msec

In Figure 7.2 we can see that Nodes 3, 4, 6, 10 and 11 spent twice energy units than other nodes. The other nodes have greater amount of energy. As described previously, energy consumption is related to the number of incoming and outgoing amount of messages.

At this time period Node 4 is a cluster head (Figure 6.7). This means that must accept and retransmit a large amount of data. Nodes 3, 6, 10 and 11 have limited energy. This findings confirm the representation of nodes as white nodes, interpreted as "weak nodes". By methodology, at this phase of algorithm, clusters have been set and nodes that are elected as cluster members only accept messages. The network overload increases on the CHs.

The residual energy of nodes at the end of simulation (after 600 millisec) fluctuates between 0 energy units and approximately 900 energy units. Although this variation is very large, only four of nodes tend to have a "strong battery " remaining after simulation end. Figure 7.3 presents the remaining energy units after stopping the simulation model.
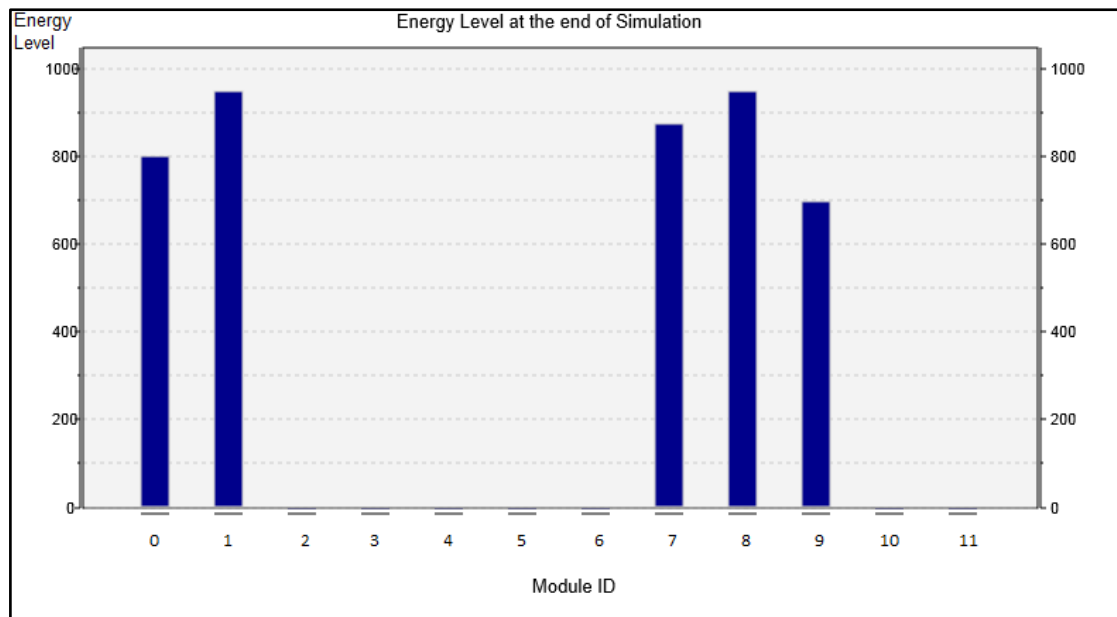
Figure 7.3 Energy Level at the End of Simulation

Levels of Node 0, 1, 7, 8 and 9 are high and above 700 energy units. The nodes which belong to the optimal path from Source to Sink   (Node [6]-Node[4]-Node[3]-Node[11]) are exhausted, hypothesis expected according our algorithm. Nodes near Source and Sink tend to spent more energy and tend to be terminated more quickly. Our technique shows clearly a "hot spot" problem, a problem in which energy might be not distributed equally in all nodes. It is fundamental to note that modules near the source and the distance are not able to achieve a "fair" energy cost, a balanced energy consumption otherwise, and this kind of energy cost is strongly related with node position.

7.2.3 Energy Consumption of overall system

The samples highlight an extreme variation between nodes according their location. The overall energy consumption will reveal the general behavior of our system. This approach concerns the energy consumption and its main goal is to examine a new energy efficient solution for wireless sensor networks.

Assume that each energy unit corresponds to 20 mJ.

The average energy consumption can be measured as energy units divided by simulation time. The network model spent 1209 total energy units in 600 milliseconds. This corresponds to 24180 milliJoule. This result identifies that 40300 mJ are spent every millisecond or else 40.3 J/sec. This value can be interpreted as 0.144 J per unique event in 12 nodes. Apparently, this value seems extremely large. In [2] authors achieve a large reduction in energy dissipated. PCDD algorithm uses only 0.04 J/sec per unique event in a 50 nodes network. Further discussion will be presented in next chapter.

Simulation models that varies in number of nodes, showed another result about energy cost. A network having 32 nodes spends 110,96 Joule/sec. In same manner, a network consist of 64 nodes spends 50,64 Joule/sec. These values can be formed as 0,301 Joule per unique event for former network and 0,177 Joule per unique event for latter system. This shows that when number of nodes increases the energy cost decreases.

Chapter 7 Result Discussion   106

Nonetheless, this fact seems wrong because more nodes will generate and forward more messages. We can mention that here we have a fixed-area scenario but topology is different and node density varies from small simulation model. These two factors affect strongly the network performance. Also, here we must refer to assumptions about largest model. The network's simulation time is slightly shorter, end condition still remains that when node near initial source is dead, system stops, and eventually nodes of other neighborhoods do not have the same delivery rate with our initial system.

Another point of view is to measure the behavior of algorithm is the percentage of "Dead" nodes at the end. Figure 7.4 present that 64 per cent of all nodes are exhausted after 600 milliseconds. By methodology, the stop condition was the fact of data could not reach final destination. In other words, when all nodes in optimal path got out of energy, communication between Source and Sink was no further feasible.

Rapidly energy waste of nodes in optimal path contributes to have an important amount of nodes having energy remaining but not used for a long time as alternative path to reach final destination. This amount reaches 34% of total nodes in model.
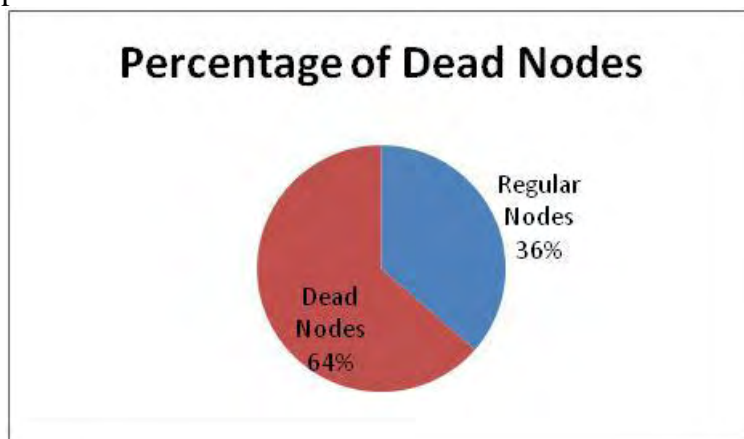
Figure 7.4 Dead Nodes of Network

7.2.4 Energy Consumption as a Function of Network Size

In order to examine the energy consumption of system, the implementation of two other systems is necessary. The first system has 32 Hosts and has two Sources and two Sinks. The second network has 64 Nodes with three Sources and two Sinks. We suppose that second sink can communicate with two sources.

Now there is a presentation about topology of these two models. Topology of the networks is quite different. In medium size network, nodes are deployed as a chain-topology between sources and sinks. There are parts of nodes in chain topology and these parts are connected between them. In the largest size network, nodes are placed like groups around sources and sinks. Each group seems like mesh-topology and node density is high. (fixed-area scenario).

The methodology chosen is similar to previous model. Here we try to explore how nodes communicate and what are the energy resources during communication phase when number of nodes is increased. The end condition of simulation remains the same with previous model, which is that when nodes directly connected with sources and

Chapter 7 Result Discussion 107

sinks are exhausted, no more gradients can be maintained and created, so communication between source and sink is no more feasible.

In figure 7.5 we can see the structure of system with 32 Hosts and in figure 7.6 we can see the system of 64 Hosts.

Figure 7.5 Network with 32 Sensors
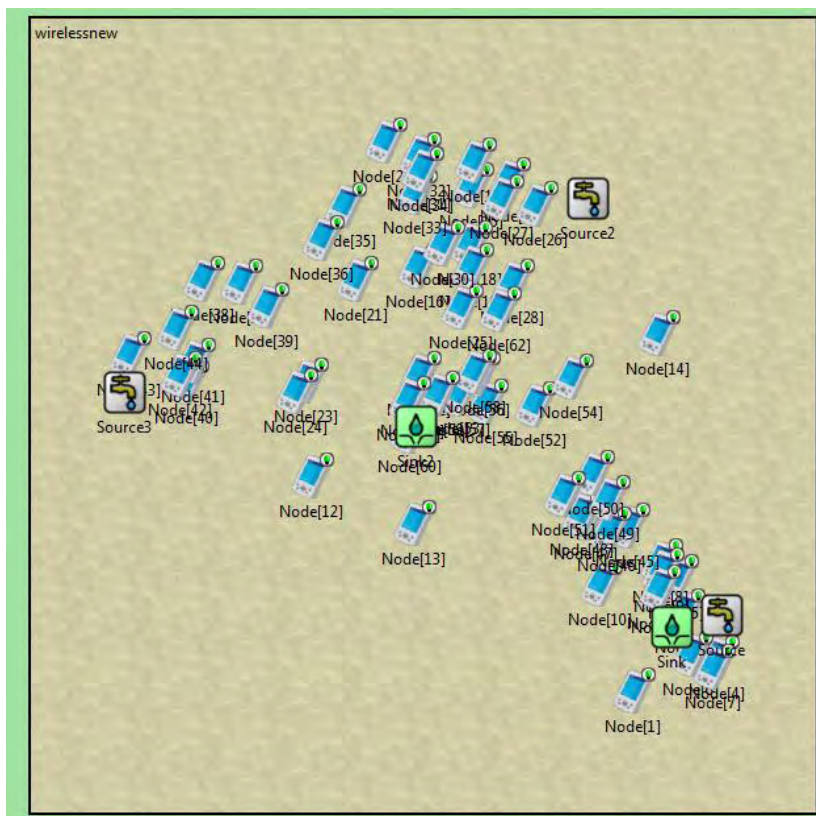
Figure 7.6 Network with 64 Sensors

Chapter 7 Result Discussion 108

The most interesting of these two systems is not the whole function and propagation of messages. Obviously, here there is an extremely greater amount of messages created and forwarding. The basic idea of these two systems is to show the energy consumption when size of network varies . In the following figures (Figure 7.7, Figure 7.8, Figure 7.9) we will show the energy consumption of three networks, according their number of nodes.
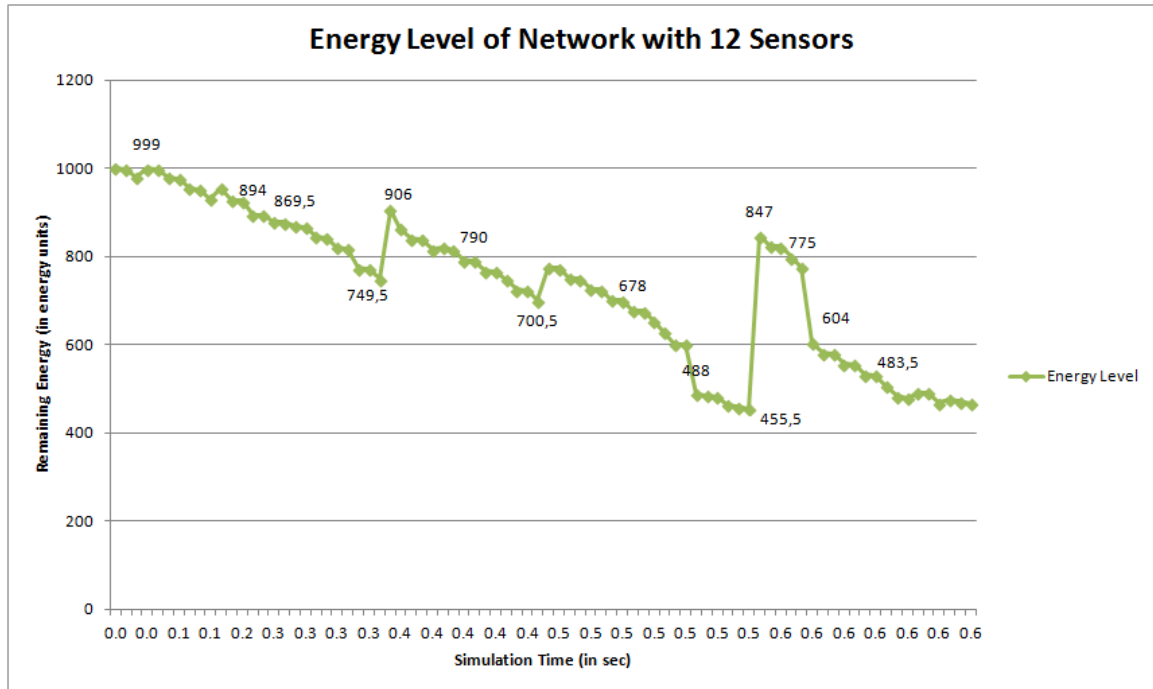
Figure 7.7 Energy Consumption of model with 12 Sensors
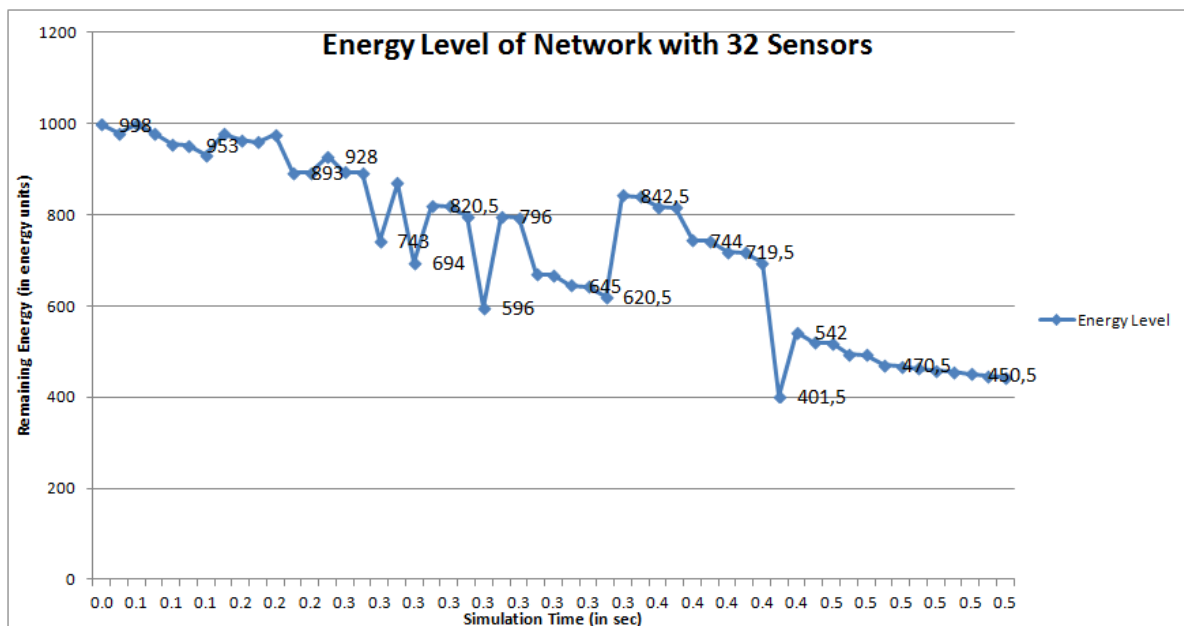
Figure 7.8 Energy Consumption of model with 32 Sensors

Chapter 7 Result Discussion   109
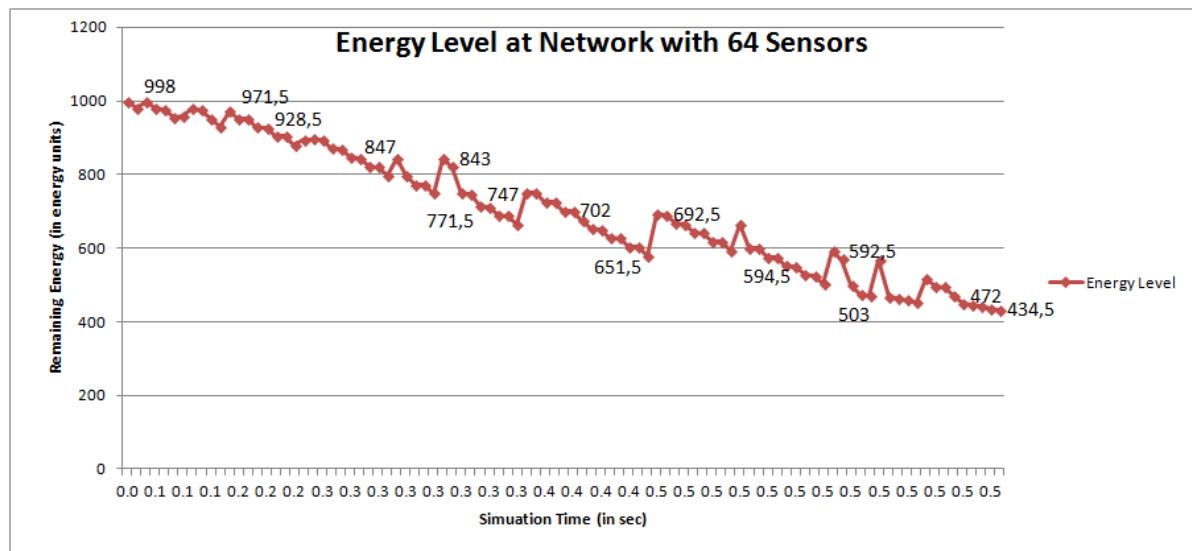
Figure 7.9 Energy Consumption with 64 Sensors

As figures indicate, the average energy level of systems tends to decrease during simulation time. This is obvious because in our system, the energy decreases according number of messages are exchanged, operations like clustering and setting gradients cost in energy terms and generally nodes, especially those that retransmit messages and are close to sources and sinks, tend to loose more quickly their energy resources.

At this point we must refer that those figures represent mainly, the nodes that are mostly active, nodes that become cluster heads and propagate the most number of messages.

One more significant fact is that energy consumption of network with 16 and 64 Sensors are similar but there is a slightly difference from figure that shows energy for network with 32 Sensors. In this figure (Figure 7.8) we can see more fluctuations and at simulation time of 0.4 seconds there is a steep reduction of energy resources. This can be translated as the creation of more cluster heads at different parts of network. To put it another way, as more cluster heads are elected close to the other sources , more energy resources of these nodes are consumed.

Another interesting point to emphasize is the sharp decreases of energy level that second model shows. This is due to topology pattern. Topology in this network shows that nodes are close to source and sink. The nodes are located like a big chain having many nodes in the upper and lower level. As nodes are near source, there is more probability more nodes to be selected as cluster heads. To sum up, as topology indicates, nodes that are set in the center of simulation area are not used very effectively. These nodes tend to remain ordinary nodes and clusters are constructed near sources but not distributed to save energy and create subgroups for the parts of network.

Moreover, we can highlight that a topology of nodes contributes to greater energy reduction and generally, this approach shows better results if nodes are located at larger distance, placed in distributed positions and farther than others and not gathered near source and connected in chain topology.

In contrast, the energy consumption of first network and the large network shows that nodes location plays crucial role in network performance.

In addition, networks with greater number of nodes, tend to have a larger packet loss ratio. This is an expected phenomenon according the idea that not all nodes participate equally in message exchanging and message forwarding. Only a small part of nodes react with sources and sinks and a large number of nodes are inactive. This assump-

tion, helps to understand that networks do not operate efficiently and many nodes do not even consume none of their energy units.

All previous observations, leads to the idea that network topology, methodology and clustering is affected strongly by number of nodes. This, unfortunately, shows that our approach is strictly connected with number of nodes and topology of network. These two factors indicate that DDRA had to prevent unequal clustering and unfair usage of nodes. These two disadvantages actually exist both in larger networks.

7.2.5 Comparison with other approaches

In this part to evaluate the disadvantages and advantages of DDRA for WSN we will compare our results with DD and PCDD algorithms. There are three main factors to compare on these approaches. The average energy dissipated of system (in Joule), the average energy cost on single sink scenario and average energy cost on multiple sinks scenario.

In Figure 7.10 we can see the comparison between those three approaches. *Fixed Area Scenario* is a condition where simulation area is constant but average node degree in the network varies. That means that node degree of each module is different but network area size remains constant and is the same for all types of network models (WSN 12, WSN 32, WSN 64).

As authors highlight in "*PCDD*" approach " *DD and PC is a pure local interaction*",[2] which means that increase in number of neighbors, leads to increase of Interest propagation and exploratory data exchange.

As Figure 7.10 present, the results about energy efficiency of DDRA are not quite improved compared with two other protocols. Network system with 12 Sensors consume greater amount of energy than PCDD. To continue, when 64 nodes exist in the system there is a great reduction in energy cost, compared to medium size network.

On the other hand, system consists of 32 nodes tends to increase energy consumption, compared with smallest network of 12 sensors. Increased energy cost is relevant with increased number of nodes as more messages are generated and exchanged in network.

Remarkably, network with 64 nodes seems to be more energy efficient than network with 32 nodes. Actually, the node density here is greater. Communication between source and sink is a multi-hop communication but distance between nodes (hops actually as we have hop-by-hop transmission) is less. Nodes are closer to each other, more clusters have been created and we have two sinks and three sources. There is not a large scale energy-hole near Source1 and Sink1. Data and control messages are exchanged in many other parts of network area. That creates a more distributed traffic flow and clusters contribute to a better balance of energy consumption.

7.2.6 Network Size gains Energy Saving

On the other hand, we can summarize that system with 64 nodes consumes less energy level than smaller system. This is quite unexpected. DD and PCDD increase their energy cost as network operates with more nodes.

This phenomenon in which network with 64 nodes consumes slightly more energy than the network with 12 nodes is an advantage of our implementation. As network of 32 nodes does not achieve great energy-efficient operation, while number of nodes

are duplicated there is a significant reduction of energy waste. Many reasons can explain this function:

1. In the 64 nodes network, there are more nodes participate in clustering and message exchanging. Especially, we can see that near Sink2 there are many cluster members nodes.

2. Consequently, more clusters are created. This leads to better and "fair" distribution of energy consumption to nodes that are not very close to initial Source. In previous models, the network tends to create a "Hot-Spot" near source, causing quick exhaustion of cluster heads near this position. For example, Node 6 (directly connected with Source), is very active and plays the role of a great "bottleneck" of several message forwarding in other two network models.

3. In corresponding model there are two sinks and three sources. This means that network load is not only on one single route, but here there is an effort to create optimal paths involving nodes in further location, sparse located in area. This helps to avoid an extremely exchanging of massive messages on few nodes, causing much delay and perhaps many collisions. The amount of messages in other clusters can travel more easily and in higher delivery rate.

4. Another point is that the topology of networks is considerably different of two other networks with smaller size. In larger network the nodes are located almost in five different groups, clusters. The nodes are located like mesh topology but separated into five main subareas. In this way, nodes communicate with others in their local area and the hops of transmitting a message are limited (intra-cluster communication). On the contrary, network with 32 nodes has actually chain topology.

5. In addition, the location of sources plays a crucial role in advance. Sources are located sparse and further each other. Suppose that, the first network unfortunately suffers from "Hot-Spot" problem, here sources are far from each other. To communicate with Sink (placed almost in the middle of distance between two sources), there are many nodes to forward data, there is a multi-hop communication but the more nodes participate, the more energy save we gain because data can be forwarded by many alternates routes, not few and particular nodes keep extensively used. Distance between nodes is less as node density is much greater.

6. To conclude, our DDRA has as main idea to create clusters which are firstly in energy terms feasible to operate, close to Sink and have less hops in a cluster. By getting more subgroups we manage to have more cluster heads, communication with limited number of hops and distribution of energy consumption by sharing network load to many nodes.

Here we must point that network with more nodes, tends to complete its operation slightly sooner (stop at 500 milliseconds), and some nodes at the end of simulation are not used.

In other words, many nodes are inactive, but the procedure is the same as the previous models and for reasons explained above we result in a better solution.
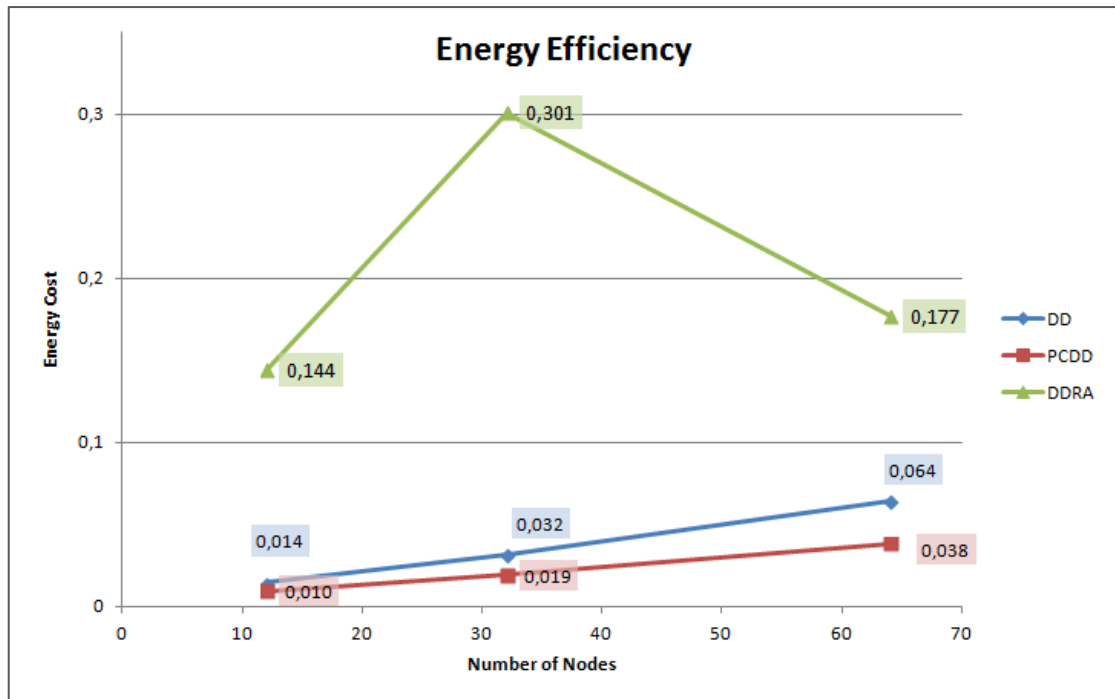
Figure 7.10 Energy Efficiency Fixed-Area Scenario

Generally, authors in [2] managed to prove that when a network is denser, the energy cost is less because many redundant paths can be avoided by PC. The results provided by 64-nodes system, confirm this conclusion.

On the contrary, in 32-nodes system the energy cost sharply increases. The chain topology affects strongly clustering and reduction of energy cost is not achievable. According the methodology and the messages travelling into network, energy cost increases as nodes increase but when chain topology is used and node density is high we can have a better performance.

7.3.1. Energy Consumption as a Function of Traffic Pattern

In next paragraph we will discuss energy consumption depending on number of sources. The PCDD algorithm achieved to decrease average energy cost when number of sources varies. More sources, means more Interests, more exploratory data flooding and to sum up much more network load. In DDRA we have a system with two sources on network of 32 Sensors and three sources on network of 64 Sensors.

The more sources we have, the more events we have. This can be explained as network with 32 Sensors ends at #368 event number and network with 64 Sensors ends at #285 event number. The initial network (the smaller) ends up at #278 event number.

In case where we have more sinks, but single source , the messages are approximately the same because source is the main generator of messages in the network, starting with Interests. On the other hand, more sinks leads to more paths, more updates on gradients, and more Events, so the network load increases as the energy cost increases too.

In DD and PCDD approach, the major discrepancy in energy metric happens when we have more sinks. Scientists issue that although data are not many more, number of

Chapter 7 Result Discussion   113

events is smaller, messages have to travel in more paths (more destinations) and if we consider that Interests are more due to many destinations, the remaining energy is less and energy cost is significant. Especially in DD there is a big difference in energy cost. Although PCDD manages to maintain energy cost in low level for both traffic patterns (increased number of sources, increased number of sinks).

In figure 7.11 there is a comparison on average energy cost between DD and PCDD when sources varies and afterwards the Figure 7.12 shows the same metric but now changing the number of sinks. As reported before, the main difference is when sinks are more. Sources do not change in large scale the performance of network because we count the energy per event and events are more when more sources exist.

In contrast, when sinks are increased (for example for two sinks), there is a significant discrepancy because more messages have to travel in more paths and more updates must be done on the gradients of each node. Undoubtedly, this occurs for another reason too. Sink in our design floods network with Interest, which means almost duplicated number of Interest transmission.

In this part we must emphasize to generator of Interests. Although in DD and PCDD approach Sink is the module of generating and flooding network with Interest, in our design this is done by Source. We have chosen source, in combination with our CH election scheme.
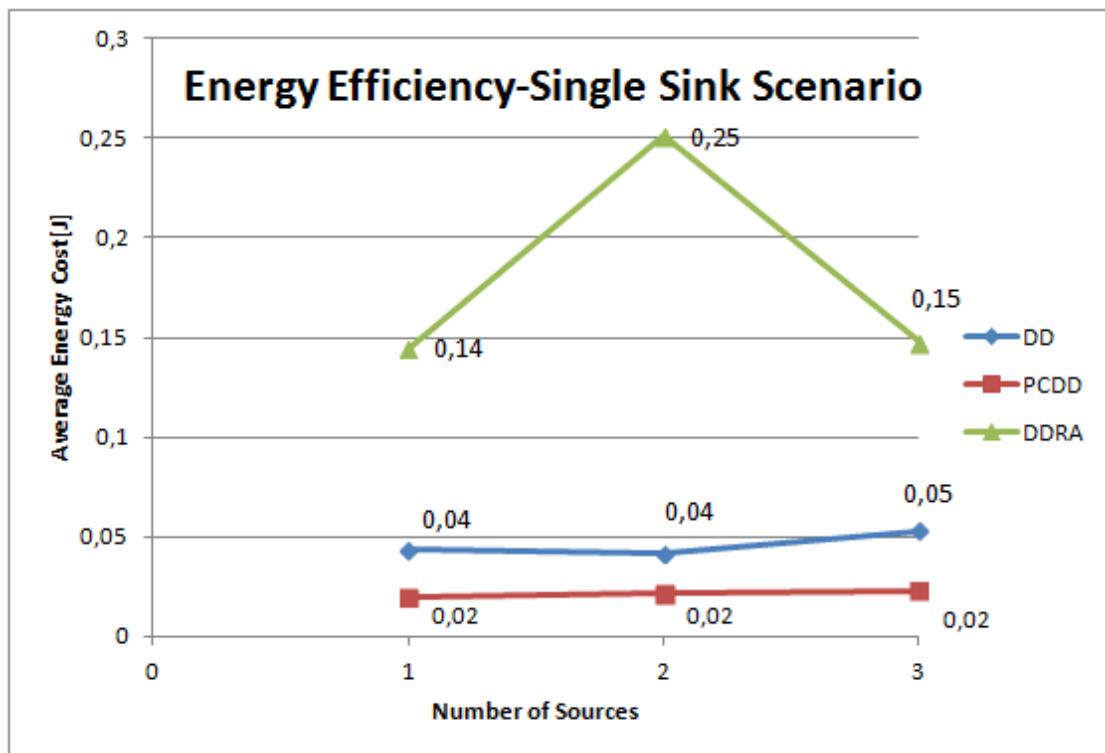
Figure 7.11 Energy Efficiency on multiple Sources

Chapter 7 Result Discussion 114
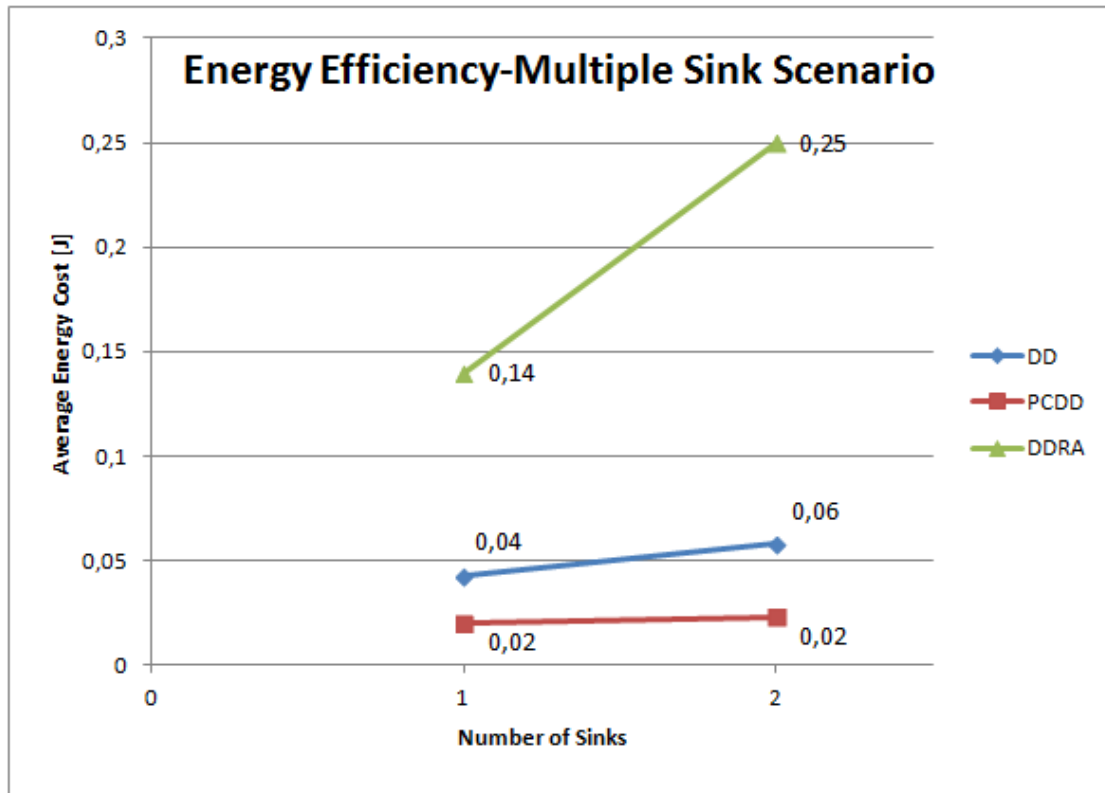
Figure 7.12 Energy Efficiency on multiple Sinks

7.4 Throughput of Network

Throughput refers to the amount of data transferred from source to destination given a specific amount of time. In our model, the amount of data that start from source and reach destination, specified as Sink in our case. Table 8.1 shows the number of messages each node exchange according their type.

Module ID	Data Packet	Interest	Exploratory Data	Gradient	Optimal Path	Total
Node[0]	5	18	405	398	404	1230
Node[1]	0	0	0	0	0	0
Node[2]	0	0	0	0	0	0
Node[3]	11	0	923	916	922	3703
Node[4]	8	0	383	376	382	2152
Node[5]	1	0	904	897	903	2705
Node[6]	14	0	818	808	814	2454
Node[7]	3	0	636	629	635	1903
Node[8]	0	0	0	0	0	0
Node[9]	0	0	0	0	0	0
Node[10]	0	0	0	0	0	0
Node[11]	4	0	439	432	438	2312
Source	0	29	0	-	-	29
Sink	480	0	0	-	-	960

Table 7.1 Number of Messages in Network Communication

It is obvious to mention that Node 3, Node 4, Node 5, Node 6 and Node 11 exchange the greatest amount of data among all other nodes. In more details, they transmit and receive 2665 messages at average.

Chapter 7 Result Discussion 115

On the other hand, Sink exchange 960 messages and Source has only 29 messages. This large discrepancy identifies that Sink and Source haven't got a balance of data exchanging and emphasize the great packet loss ratio network shows.

In the next Figure (Figure 7.14) there is a presentation of the data messages each module creates and forward. This is the total amount of messages during communication phase. The messages here do not include the data packets that are exchanged in the final phase, the data packets that follow the nodes in the optimal path. There are all data and control messages of communication, but data packets are not included. This is done due to focus on control messages before routing establishment.

Similarly, there is a huge difference between Nodes 1, 2, 8, 9 and 10, which have no messages. Seems they are inactive and do not participate in the process of communication. This can be explained as they are not chosen as Cluster Heads.

Moreover, all types of messages fluctuate in same level except Node 11, which holds one of the biggest amount of data messages. All other kind of messages (control messages) hold a much lower level. Node 11 is the last module before messages reach sink. As Source generates Interest to flood the network, simultaneously data messages keep travel. Finally, data messages travel until they reach their initial generator module (where we count them and we delete them) and travel until they get to Sink. Node 3 and Node 8 keep sending data messages to Node 11.

Node 11 actually is the receiver of two main communication path of network. The path created by nodes on upper level, and the path created by nodes on lower level (Figure 7.13).

Indeed, Node 11 ends up to be a "bottleneck" in the network communication.
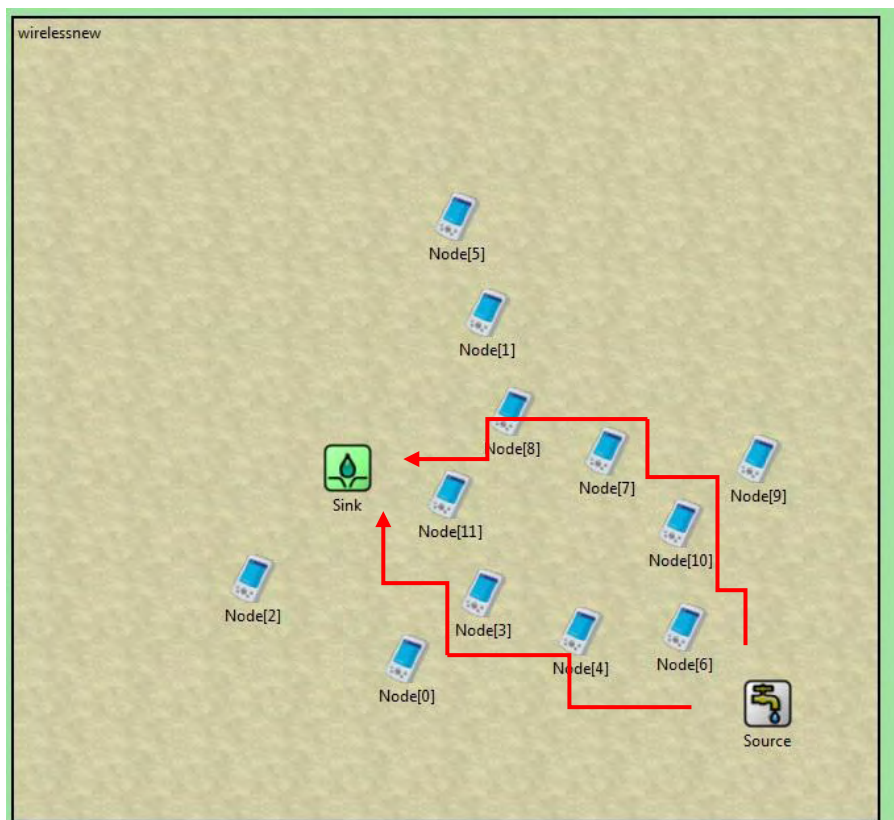
Figure 7.13 Paths from Source to Sink
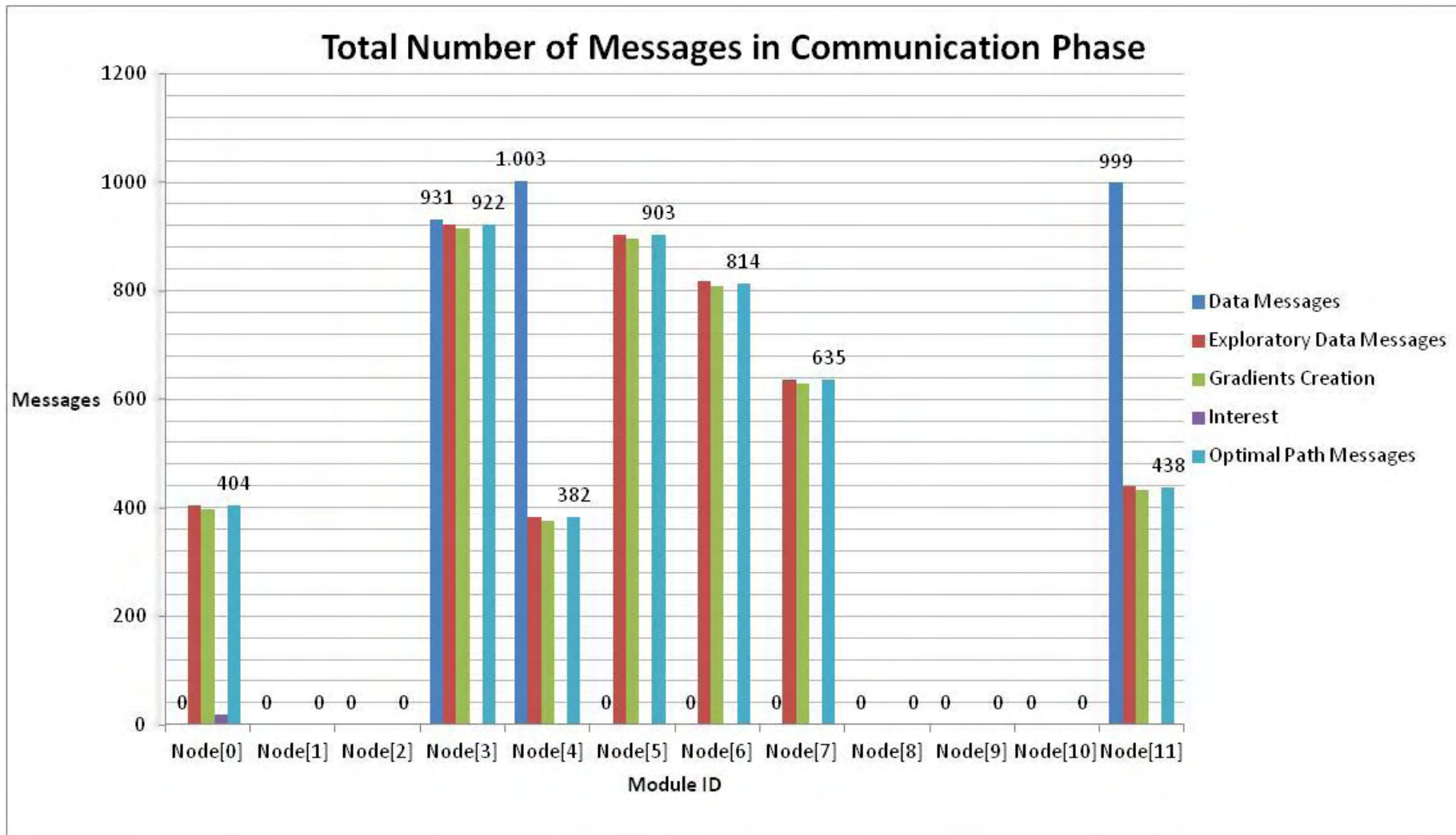
Figure 7.14 Amount of Messages During Communication

The sum of all messages is approximately 4.400 messages. Regarding that data messages have 100 bytes length, control messages have either 25 or 50 bytes, the amount of data reaches at 4.962.800 bits. This can be interpreted as 4,73 Mb (Mega Bits) and 7,88 Mbps (Mega bits per second).

Typically a network of 802.11b Wi-Fi standard has 11Mbps throughput and of 802.11g Wi-Fi standard has 54Mbps [47]. The throughput of 7,88 Mbps is rather a small value, which do not corresponds to high speed networks of nowadays. A typical WSN based on IEEE802.11b has almost 11.0 Mbps.

7.5. Packet Loss Ratio

7.5 .1 Packet Loss Ratio (*fixed-area scenario*)

In order to identify the energy efficiency and the network performance, we must analyze the packet loss. Packet Loss is defined as the failure of the messages to reach the final destination. *Fixed-area scenario* is about the analysis of network consist of 12 modules where node degree is a constant value and we have a single source-sink communication.

In our approach, used a variation in finding the packet loss of overall messages. Data packets was used (in 100 bytes length) independently of control and typical messages nodes exchange. Data packets can be forwarded by source-node to destination-node only after finding optimal path.
In more details, in our model Data packets start to forwarded after optimal path establishment, and only Nodes of optimal path can forward those data (Nodes in optimal path).
In this part, we have to recall that Nodes in best routing path is 1, 3, 4, 6 and 8 and 11. The figure 7.15 shows the modules and the number of lost packets they have.
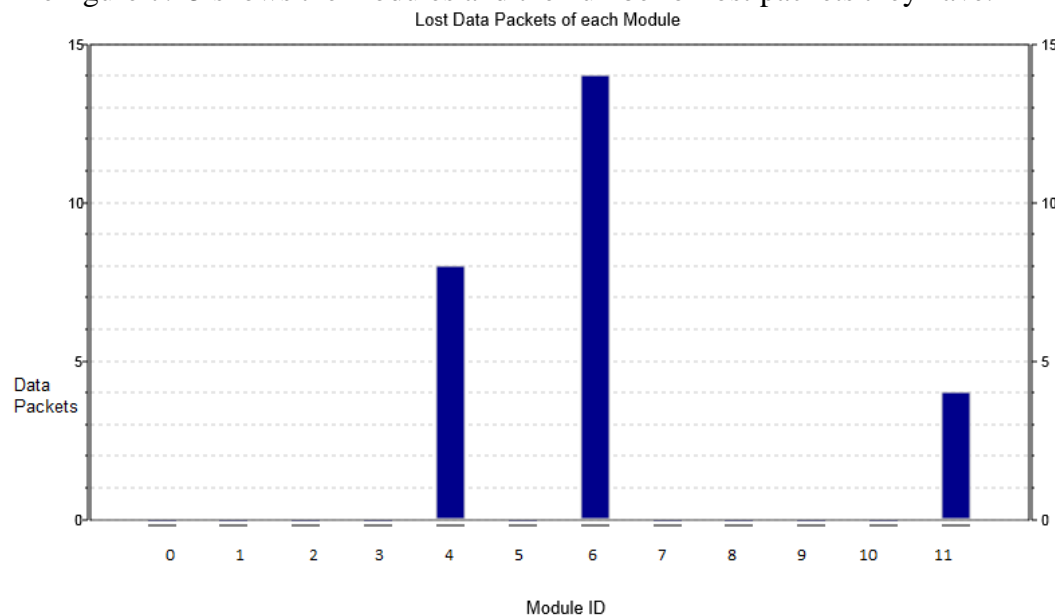
Figure 7.15 Number of Lost Data Packets

In this chart we can highlight that Nodes 4 , 6 and 11 lost 8, 14 and 4 data packets respectively. As in the graphical Tkenv environment is presented, Nodes 4 , 6 and 11 have a 100% per cent of Data loss. Node 4 and Node 6 are in path. This contributes to a big disadvantage of transmission in chosen methodology.

On the other hand, all other nodes have 0 % per cent of packet loss. More significant is the packet loss of Nodes 1, 3 and 8. Although simulation model results in 0% packet loss for Nodes 1,3 and 8, actually Node 3 have zero packet loss ratio. Node 1 and Node 8 transmit no data messages at all because they do not participate into optimal routing . Figure 7.16 shows this fact. This is a graph for showing number of data transmitted.
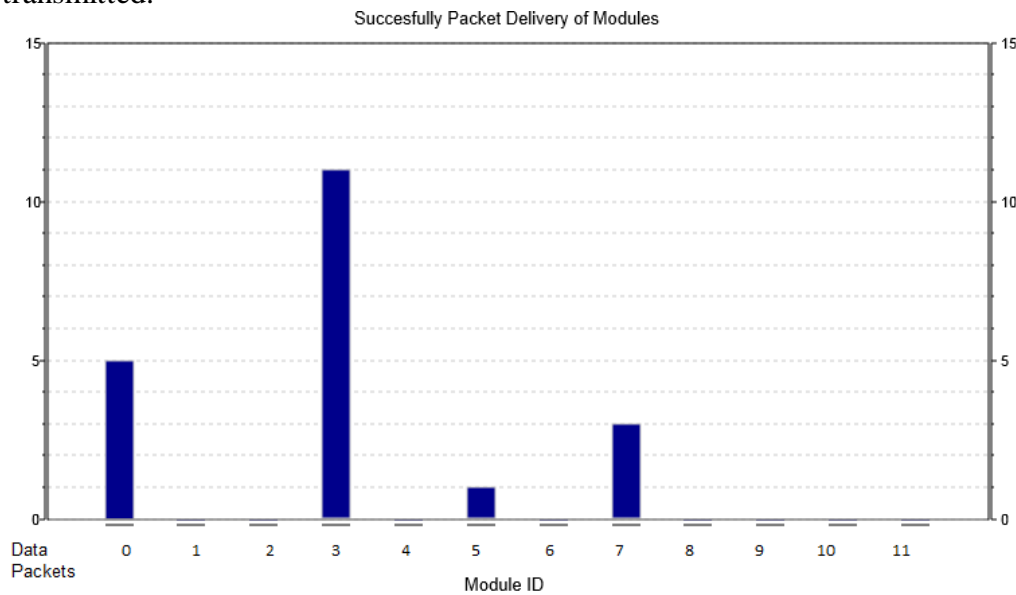
Figure 7.16 Successful Transmission-Data Packet Delivery

7.5.2 Packet Loss Ratio as a Function of Network Size

Packet loss ratio is an important performance metric for networks because indicates in what degree messages can be successful transmitted or else, how much our routing approach can improve communication to final destination. In this part there is an analysis of packet loss ratio metric but here we examine this metric for three different network sizes.

In our models (WSN 12, WSN 32, WSN 64) we have the percentage of successful packets transmitted in Figure 7.17. We measured the average packet loss ratio in order to view packet delivery for overall systems and not for each module.
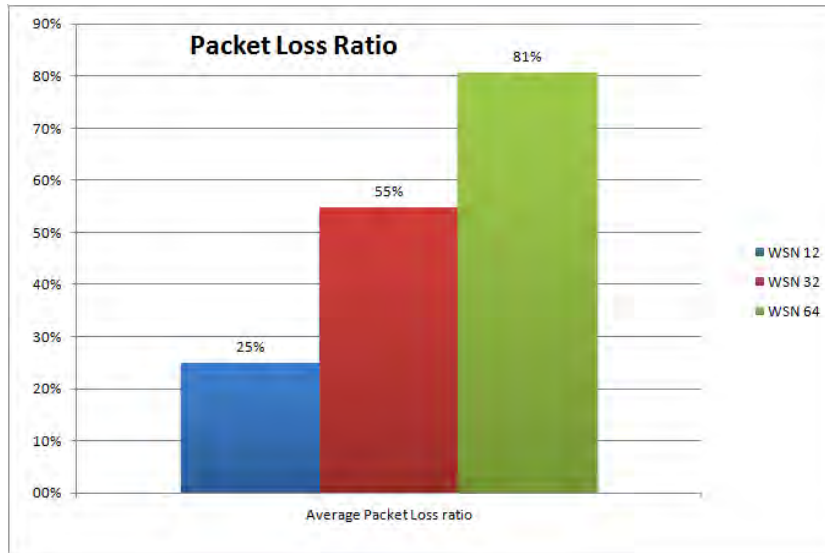
Figure 7.17 Packet Loss Ratio

As Figure 7.17 shows while the size of network increases, the failure in transmissions unfortunately increases too. The system with 12 nodes (WSN 12), seems to have the best delivery ratio as only 25% of packets are dropped. WSN 32 and WSN 64 have worst performance, something that can be explained due to chain topology. Especially, system with 64 nodes almost successfully deliver only 20 percent of packets, measurement that indicates the poor throughput and performance of the system.

7.5.3 Conclusion on Packet Loss Ratio

To explain these results, we must consider that in WSN 64 there are many nodes that are inactive, they do not receive neither send any message at all, so there is a great number of unused nodes that can not exchange data.

Similarly, WSN 32 has almost half of total messages transmitted without any problem and reached destination. This poor result is consequence of the same reason, here some nodes do not operate normally (as inactive) and moreover, nodes near initial sink and source have a performance similar to our initial system, the system with 12 nodes (WSN 12). This results to have approximately 15 nodes exchange data and the other 17 nodes transmit minimal or none of data or other types of messages.

In addition, there are two sources and two sinks in the system (in 32 nodes scenario). The first sink and source communicate successful but the other part of system (the second sink and source) do not communicate with same amount of data exchange. The nodes near first source are exhausted to soon and this makes the system end its operation.

As we can conclude, the size and the topology of network affects the packet loss ratio. As size is increased, more failures occur, increased multi-hop communication causes packet be discarded before reaching the destination and system tends to lose accordingly data messages.

Chapter 7 Result Discussion   120

7.6 Delivery Packet Ratio

To evaluate the performance of our system there is a presentation of delivery packet ratio. This evaluation criteria indicates the level of delivered packets to the destination. In order to measure this value, we consider all types of messages from generator module to receiver module. In Figure 7.18 we can see the results.
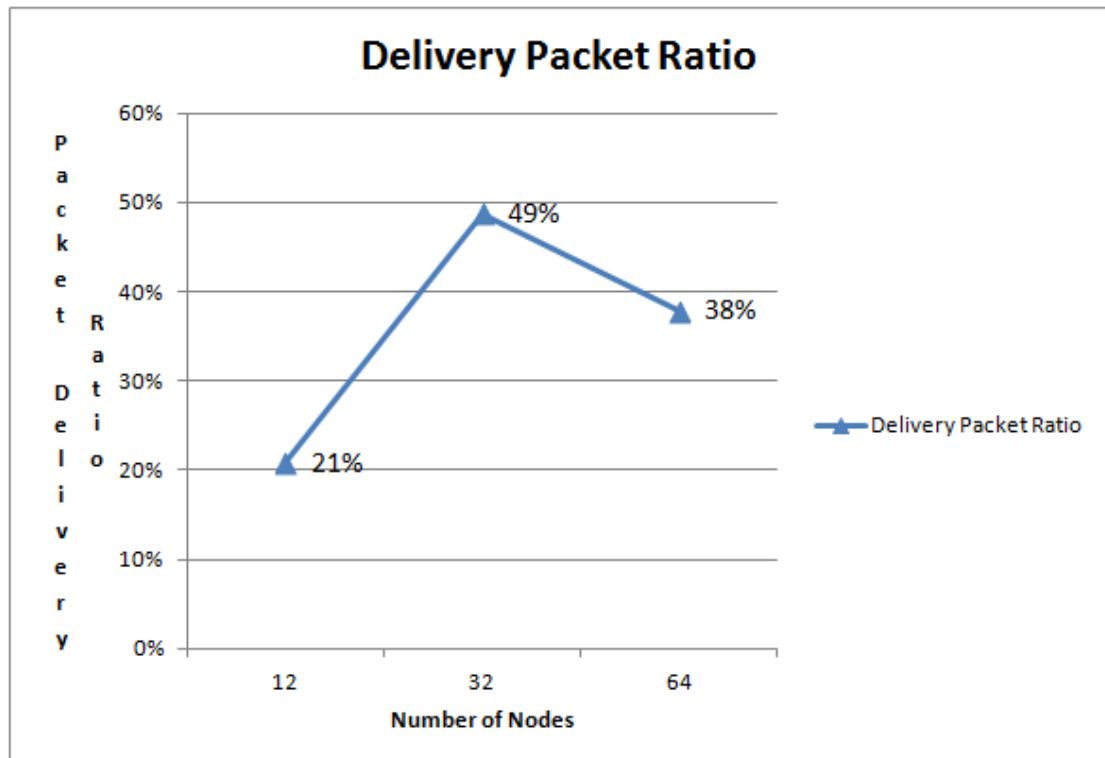
Figure 7.18 Packet Delivery Ratio

The Figure 7.18 shows generally that all network systems do not manage to have a great value of packet delivery ratio. In other words, the highest value is at network consist of 32 nodes, but still the PDR is quite above 50 per cent of total transmissions. This means that half of the messages will not be delivered successfully, indicating the poor performance of network.

The other networks have minimized PDR values , the smallest network has only approximately 20 per cent and biggest network with 64 nodes has almost 40 per cent PDR.

These values reveal that although systems generate a great amount of messages and packets, an important amount of them will not be able to reach destination. The main reasons for this can be firstly, the large energy wastage that leads to many link failures of routes from Sink to Source, secondly in the first network, nodes are deployed in a sparse manner, which shows that nodes are not very close to each other , thirdly there is a great amount of data packets that can not be forwarded considering the high packet loss ratio of some nodes and finally many redundant and duplicated messages will be discarded and will not reach destination especially referring to Interest messages.

Controversially, the networks having increased density and particular having chain topology manage to perform better than initial small system (with 12 nodes). This can

Chapter 7 Result Discussion   121

be viewed above in the following Figure 7.19 where there is a comparison between our DDRA approach and AODV approach. In AODV node density is a crucial factor for having better communication and better QoS. AODV is the Ad Hoc on Demand Distance Vector where is a reactive protocol routing and routes are established when there is a request for data transmission.[48] Although our approach is a hybrid routing protocol (proactive and reactive) there is a similarity on routing tables that both protocols support.
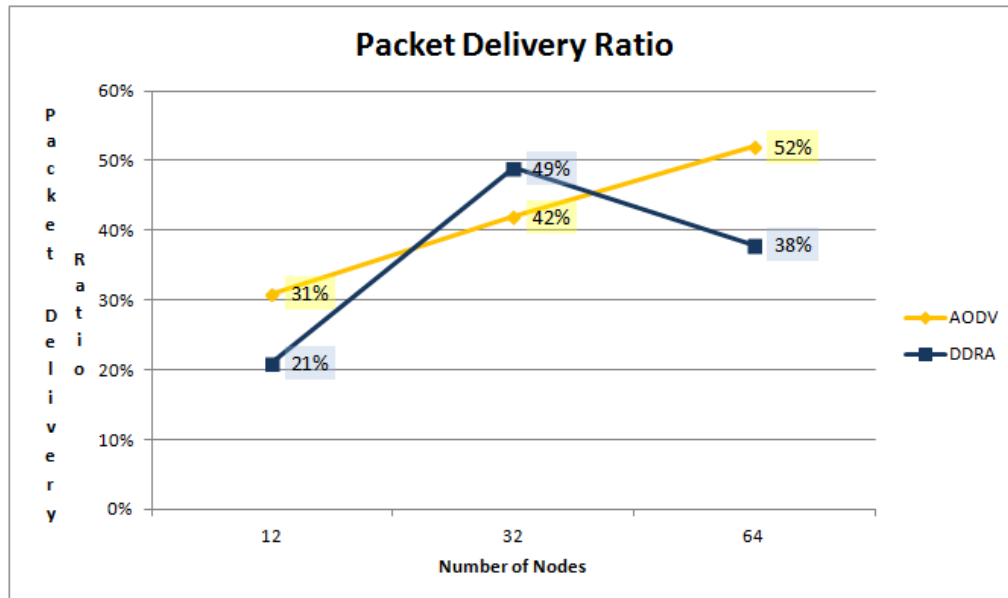
Figure 7.19 Packet Delivery ratio (AODV-DDRA)

One advantage shown in this figure(Figure 7.19) is the performance of medium size network. Our solution seems to overcome and achieve a slight difference of 49 per cent in PDR. This happens perhaps due to chain topology of subgroups and limited distance between nodes.

As node density increases the number of packets generated will be increased too. This leads to network congestion [42]. Another very interesting key feature that authors in [42] suggest is the implosion. Implosion when networks have a large node density is a proof of having many duplicated messages, many copies of the same messages has to be transmitted and consequently the energy wastage is great.

Implosion is also a main disadvantage of flooding protocols, a feature that in our network model exists and occurs due to node degree and Interest flooding.

7.7 Average End To End Delay

7.7.1 Packet Delay for Traffic Flow Patterns

Another evaluation criteria to discuss is average end to end delay. End to end delay refers to the time span between the packet sent from a sensor node and received at destination. In our model we measured this time difference for packets following optimal path at the final phase of our algorithm. To find average end-to-end delay of packets we sum the packets that reach successfully the destination, the next target. The following chart shows the average end to end delay of Systems. In Figure 7.20

there is the average delay of DDRA, PCDD and DD approaches. It is obvious that DD does not manage to keep delay in minimized levels when sources are increased.

On the other hand, PCDD has the minimum values for Packet Delay. The performance of PCDD gains as can be viewed here, in Figure 7.20. The number of sources increases but the level of delay is reduced. This happens because, although more sources flood network with more exploratory data, PC is a successful technique that helps delay not to increase significantly.
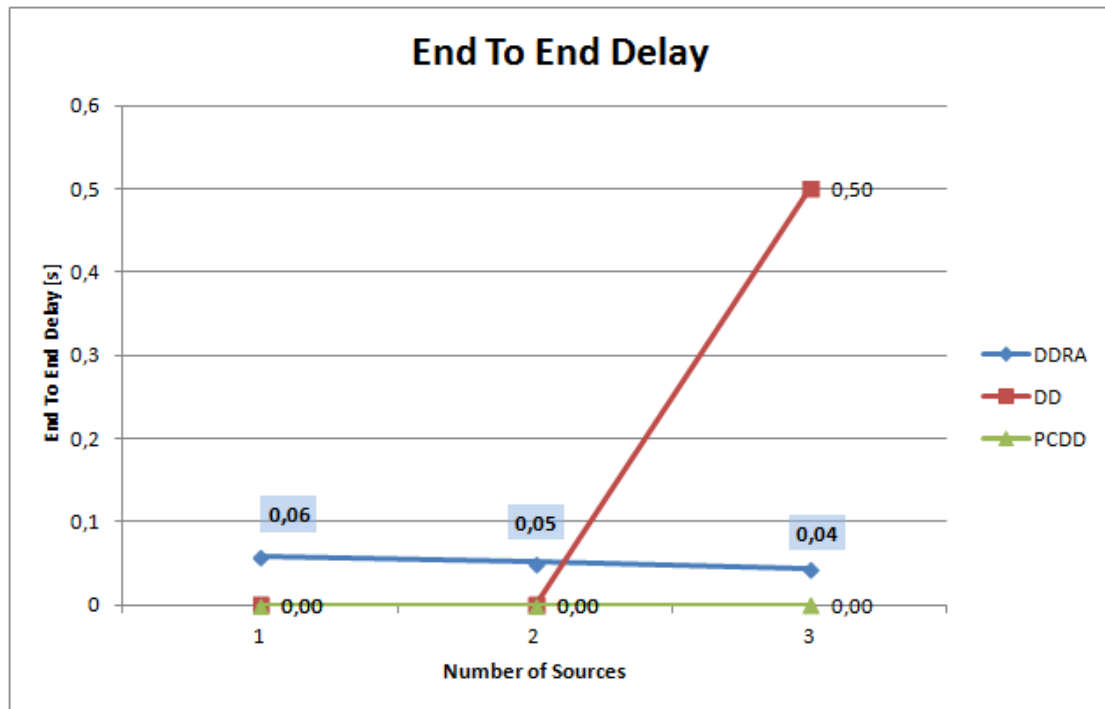
Figure 7.20 Average End To End Delay

7.7.2 Conclusions on End To End Delay

In our system, as the number of sources increase, the node density (consider the three network systems with 12, 32 and 64 nodes) results to minimal delay. As the sources increase, the distance and consequently the hops between source and sink decrease. As topology of system with 32 sensors show, sources are placed intermediate other nodes and the path that packets have to travel is shorter.

Also, as node degree increases, there are more neighbors to communicate, so there are many options to choose a better and shorter path.

To conclude, PCDD remains the best solution on delay of packets but our approach managed to get better results from traditional approach of DD routing algorithm, where all nodes have to relay their data all the way to sink in multi-hop communication, but in DDRA we use clustering to minimize delay.

7.8 Conclusions on Result Discussion

As Figures indicate, the DDRA didn't manage to keep energy consumption under the energy cost of DD and PCDD approaches. This is a result produced by many factors. These can be summarized as following categories:

- Increased generated messages
- Redundant paths and message travelling through them
- No efficient solution about nodes belong on backbone
- Limited usage of small number of nodes
- Operations of high energy cost lead to node exhaustion

Chapter 8

Conclusions

This chapter is about conclusions, evaluations and suggestions of future research.

8.1 Aims of Thesis

The aims of DDRA for WSN can be summarized to the following points:

1. Reduce Data Redundancy, by decreasing number of messages. Interest messages could be reduced and Gradients would be created by minimized energy cost. Interest follow paths according clustering and localization.
2. Avoid "Hot Spots" problem by setting cluster heads based on greater energy resources and based on topology. Clustering scheme selects nodes far enough from source and close enough to sink to reduce energy cost of data transmission.
3. Decrease energy consumption by modifying inter-cluster and intra-cluster communication. For inter-cluster communication CHs forward messages to next neighbor. For intra-cluster communication, cluster members or ordinary nodes accept data and do not retransmit information to other clusters until they are elected as CHs. When CH of a cluster is exhausted, cluster members compete for this role.
4. Percent decrease of Dead Nodes by setting a hybrid method for CH election. Proactive method is used in the initial phase of network and Reactive method is used in the next phases of communication. Source initiates requests and CH election is recalculated depending on energy.
5. Avoid Additional Headers and Filters on messages, to maintain messages having limited size. Messages have fixed length.
6. Avoid duplicated messages exchange. Tables in cache memory of sensors are used to record paths of Interests and to update information about routing paths.
7. Sustain connectivity and maintenance of network. Allow alternate nodes play the role of cluster heads when CHs are out of energy to maintain communication.

8.2 Evaluation

To evaluate our approach we will distinguish the basic features into two categories, the strong points of DDRA and the weak points of DDRA.

8.2.1 Strong Points of DDRA protocol design

DDRA is a routing protocol that has as main goal the reduction of energy consumption. Although, most of results indicate that many attributes and many features make this approach difficult to operate properly and bring on effective results, there are some interesting points that this approach reveal.

These points can be summarized as following:

1. Hierarchical structure of system after cluster formation

2. Elimination of redundant data by allowing not each module create gradients. Matching on same event allows only nodes having same event create gradients. Gradients help to create reverse path from Sink to Source.
3. Re-election of a Cluster Head when energy resources are exhausted. This manages to maintain network operation without failures and collisions.
4. Query-based approach on finding the requested event. Sensors having other events, do not participate in sensing and data processing. This helps to save energy of duplicated data and redundant information exchange by sensors that can not sense the requested Event.
5. No complicated method is used like Filtering of messages which adds more network overload of additional headers. Messages are predefined in the initial state and include the expected information. This is an effort to avoid complexity of many updates and new type of messages transmitted.

8.2.2 *Weak Points of DDRA protocol design-Evaluation on Design Issues*

Although the proposed system tries to combine two methods, Clustering and Data-Diffusion the results showed such technique is not an easy and effective idea. The following points contradict to the main objectives of approach.

1. Energy Holes created near static sink and source. These nodes got energy exhausted very quickly in comparison with farther located nodes.
2. Enormous traffic caused network overload to nodes. Nodes played the role of Cluster Head was obliged to forward greater amount of data. This leads to unequal energy consumption. Energy Consumption differs in large scale between sensors of network and there is an unbalanced energy consumption in our system. This is crucial disadvantage considering the main goal of this approach as an energy-efficient method.
3. Redundant messages still travel in communication phase, generated by diffusion method in which is necessary the flooding of Interest. Moreover, exploratory data exchange creates additional overload.
4. Main backbone of CHs is the centralized element of network. Failures and congestion on this route affects directly network performance. Single-point failure in some case will cause end of process. No recovery mechanism.
5. Network lifetime is not successfully prolonged due to large energy wastage. Network operates similar to traditional approaches (e.g. DD) and PC does not achieve to increase lifetime of network components.

8.2.3 So what are the new features in this approach?

Regarding the previous discussion we can summarize and highlight new features that our simulation model has compared with other solutions. These features are:

- No headers are included and no filters are imposed. This avoids messages to keep initial length and not to increase. As size of message increases, transmission will increase in duration time and required energy.
- There are no multiple states of nodes, they are simply four states, one of them indicated as Dead. This is more simple and adequate feature to prevent multiple changes of status of network elements.
- Election of Cluster Heads and Forming clusters is a process that can be placed either in the beginning or in the mean time of simulation. Acts in a proactive

and reactive manner, more simply hybrid method. The factors of node degree, location and energy resources contribute to calculate probability ratio for each node any time it is necessary. This is not much complex to calculate this value.

- Energy consumption is decreased in comparison with system having no clusters, although this is not a meaningful reference point based on other approaches. Energy consumption tends to be less in case of large network but specific topology, like groups around source and sink. Chain topology is inversely related to energy efficiency on DDRA design.

8.2.4 Research Objectives-Questions and Answers

In this part of Dissertation we will examine and discuss two basic characteristics. The first one is about the research objectives of DDRA for WSN and the reasons of why we got these results.

Objective 1: Reduce Data Redundancy. Although in our system it is a necessity to flood network with Interests, otherwise we couldn't have gradients to point to node which sends Interest, data redundancy still exist in a large scale. In our design gradients are not kept in cache memory, but they are exchanged. This allow to sensors, devices with limited memory storage, use cache memory for Events. Events are stored on Tables in sensors. Gradients are the message that carry the event id, the request we search on the network.

By this way, we limit the Interest message size, conserve more energy level in the beginning. On the other hand, gradients keep travel and updated and this communication tend to waste important remaining energy on devices.

Objective 2: Avoidance of "Hot Spots" problem. Formation of clusters leads to balance energy in nodes. On the other hand, using specific CH election scheme resulted into creation of a backbone of network, transmitting major amount of data and having a large extra overhead.

Objective 3: Decrease energy consumption. Inter-cluster and intra-cluster communication ends up to nodes exhaust of energy very quickly, in contrast other nodes that either set as cluster members or they do not belong in the backbone and maintain large residual energy at the final phase of simulation. This large difference reveals that this approach could not balance energy consumption and again "hot spots" problem exist too.

Objective 4: Avoidance of additional headers and filtering of routing core. The approach used has a more simple solution about messages. Message types are five but each of them has a constant length. This means that the energy cost of transmitting each one of it it is stable, does not affect the node differently at any phase of algorithm.

Objective 5: Allow alternate nodes become CHs. This is a method in DDRA that allow any other node take the role of CH. Possibilities to become cluster head are recalculated during network performance and they are updated. Obviously, CH election depends on energy level reductions so as energy resources change, nodes become "weaker" than others.

8.2.5 Discussion about Results

The results of our research showed that this technique has several disadvantages. The main reasons of coming up to these results can be summarized to next points:

- Small size of network

- No correct management of redundant messages
- Multiple factors of CH election
- No distributed and balanced solution of energy consumption
- Small lifetime of Network
- No scalable approach, requirement of knowledge on network topology

These facts lead to having unexpected results and conclude that DDRA can not be characterized as an energy-efficient solution compared with recent related works.

8.3 Literature Review

As many times was mentioned in this research the main motivation was the "*Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering*" research of Vlado Handziski et al. This analysis suggests that clustering can improve energy efficiency on a wireless sensor network. DD was used as a data-centric routing and design of application layer is based on this technique. Authors in [2] have set the following question: "*Is it feasible to execute directed diffusion on top of a sensor network where the topology is implicitly constructed by passive clustering?*" [2]. They developed a proposed method and a model that achieved previous assumption. As they claim, "*depending on the scenario, PC can significantly reduce the required energy, even improve delay and delivery rate*".[2]

Many attempts have been made [*Maizate 2013, Akbas 2013*] with the purpose to improve clustering protocol and dynamic interests routing, respectively. Those approaches managed to improve energy conservation in WSNs.

In more details, the election of CHs in our approach was motivated by "*(EBHCR) Energy Balancing and Hierarchical Clustering Based Routing algorithm for Wireless Sensor Networks*" [36], 2011 where authors give as competence value of each node an equation, a formula depending the energy resources, the number of neighbors and the distance from Base Station (in our case Sink).

The major difference in our approach was the probability each node had to become CH.

Similarly energy, distance and node degree still are used but here we set the maximum hops we want in a cluster . Another feature is that subtracting the distance of candidate node to Sink from the maximum hops from Source to Sink we set cluster heads be distributed and farther from Sink in order to prevent nodes near sink be elected as CHs. This helps to avoid creation of bottlenecks near sink. Sink is a point of system where massive amount of data reach and are accepted.

8.4 Proposed System

In this part there is a summary on the DDRA for WSN. We will present DDRA in 6 steps:

1. Source starts to flood network with Interest messages. This is a query-based approach. Interests are actually requests. They are descriptions of a specific event in the geographical area. Each neighbor propagate the Interest to next neighbors. Eventually, Sink accept this Interests.
2. Event Tables are in cache memory of sensors. This helps to record the Interests they accepted and if they have an Event similar to the Interest has been received they create gradients. Gradients will be created after clustering. In this phase they only show Interest propagation from Sink to Source.

3. Clustering begins in a hybrid manner. In this phase clusters are formed on energy resources and localization. Later clusters will be re-formed as energy will be decreased. The aim is to choose nodes with large energy resources ,located farther from sink and source and connected with sufficient number of neighbors (node degree). After communication alternate advanced sensors play the role of CHs.

4. After clustering, each sensor updates its gradients based on clustering and Event Table. Now gradients show more energy-efficient paths from Sink to Source as clustering technique contribute to this. Gradients are important to make message exchanging more energy-efficient in next procedures.

5. Communication starts. If there is a match of a gradient and Event id in cache memory, exploratory data start transmitted from Source to Sink. This is the reverse path gradients have set up. To mention again, gradients show the reverse path of Interest propagation from Source to Sink, the path from Sink to Source. The aim here is to make sensors exchange exploratory data and reach Sink. Only nodes having gradients to next node and match in event can make exploratory data.

6. When Sink accepts exploratory data, can start path reinforcement. Optimal path is created from Sink to Source and show the energy-efficient routing for data. The route created by exploratory data indicates the lower-energy cost transmission.

8.5 Recommendations for Future Research

Future Work

This section contains new ideas and new reviews on the DDRA approach. As many disadvantages have been brought up by measurements of energy cost and number of messages, there are many and very interesting alternative methods we could use to improve this network model. There are two main parts in this dissertation about working on this idea in the future. Firstly, to present more ideas about elimination of disadvantages and "weak points" and the second is about implementations of the approach, in other words how this routing algorithm can be used to improve wireless sensor networks.

8.5.1 Part 1: Improvements on DDRA for WSN Design

In this section there is a discussion on the performance metrics and new attributes that our design could farther include. Each topic is categorized to the following:

a. **Channel Attributes**

In the simulation model, although initially was purposed on wireless communication, nodes are connected by a channel of 100 msec delay. One of the most interesting issue is to change this characteristic. Wireless channel must consider the path loss jitter, the size of a cell (area is divided by transmission cells with certain size) and generally the option a node can reach or not a neighbor. So, to conclude, wireless communication must consider the transmission range of each node, the size of cell, the strength of signal transmitted and the MAC and Physical layer protocol we could use for communication (for example IEEE 802.15.4 and ZigBee standard for network and application layer). In Figure 8.1 [48] we can see the protocol stack using IEEE 802.15.4 standard. This standard is for Physical and MAC layer and for Network, Application Support Sublayer and Application layer there are protocols like ZigBee, 6LowPAN.
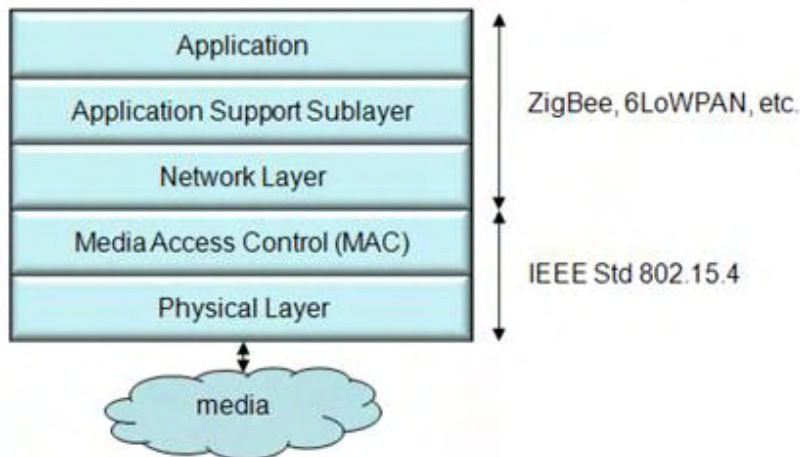
Figure 8. 1 Protocol Stack for WSN of IEEE 802.15.4

b. Sensor Mobility

Another critical issue about wireless sensor networks is the movement of sensors. As typically, when we talk about sensors, we expect nodes are not static and their movement makes routing more difficult. Apparently, when sensors move, they may leave a cluster, they may belong in a transmission range of two clusters (depending the network density) and may join a new cluster. These options are not included in our approach. It is mandatory to consider these changes in node location and DDRA must proceed to include such issues.

c. Fault Tolerance

As authors highlight in [50] one of the most important factors that influence sensor network design is fault tolerance. Questions about what happens when a sensor fails or is blocked due to lack of energy resources or has a physical damage must be taken account.

In DDRA nodes that are weak , are automatically participate again in clustering competition and re-assigned with a new probability value to become CH. This technique helps us to avoid failure of network system when some nodes have lost a significant amount of energy.

In fact, when nodes around and directly connected with source are out of energy and do not manage to sustain network functionality, like routing between source and sink, we set this condition as the stop of simulation. Network stops the operation when nodes can not any more communicate to have a route from source to destination, in our case from source to sink.

To sum up, fault tolerance in our approach is strongly depended on sensor's functionality, sensor's ability not to be recharged when they are out of energy, but mostly when as a set can not anymore operate and exchange data.

d. Scalability

The number of sensor nodes deployed in a area can vary from few to thousands. This depends on the application that system performs. For environmental applications like agriculture of natural disaster detection, the node density may include great amount of sensors (for example 20 nodes/m^3 authors indicate in [50]). Another example is that for Sensors for Habitat Monitoring Application can reach even 100 per region [50].

Our approach can be scalable and include many sensor nodes, many sources and many sinks. The information that nodes exchange are local information. This feature

Chapter 8 Conclusions 131

helps to create networks without a priori information of node location. The system can measure the distance of each sensor from source, can exchange data for residual energy and graph Diameter is used to determine the election of a Cluster Head.

Although clustering in our design requires topology information. Alternative methodology choice could result in node discovery by hop-by-hop tracking.

Nevertheless, we can make the number of maximum hops in a cluster be not a constant value, but be a parameter calculated by network size. In other words, if there are 15 sensors we could have 3 maximum hops in a cluster, if there are 25 sensors we could have 4 maximum hops and so on.

e. Production Costs

We could have a non expensive solution for our network deployment. In Apple Store we can see that an "Nike+iPod Sensor" costs $ 19.00. This sensor can be applied in a pair of Nike shoes , beneath the insole, and works with iPod Nano Device or iPhone to track the workouts of a human. In combination with iPod- receiver, this sensor can count and inform us about training [52].

As we can highlight the cost of a sensor is not very expensive. Although in the case where thousands of sensors must be deployed, cheap solutions are provided by several manufacturers.

In our case, we have to consider the cost of sensors, the cost of receiver and transmitter and the cost of Bluetooth radio system (e.g.) which can be bought at less than $ 10.00.

f. Sensor Network Topology

One of the major challenges in routing approaches is the issue of topology maintenance. As in our approach, channel is not wireless and nodes do not move this subject is rather solved.

In contrast, in real communication, sensors are devices mainly conducted for wireless communication and must provide avoidance of failures. As described before, factors like node movement and node exhaustion are very important.

In our approach, nodes do not have mobility attribute, so topology is static. Network topology affects strongly methods to choose clusters and create communication paths.

g. Network Size

One interesting option is to evaluate the system with more sources/sinks and more nodes. Previously results show that energy tends to reduce when the network size is larger , three sources and two sinks. Further discussion about 100, 150 and 200 nodes would reveal the network performance and help discover different traffic patterns and findings.

h. Transmission Media

As our network is a multi-hop network, a perspective for future work is to consider the wireless medium for communication. There are three options described in [50] as wireless medium: radio, infrared and optical media. Especially for radio communication there is the use of industrial, scientific and medical (ISM) bands. For Europe 433 MHz ISM band is available and for North America there is 915 MHz frequency range in ISM [50].

Infrared is another solution in wireless media communication but unfortunately there is a basic constraint of line of sight. To conclude, optical media are very interesting too. A brilliant idea is to use "Smart Dust motes". As [53] shows, this system consists

of tiny sensor nodes "motes" of mm^3. Motes can communicate with active transmitters using laser diode and for optical communication, there is Corner Cube Reflector (CCR). By optical communication the basic goal here is to construct a system, with so tiny devices to " Track Every Thing ,Every where", as [54] presents.

In our model, we could use radio transmission and measure network metrics like QoS and fault tolerance.

i. Power Consumption

Although the power consumption of sensors are divided into three main tasks, sensing, communication and data processing, there are some interesting points to mention. First of all, as our routing approach enhance data-aggregation a adequate part of power consumption will be wasted in communication and data processing. The fact that we consider nodes have information in their cache memory, leads to the conclusion that we would need more capacity on memory unit.

Besides, the sensing is a process done after gradients set up and after there is Event matching. This is a major difference while we try to limit power consumption of sensing. Not all nodes will sense and just transmit information to neighbors but with gradients and other messages we manage to keep only specified sensors (indicated by a query-like approach) transmit an event. This is also managed through clustering by which we try to keep other cluster members in energy-save mode.

8.5.2 Part 2: Applications related to DDRA

a. Usage and conclusions on DDRA routing approach

In this part we will discuss possible implementations and applications on DDRA.

b. Ethical Issues on DDRA

Here there is a discussion about implementation of DDRA. Although routing protocols of WSN are based on limited energy resources and limited processing capabilities of sensors, some implementations and future researches may raise social and political issues about privacy and public life.

In the previous part of Future Work there is a reference of "Smart Dust Motes" [55]. This innovative idea has been proposed in 2004. As Yao-Jung Wen states in his research titled "Smart Dust Sensor Mote Characterization, Validation, Fusion and Actuation" , *"Smart Dust is proposed as a futuristic dust-sized sensing and communication unit based on MEMS technology"*. Motes having size of millimeter are configured within a dense distributed sensor network. These minimal devices have consist of microcontroller, a communication unit and they can integrate sensor modules. Operating system must be compatable with such limited memory unit, for example TinyOS. The main goal of motes is to use the sensors to sense physical phenomena and provide local and global knowledge without high rates of failures.

In [56] authors also claim that the main reason of "Smart Dust Motes" is to create minimal platforms to sense and communicate for environmental monitoring and military or commercial applications. The main attributes of "Smart Dust Motes" in [56] can be summarized to the following points:
- MEMS structure
- Passive Corner Cube Reflector (for Base Station communication)
- Inter-mote communication

- Thick-film battery charged by solar cells

To summarize the general idea is the combination of thousand motes attached with sensors to communicate and provide information.

Although the task of monitoring environmental parameters like humidity, temperature and pressure is necessary for applications like Climate Change Monitoring, Natural Disaster Detection and Agriculture, the operation of a "Smart Dust " system may includes some controversial aspects.

In the case DDRA protocol would be applied in such a system or any other energy-efficient approach , because one of the main constraint in this system is the limited storage of energy, possible solutions may not have an "innocent" character.

In [57], Mark Anderson claims that "smart dust motes" are like tiny computers that can be embedded in things and buildings. The article with title "The size of a tiny computer is no bigger than a snowflake " of "New Scientist Tech" Magazine reveals how tiny computers can be used.

Team of University of Michigan reports that motes can be embedded even in our belongings. One example is in case of someone has lost his keys. He can ask Google, "where are my keys" and if keys have an appropriate mote then Google App could give the right answer.

In contrast, consider the case motes monitor every change and movement of things or even humans. An application of finding the presence or absence of a person in an area could be very stressful.

While our world around us can be sensed, detected and reported (in worst case to unauthorized users) is this socially approved?

Imagine an application scenario where there movement of humans in a town e.g. is tracked. Obviously, monitoring environments and buildings may prevent society of illegal actions or crimes but social issues arise like security, privacy and legal issues. The perspective to be tracked even with or without our consent rises questioning and dilemma on our privacy . To add, considering the Wireless Body Area Networks (WBAN), where wearable devices will report and record our body's actions and movements we must think if human tracking is a fair enough prize to pay for our security.

8.6 Applications related with DDRA

At this point, WSN applications are presented. DDRA for WSN could be used and combined with other protocols (transport protocol, MAC protocol).

As an approach based on data-aggregation and clustering scheme, DDRA could be an routing approach for many applications about WSN. In this section, there is a presentation of the most recent applications about WSN. Applications about WSN will be summarize into three categories:

I. Smart Environmental Monitoring

Authors in [42] present smart environmental applications. As heavy industry has developed globally, this leads to increase pollution of ecosystem. Moreover, Health Care applications too, make the need of monitoring, controlling and treating pollution nowadays, important. The key is to develop reliable, cost-effective and real-time methodologies integrating WSN technology. By this way, control prevention and notification

of harmful changes in environmental status like humans, animals, or even as in [42] mentioned, changes "on the shape of Earth" , can provide a self-monitoring and self-protecting system. Also, it is important to develop systems to operate reactive and proactive in case of many phenomena that occur. WSN systems must collect data, process, analyze and disseminate data. A great challenge is to build WSN systems that can act reliable and fast on any unwanted changes. That's why it is so important to develop routing algorithms with high energy-efficiency and minimized delay on dissemination of data.

Human health is directly affected by environmental pollution and many problems appear in undeveloped countries [42] where renewable sources of energy are not widely used. Here there is a categorization of the most important issues on environmental pollution :

- Air pollution (toxic gases, sulfur dioxide, nitrogen oxides etc)
- Natural Disasters (quakes, floods, tsunamis, etc)
- Water pollution (chemical discharge, petroleum leaks etc)
- Global warming and Ocean acidification
- Agricultural pollution (soil contamination, Pesticide leaching)

II. Ambient Intelligence (AmI)

Authors in [58] include in their research the concept of Ambient Intelligence. Ambient Intelligence refers to " electronic environments, which are sensitive and responsive to the presence of people, where devices …support people in carrying out their daily life activities and tasks easily". In AmI the main idea and challenge is based on nano-technology to let only user interface interact with people , interconnection of sensors and devices in WSN and intelligence of software for decision-making [58]. The science of AmI involves many scientific fields as robotics, pervasive computing, artificial intelligence and human-computer interfaces (HCIs). "Smart Devices" now are introduced to humans and the science of AI contributes to perform activities that require intelligence, knowledge and perception. Pervasive computing is also based on the usage of microprocessors for communication and information exchange in objects we all use every day.

Some of the applications in every day life as Michael Friedewald and Olivier Da Costa present in their research book titled "Science and Technology Road mapping: Ambient Intelligence in Everyday Life" are:

- Home Automation (security, physical access control, control of electronic appliances)
- Health-Care and Well-Being (logging health data to the database, prevention of harmful events associated with elderly, disabled and children)
- Communication (socialization, intra-person communication in house)
- Entertainment (information about environment of a person, commercial information for tourists and business travelers)
- Avatar (Personal Information Assistants)

The reason that our DDRA approach is connected with AmI is the need of collecting data in a wireless network and the process of sending data to a centralized node , a sink for data analysis and decision making. In our approach the basic idea is the optimal communication between nodes and sink and the effort of an energy-efficient solution.

III. Sensor-Cloud Applications

Although WSN are used in several areas like healthcare, defense as military target tracking there are constraints for example short communication range and resources. On the other hand, Cloud computing is a future technology. US NIST (National Institute of Standards and Technology) defines that Cloud Computing is a "on demand network access to a shared pool of computing resources…requiring minimal management sensor or service provider interaction". In [59] authors present a survey of sensor-cloud architecture and applications.

According to IntelliSys, Sensor-Cloud is an infrastructure consist of three main parts[59]:

- sensors for pervasive computing and gathering data in physical world
- clusters for computing data and playing the role of backbone of this network
- internet as the communication medium

As this definition shows, clustering is a basic element in the sensor-cloud architecture so routing algorithms based on clustering are obviously quite important.

In this part although, we will present basic applications and no further discussion of provisions and challenges of this scientific area. But the main advantage of Sensor-Cloud computing is the ability of Increased Data Storage and Processing Power, feature which is a limitation of sensors.

Some of existing Sensor-Cloud applications that authors in [59] discuss are:

- Nimbits (system that data on cloud can be connected with Scalable Vector Graphics (SVG) and notify with alerts social networks)
- IDigi Platform (manages communication between remote device assets and enterprise applications).
- ThingSpeak (open source IoT application for location tracking, sensor logging applications and social network of devices which can update their status)
- Google Health (service of Google for personal health information and including cloud health data storages *)
- Microsoft Health Vault (this platform helps users store health and fitness information, pharmacies, health employees and users can obtain such data)

* Google announced that this service is no more available

8.7 Conclusions

The DDRA approach is a routing protocol based on clustering and data dissemination in fashion of DD. In this dissertation an energy-efficient approach is presented. The main motivation of this research is the WSN technology and their growing trend into various fields like industry, commercial and healthcare applications.

Our design of DDRA is inspired by research work of Telecommunication Networks Group of Berlin Institute of Technology. Vlado Handziski and his co-authors achieved to combine data dissemination and PC. Their approach improved evaluation criteria of their network model like delay, delivery ratio and most significant factor, energy efficiency.

Many researches came after with alternative methods. Rumour routing by creating agents travel in the network, alternative selection of clusterheads, other data dissemination schemes like energy based scheme where nodes having low residual energy are excluded for data communication and many others were developed.

Although the main factors that DDRA approach for WSN are three : energy level of node, distance from destination (sink) and node degree, there are limitations which lead in more research, more effort and more work on network's performance. Many disadvantages like energy holes and unbalanced usage of sensors show the necessity of farther investigation.

Finally, the DDRA approach was a solution with main goal to prolong network lifetime and operate in low energy cost. As mention above, routing protocols on WSN are crucial when the future world trend is data gathering, monitoring and decision-making services.

Bibliography

Chapter 1

[1]   Yu-Chee Tseng, Chih-Shun Hsu, Ten-Yueng Hsieh,1999, *"Power-Saving Protocols for IEEE 802.11-Based On Multi-Hop Ad Hoc Networks"*, IEEE, 2002

[2]   V. Handziski, A. Kopke, H. Karl, C. Frank, and W. Drytkiewicz, "*Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering*" ,EWSN ,Springer, 2007

[3]   Md. Mamun-or-Rashid, Muhammad Mahbub Alam and Choong Seon Hong" *Energy Conserving Passive Clustering for Efficient Routing in Wireless Sensor Network"* ,Department of Computer Engineering, Kyung Hee University, Korea Research Foundation, Grant, 2006

[4]   Zuhal Can , Murat Demirbas, *"A survey on in-network querying and tracking services for wireless sensor networks"*, Department of Computer Science and Engineering, University at Buffalo, SUNY, Buffalo, , United States, August 2012

[5]   F. Ye, G. Zhong, S. Lu, L. Zhang*, "Gradient broadcast: a robust data delivery protocol for large scale sensor networks"* , Wireless Networks 11 ,2005.

Chapter 2

[6]   Basilis Mamalis, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou*," Clustering in Wireless Sensor Networks"* , Zhang/RFID and Sensor Networks, 2009

[7]   Ameer Ahmed Abbasi ,Mohamed Younis , *"A survey on clustering algorithms for wireless sensor networks"* , Elsevier B.V., November 2003

[8]   D.J. Baker, A. Ephremides, *"The architectural organization of a mobile radio network via a distributed algorithm"* , IEEE, Transactions on Communications, 1981

[9]   K. Xu, X. Hong, M. Gerla*," Landmark routing in ad hoc networks with mobile backbones"*, Journal of Parallel and Distributed Computing,2003

[10] K. Xu, M. Gerla," A heterogeneous routing protocol based on a new stable clustering scheme", Proceeding of IEEE Military, Communications Conference (MILCOM 2002), Anaheim, CA,October 2002.

[11] R. Nagpal, D. Coore," *An algorithm for group formation in an amorphous computer*", Proceedings of the 10th International Conference on Parallel and Distributed Systems (PDCS'98), Las Vegas, NV, October 1998.

[12] S. Banerjee, S. Khuller, "*A clustering scheme for hierarchical control in multi-hop wireless networks*" Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM' 01), Anchorage, AK, April 2001.

[13] H. Zhang, A. Arora, "GS3: scalable self-configuration and self-healing in wireless networks'',Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002), Monterey, CA, July 2002.

[14] S. Bandyopadhyay, E. Coyle, "*An energy efficient hierarchical clustering algorithm for wireless sensor networks*" , Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, California, April 2003.

[15] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, "*Application specific protocol architecture for wireless microsensor networks*", IEEE Transactions on Wireless Networking (2002).

[16] R. Sivakumar, P. Sinha, V. Bharghavan,*" CEDAR: a core-extraction distributed ad hoc routing algorithm"*, IEEE Journal on Selected Areas in Communications ,1999.

[17] O. Younis, S. Fahmy,*" HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks",* IEEE Transactions on Mobile Computing ,2004.

[18] H. Huang, J. Wu,*" A probabilistic clustering algorithm in wireless sensor networks"*, Proceeding of IEEE 62nd Semiannual Vehicular Technology Conference (VTC), Dallas, September 2005.

[19] P. Ding, J. Holliday, A. Celik, "*Distributed energy efficient hierarchical clustering for wireless sensor networks",* Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems,Marina Del Rey, CA, June 2005.

[20] S. Lindsey, C.S. Raghavendra*, "PEGASIS: power efficient gathering in sensor information systems",* Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.

[21] Kemal Akkaya , Mohamed Younis,*"A survey on routing protocols for wireless sensor networks",*Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, USA,2003.

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

[23] A. Manjeshwar, D.P. Agrawal, *"TEEN: a protocol for enhanced efficiency in wireless sensor networks"*, Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[24] Arati Manjeshwar and Dharma P. Agrawaly *"APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks"*, Center for Distributed and Mobile Computing, Proceedings of the International Parallel and Distributed Processing Symposium , IEEE, 2002

[25] M. Younis, M. Youssef, K. Arisha, *"Energy-aware routing in cluster-based sensor networks"*, Proceedings of the 10th IEEE/ACM International Symposium on Modeling,Analysis and Simulation of Computer and Telecommunication Systems ,Fort Worth, TX, October 2002.

[26] Rajesh Krishnan ,David Starobinski , *"Efficient clustering algorithms for self-organizing wireless sensor networks "*,Department, Boston University, USA, Elsevier B.V. 2004

[27] Azzedine Boukerche, Begumhan Turgut , Nevin Aydin, Mohammad Z. Ahmad, Ladislau Bölöni, Damla Turgutd*," Routing protocols in ad hoc networks: A survey"*, The International Journal of Computer and Telecommunications Networking, September, 2011

[28] V. Rodoplu and T. H. Meng, *"Minimum Energy Mobile Wireless Networks"*, IEEE Journal Selected Areas in Communications, vol. 17, no. 8, Aug. 1999.

[29] Jamal N. Al-Karaki Ahmed E. Kamal*," Routing Techniques in Wireless Sensor Networks: A Survey "*, Dept. of Electrical and Computer Engineering Iowa State University, Ames, Iowa , Wireless Communications, IEEE, December 2004.

[30] Q. Li and J. Aslam and D. Rus,*"Hierarchical Power-aware Routing in Sensor Networks"*, Proceedings of the DIMACS, Workshop on Pervasive Networking, May, 2001.

Chapter 3

[31] Umesh Rawat,*"Directed Diffusion: Features, Current Developments, Issues and Analysis"* ,Computer Science and Engineering,The Technological

Institute of Textile and Sciences, Bhiwani, Mukesh Sharma2 Phd,2 Associate Professor, Department of Computer Science and Engineering , International Journal of Computer Applications (Volume 49), July 2012.

[32] Miresmaeil Mirnabibaboli*,"Improving the Directed Diffusion in Order to Reduce the Average of Energy Consumption in Wireless Sensor Networks"*, Yerevan Telecommunication Research Institute, IARIA, 2011.

[33] Abderrahim MAIZATE , Chouaib Doukkali, Najib EL KAMOUN, "*Passive Clustering for Efficient Energy Conservation in Wireless Sensor Network*", STIC Laboratory University Morocco, International Journal of Advanced Computer Science and Applications,Vol. 4, 2013.

[34] Arash Nasiri Eghbali, Hadi Sanjani, Mehdi Dehghan ,*"ODCP: An On-Demand clustering protocol for Directed Diffusion",* Computer Engineering Department, Armirkabir University of Technology, Tehran, Iran, ADHOC-NOW, Springer, 2007

[35] Springer Publications web site
http://link.springer.com/chapter/10.1007%2F978-3-540-24606-0_12

[36] Roghaiyeh Gachpaz Hamed, Jaber Karimpour *,"(EBHCR) Energy Balancing and Hierarchical Clustering Based Routing algorithm for Wireless Sensor Networks."* Computer Engineering Department Islamic Azad University, Qazvin, Iran, Australian Journal of Basic and Applied Sciences, 2011.

[37] Sungju Lee, Jangsoo Lee , Hongjoong Sin, Seunghwan Yoo, Sanghyuck Lee, Jaesik Lee, Yongjun Lee and Sungchun Kim, *"An Energy-Efficient Distributed Unequal Clustering Protocol for Wireless Sensor Networks",* World Academy of Science, Engineering and Technology, 2008.

[38] El Ghanami, D. Kwon, T.J. ; Hafid A., *" GRIDS: Geographically Repulsive Insomnious Distributed Sensors – An Efficient Node Selection Mechanism Using Passive Clustering"*, Networking and Communications, IEEE International Conference on Wireless and Mobile Computing, 2008.

[39] Heinzelman, W.B., A.P. Chandrakasan, and H. Balakrishnan, "*Energy-Efficient Communication Protocol for Wireless Microsensor Networks"* ,Proceedings of the Hawaii International Conference on System Sciences, 2000.

Chapter 4

[40] Jang, Hung-Chin Lee, Hon-Chung Huang, Jun-Xiang, "Optimal Energy Consumption for Wireless Sensor Networks", Department of Computer Science, National Cheng Chi University, Taiwan, 2004.

[41] S.Cui and A.J. Goldsmith, *"Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks"*, IEEE Journal on Selected Areas in Communications, vol. 22, August 2004.

[42] Ibrahiem M.M. El Emary, S.Ramakrishnan,*"Wireless Sensor Networks From Theory to Applications"*, CRC Press, Taylor & Francis Group, 2014.

[43] Article on omnet official site:
http://www.omnetpp.org/component/content/article/9-software/3679-castalia-30-released

[44] Athanassios Boulis *"Castalia, A simulator for Wireless Sensor Networks and Body Area Networks"*, Version 3.2, Users Manual, , March 2011.

[45] Marco Zuniga, Bhaskar Krishnamachari, *"Analyzing the Transitional Region in Low Power Wireless Links"*, First IEEE International Conference on Sensor and Ad hoc Communications and Networks, Santa Clara, CA, October 2004.

Chapter 5

[46] Babar Nazir , Halabi Hasbullah ,"Energy efficient and QoS aware routing protocol for Clustered Wireless Sensor Network Department of Computer Science, Institute of Information Technology, Abbottabad, Pakistan Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia (Energy and Quality of Service),Elsevier, 2013.

Chapter 7

[47] Article on web site: http://www.speedguide.net/faq_in_q.php?qid=374

Chapter 8

[48] Bijan Paul, Md.Ibrahim and Md. Abu Naser Bikas, *"Performance Evaluation of AODV & DSR with Varying Pause Time & Speed Time Over TCP & CBR Connections in VANET"*, International Journal of Computer Science and Network Security, VOL.11, July 2011.

[49] Figure from web site
http://www.techonlineindia.com/techonline/design_centers/171033/introduction-wireless-sensor-network-concepts

[50] I.F. Akyildiz, W. Su*, Y. Sankarasubramaniam, E. Cayirci, *"Wireless sensor networks: a survey"*, Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology,Atlanta, GA 30332, USA, December 2001.

[51] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, *"Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks",* Proceedings of ACM MobiCom'01, Rome, Italy, July 2001.

[52] Article on web site: http://store.apple.com/us/product/MA368LL/E/nike-ipod-sensor

[53] Article on web site: http://www.hotchips.org/wp-content/uploads/hc_archives/hc12/hc12pres_pdf/smart-dust-mote.pdf

[54] Article on web site: http://readwrite.com/2013/11/14/what-is-smartdust-what-is-smartdust-used-for#awesm=~oB3GmlcfIFb9aw

[55] Yao-Jung Wen, *"Smart Dust Sensor Mote Characterization, Validation, Fusion and Actuation"*, B.S. (National Taiwan University, Taipei, Taiwan), University of California, Berkeley, 1999.

[56] Brett Warneke, Sunil Bhave, *"Smart Dust Mote Core Architecture",* Project Report ,Berkeley Sensor and Actuator Center, Spring 2000.

[57] Article on web site: http://www.newscientist.com/article/mg21829146.400-smart-dust-computers-are-no-bigger-than-a-snowflake.html

[58] Michael Friedewald , Olivier Da Costa ,"Science and Technology Roadmapping: Ambient Intelligence in Everyday Life(AmI@Life)", June 2003.

[59] Atif Alamri, Wasai Shadab Ansari, Mohammad Mehedi Hassan, M. Shamim Hossain, Abdulhameed Alelaiwi, and M. Anwar Hossain *,"A Survey on Sensor-Cloud: Architecture, Applications, and Approaches"*,College of Computer and Information Sciences, King Saud University, Saudi Arabia, International Journal of Distributed Sensor Networks, Volume 2013.

Appendix A

Source Code

OMNeT++ 4.3.1 was used to simulate DDRA protocol. The following parts are the source code for the implementation. There are three main parts about Node module, Source module and Sink module.

I. Node Module

Part 1: Initialization of the Network

    a. Energy Resources

```cpp
void Node::SetInitialEnergy(cMessage *msg){

    cTopology *topo = new cTopology("topo");
    simtime_t now;
    simtime_t start=0.0;

// Extract the topology. We are only interested in hosts and routers.
        topo->extractByNedTypeName(cStringTokenizer("wireless15.Node wire-
less15.Sink wireless15.Source").asVector());

        for (int i=0; i<topo->getNumNodes();i++)
                {

                    // initialize table
                      energy[i][0]=i;
                      energy[i][1]=1000.0;

                }
            if (msg->getKind()==INITIAL_ENERGY){
                for (int i=0; i<15;i++){
                energy[i][1]=1000.0;
                }
            }
        else if (msg->getKind()==UPDATE)
        {
            // update battery status only of module getting index()

            now=simTime();
            int i=getIndex();
            double energyres=energy[i][1];
            //Update value energy
            energy[i][1]=energyres-energyspentvalue;
            EV<<"Updating energy resources in module ["<<i<<"] "<< "in time
"<<now<<"\n";

        }
        else if (msg->getKind()==TERMINATE){
            now=simTime();
```

Appendix A: Source Code of Node Module 146

```
        int i=getIndex();

        // set node to have exhausted battery
        energy[i][1]=0.0;
EV<<"The module ["<<i<<"] will be exhausted soon,Battery almost empty!
"<<endl;
        EV<<"Lasted : "<<now-start<<endl;
    }
    else
        EV<<"Failed giving info about Battery States"<<endl;
}
```

b. <u>Topology of Network</u>

```
void Node::PrintNeighbor(){

    cTopology *topo = new cTopology("topo");

    int Routes[15][2];
    int NumTwoHops[15][2] ;


    // Extract the topology. We are only interested in hosts and routers.
    topo->extractByNedTypeName(cStringTokenizer("wireless15.Node wire-
less15.Sink wireless15.Source").asVector());


    int hops;
    hops=0;
EV<<"___Distance to the Sink_____"<<endl;
 EV<<"____Neighbors and Information_____"<<endl;

EV<<"The Number of total nodes are: "<<topo->getNumNodes()<<endl;


    for (int i=0; i<topo->getNumNodes();i++)
    {
        cTopology::Node *node=topo->getNode(i);
        EV<<"Node i= "<<i<<" is "<<node->getModule()->getFullPath()<<endl;

EV<<"_____"<<endl;
        EV<<" It has "<<node->getNumOutLinks()<<" out links to other
nodes\n";
        EV<<"and "<<node->getNumInLinks()<<" in links from other nodes\n";

        // setting 0 hops for unconnected Nodes
        cTopology::Node *nodeunc=topo->getNode(i);

                    int numoutlinks= nodeunc->getNumOutLinks();
                     if (numoutlinks==0)
                        {
                        hops=0;
                        Routes[i][1]=0;
                        }
// for example Node[2] has no neighbors or links connected to others

        for (int j=0; j<node->getNumOutLinks();j++){
```

Appendix A: Source Code of Node Module   147

```cpp
            cTopology::Node *neighbor=node->getLinkOut(j)->getRemoteNode();

        EV<<"the neighbor nodes are ";
        EV<<" "<<neighbor->getModule()->getFullPath()<<"\n";
        EV<<" ID of module is "<<neighbor->getModuleId()<<"\n";


        for (int k=0; k<j-3; k++){
                cTopology::Node *sink=topo->getNode(topo->getNumNodes()-1);

                 cTopology::LinkOut *outl=neighbor->getLinkOut(k);

                 int outlid=outl->getLocalGateId();
// get the number of neighbors to 2-hops away


 for (int l=0;l<14; l++)
 {
     NumTwoHops[l][0]=l;
 }


 for (int j=0; j<14; j++)
  {
      NumTwoHops[j][1]=0;
  }

 for (int i=0; i<14; i++)
 {
            switch(i)
            {
            case 0:
            case 1:
            case 5:
            case 9:
            {
                NumTwoHops[i][1]=3;
                break;
            }

            case 2:
            case 4:
            case 8:
            case 11:
            {
                NumTwoHops[i][1]=4;
                break;

            }
            case 6:
            case 10:
            {
                NumTwoHops[i][1]=5;
                break;
            }

            case 7:
            {
                NumTwoHops[i][1]=6;
```

```
                break;
            }
            default:
            {
                NumTwoHops[i][1]=3;
                break;
            }
        } // end of switch

 } // end of for (i) NumTwoHops[i][j]

                int sinklocalid=sink->getLinkOut(0)->getLocalGateId();
                EV<<"the id of source out link is: "<<sinklocalid<<endl;


                if (((neighbor!=sink)&&(outlid!=sinklocalid))){
                    //
                     hops=hops+1;


                    }
                else
                    {
                    hops=hops+1;

                    EV<<"Reached destination, stop counting Hops"<<endl;
                    }

        } // end of (k)
} // end for (j)
    } // end of (for i all nodes in topology)
        // make a table with values of Nodes [i] and Routes [j]-numHops
            // initialize values of table Routes
            for (int j=0; j<14; j++){
                for (int i=0; i<1; i++){
                Routes[j][i]=0;
            }
        }

// set module number in the row [0] of table Routes
            for (int j=0;j<14; j++){
                Routes[j][0]=j;
                Routes[9][1]=hops-1;
                Routes[13][1]=hops;

            }
        int hopsnum=0;
        int k=1;

        // make the table NumTwoHops with all neighbors of each module have
in two-hops neighborhood
         for (int i=0; i<14; i++)
         {
        switch (i)
            {

        case 0:

            {
```

```cpp
        hopsnum=hops+3;
EV<<"------------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"------------------------------------------------------------"<<endl;
         Routes[i][k]=hopsnum;

           break;

        }
        case 5:
        {
          hopsnum=hops+4;
EV<<"------------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"------------------------------------------------------------"<<endl;
          Routes[i][k]=hopsnum;
          break;

        }
        case 1:
        {
             hopsnum=hops+8;
EV<<"------------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"------------------------------------------------------------"<<endl;
             Routes[i][k]=hopsnum;
              break;


          }
         case 3:
         case 4:
         case 6:
         {
             hopsnum=hops-1;
EV<<"------------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"------------------------------------------------------------"<<endl;
        Routes[i][k]=hopsnum;
              break;
         }
         case 7:
         case 8:

         {
             hopsnum=hops+1;
EV<<"------------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"------------------------------------------------------------"<<endl;
             Routes[i][k]=hopsnum;
             break;
         }
         case 10:
         case 12:
          {
            hopsnum=hops-2;
EV<<"------------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"------------------------------------------------------------"<<endl;
```

Appendix A: Source Code of Node Module   150

```
                    Routes[i][k]=hopsnum;
                     break;
                    }
                case 11:
                    {
                        hopsnum=1;
 EV<<"---------------------------------------------------------"<<endl;
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hopsnum<<endl;
EV<<"---------------------------------------------------------"<<endl;
                    Routes[i][k]=hopsnum;
                    break;
                    }
                default:
                    {

EV<<"---------------------------------------------------------"<<"\n";
EV<<"Hops of module [" <<i<<" ] to the Sink are :    "<<hops<<endl;
EV<<"---------------------------------------------------------"<<"\n";
                        break;
                        Routes[i][k]=hops;
                    }
                } // switch
            } // end of for i Routes [i][k]


    for (int j=0; j<14; j++){

EV<<"-----Information of Routes in Network--------------------------\n";
EV<<"The module ["<<Routes[j][0]<<" ]"<<"has number of hops to Sink:
"<<Routes[j][1]<<" \n";
EV<<"The number of neighbors in 2-Hops are: "<<NumTwoHops[j][1]<<" of mod-
ule ["<< NumTwoHops[j][0] <<"]  \n";
EV<<"Next Module_____\n";
        }
// calculate metrics about setting the Cluster Head Probability


    int totalnodes=topo->getNumNodes();
    // The maximum hops to get from Node to Sink is Node[1] to Sink
    int MaxDistance=8;
    int maxhops=3; // this is how large we want a cluster to be

    double CHProbability[15][2];

    // first calculate energy
    double Energy[15][2];


    for (int i=0; i<15; i++)
    {
            Energy[i][1]=1000.0;
            Energy[i][0]=i;

    }
// set the energy level of each node
        for (int p=0; p<15; p++)
        {
//calculate the energy remaining in nodes after exchanging messages
            EV<<"Table energy of Node ["<<p<<"] is: "<<Energy[p][1]<<endl;
```

```cpp
            }

// NeighborHood Around Nodes (neighbors of 2 Hops/Number of Nodes total)

    double NodesAround[15][2];

    for (int i=0; i<14; i++){
        for (int j=0; j<1; j++) {

        NodesAround[i][j]=0;
        NodesAround[i][0]=i;
        }
    } // end of for NodesAround

        // set table NodesAround Neighbors  in max 2-Hops  cluster / All
nodes in network
  for (int k=0; k<14; k++)
  {
      NodesAround[k][1]=NumTwoHops[k][1]/totalnodes;
  }

  double Distance[15][2];
  // This table will store the Distance in Hops of Node[i] to Sink minus
one
  // and divided by the max hops we want to have in a cluster
  // (how much large the cluster will be)

  for (int i=0; i<14; i++){
      for (int j=0; j<1; j++)
      {

      Distance[i][j]=0;
      Distance[i][0]=i;
      }
   } // end of for Distance

      for (int k=0; k<14; k++)
      {
          Distance[k][1]=MaxDistance-((Routes[k][1]-1)/maxhops);
      }
     // Now calculate the Information about CHProbability

      for (int i=0; i<14; i++)
      {
          for (int j=0; j<1; j++){
              CHProbability[j][i]=0;
              CHProbability[j][0]=j;
          }
      }

      for (int l=0; l<14; l++)
      {

CHProbability[l][1]=Energy[l][1]/1000+NodesAround[l][1]+Distance[l][1];
      }

      for (int i=0; i<14; i++){
          EV<<"The Module [ " <<i <<" ]"<<"has probability to become Clus-
ter Head in value of : "<<CHProbability[i][1]<<"\n";
```

Appendix A: Source Code of Node Module   152

```
            EV<<"This corresponds to the percentage of
"<<CHProbability[i][1]/100<< " % \n";

EV<<"_____\n";

        }

        // initialize array
                for (int i=0;i<14;i++){
                for (int j=0;j<2;j++){
                    CHArray[i][j]=0;
                        }
                }
                // insert module index in first row
                for (int i=0; i<14; i++){
                    CHArray[i][0]=i;
                }

        for (int i=0;i<14;i++){

            CHArray[i][1]=CHProbability[i][1]/100;

        }

for (int i=0; i<14; i++)
{

    CHall=CHArray[i][1]+CHall;
}
    averageCHvalue=((CHall)/14);
    averageCHp=&averageCHvalue;
    averageCHv=*averageCHp;

EV<<"------------------------PrintNeigbors()-------------------"<<endl;

 EV<<"The average CH of all nodes is: "<<averageCHv<<" % " <<endl;

 EV<<"----------------------------------------------------"<<endl;



} // end of PrintNeighbor()
```

Part 2: Clustering Formation

    a.   Create Clusters

```
void Node::setClusterHead(cMessage *msg)
{
    cTopology *topo = new cTopology("topo");
  // Extract the topology. We are only interested in hosts and routers.
topo->extractByNedTypeName(cStringTokenizer("wireless15.Node wire-
less15.Sink wireless15.Source").asVector());
    //Initialize energy resources of each Node
int nodeindex=getIndex();
int pos=1;
double CHPercentage;
```

Appendix A: Source Code of Node Module   153

```
CHPercentage=0.0;
CHPercentage=getCHvalue(nodeindex,pos);

EV<<"Percentage: "<<CHPercentage<<endl;

// calculate average value of possibilities to become Cluster Head)

double result1=round( averageCHv * 100.0 ) / 100.0;

double CHProp=round(CHPercentage* 100.0 ) / 100.0;
EV<<"--------------------Set Cluster Heads------------------------------
"<<endl;
EV<<"The average CH of all nodes is: "<<averageCHv<<" % " <<endl;
EV<<"Rounded in 2 decimal places : "<<result1<<" % " <<endl;
EV<<"The probability to become CH Node [ "<< nodeindex<<" ] is :
"<<CHPercentage<<"% "<<endl;
EV<<"-----------------------------------------------------------"<<endl;
              EV<< "Begin the cluster process"<<"\n";
              // Message arrived
// set variable result get value of pointer averageCHvalue
double *result;
result=&result1;
const double averageCHconst=*result;

 //  change the Display Image of Module and Show the Energy state
        int node1=getIndex();
         diff=CalculateEnergyLevel(node1);
         // set default graphics for Nodes
         diff=CalculateEnergyLevel(node1);

 if ((CHProp>=(averageCHconst))&&(diff>=(800.0))) {
         // generate a new message and broadcast this to the neighbors
         // broadcast the message

              int n = gateSize("gate");
              int k = intuniform(0,n-1);
            cMessage *clmsg = msg->dup();
             EV << "Forwarding cluster message " <<clmsg << " on gate[" <<
k << "]\n";
            send(clmsg, "gate$o", k);

            numExDataSent++;
            numExDataTotal++;

            CH=true;

            UpdateImage(node1,1);

          // set battery level
          batterylevel[node1][1]=1;
// become cluster head
      state[node1][1]=1;

                                              }
 else if
(CHProp>=(averageCHconst)&&(((800.0)>=(diff))||((diff)>=(500.0)))){
     // Same routing as before: random gate.
       int n = gateSize("gate");
        int k = intuniform(0,n-1);
```

```
                    cMessage *clmsg2 = msg->dup();
EV << "Forwarding cluster message " <<clmsg2 << " on gate[" << k << "]\n";
                    send(clmsg2, "gate$o", k);

                    numExDataSent++;
                    numExDataTotal++;

                     CH=true;
                     state[nodel][1]=1;
                    UpdateImage(nodel,1);
                    //set battery level
                    batterylevel[nodel][1]=2;


 }


 else if ((CHProp<averageCHconst)&&(((499.0)>=(diff))||((diff)>=(100.0))))
{
                    bubble("I am a cluster member");
                        state[nodel][1]=2;

        // generate a new message and broadcast this to the neighbors
        //  NodeMsg *tmsg = check_and_cast<NodeMsg *>(msg);

                        UpdateImage(nodel,2);
                        batterylevel[nodel][1]=2;
                    //  PrintStateBattery(nodel);


 }
 else if ((CHProp<averageCHconst)&&(((10.0)>=(diff))||((diff)<=(100.0)))) {

                        bubble("I am tired");
        cDisplayString& dispStr=getDisplayString();

dispStr.parse("i=device/pocketpc,yellow;i2=status/yellow");
                    UpdateImage(nodel,3);
                        // set battery level
                        batterylevel[nodel][1]=3;

                         state[nodel][1]=3;
 }

                        //just forward cluster message

                      if (diff==0.0){
                    bubble("Node exhausted");
                    UpdateImage(nodel,4);
                    // set battery level of node to be Dead
                    batterylevel[nodel][1]=4;
                    // update state energy of module
                    state[nodel][1]=4;

                                    }

                else if (diff<0.0)
                        {
                    UpdateImage(nodel,4);
                    SetBatteryState(4,nodel);
                            state[nodel][1]=4;
                        }
```

```
        // } // end of diff<2.0
 } // end of Cluster Member
/////////////////////////
```

b. Set Member of Cluster

```cpp
void Node::SetMember(int modid,cMessage *msg)
{
// this function sets the cluster members. If a module is a cluster member
// then find the Transmission channel and force to stop transmission at
this

// simulation time.

    int out=gateSize("gate$o");


    simtime_t now;

    now=simTime();

    cModule *modmember=getParentModule()->getSubmodule("Node",modid);
    cChannel *channel;

    for ( int i=0; i<out-1; i++)
    {
  cGate *gateout=modmember->gate("gate$o",i);

    if (hasGate("gate$o"))
    {


    channel=gateout->findTransmissionChannel();

 if ( channel->isBusy()==true)
  {
      channel->forceTransmissionFinishTime(now);
      channel=NULL;
  }
        }    // if you have gate-output type
    } // for


    EV<<"Now you are a Cluster Member"<<endl;


}
```

c. Dead Node Management

```cpp
void Node::DisconnectGate(int modid)
{
    // module id is the index of called module that is Dead

    simtime_t now;
    EV<<"Node Dead-Stop Operation"<<endl;
    int out=gateSize("gate$o");
    // int in=gateSize("gate$i");
```

Appendix A: Source Code of Node Module   156

```
    cModule *mod=getParentModule()->getSubmodule("Node",modid);

    now=simTime();
    // if module is dead, disconnect gates so that can not receive or
transmit messages

    if (hasGate("gate$o") &&(gate("gate$o",out-1)->isConnected())==true)
    {
    for (int i=0; i<out-1; i++)
    {
        mod->gate("gate$o",i)->disconnect();

    }
    }

}
```

Part 3: Setting Up Gradients and Exploratory Data Transmission

    a. Gradients pointing to source of Interest Message

```
void Node::MakeGrad(int i,int j)
{

    char msgn[20];
    sprintf(msgn, "Gradient of IntMsg of %d",i);

    Gradient *grintmsg=new Gradient(msgn);
    grintmsg->setNodeid(i);
    grintmsg->setSourceInt(j);
}
```

```
void Node::MakeGradient(){
        int src=getIndex();
        int n=gateSize("gate$o");
        int dest = intuniform(0,n-1);

        cModule *prev=gate("gate$i",0)->getPreviousGate()->getOwnerModule();

        int modid=prev->getId();

         char msgn[20];
        sprintf(msgn, "Gradient from %d-to-%d", src, dest);

         // Create message object and set source and destination field.
         Gradient *grmsg = new Gradient(msgn);
         grmsg->setNodeid(src);
         grmsg->setSourceInt(modid);

}
```

```
Gradient *Node::UpdateGradient(int ind,Gradient *grmsg)
{
    if (grmsg!=NULL)
    {
```

Appendix A: Source Code of Node Module   157

```
        int i=getIndex();
        cModule *prev=gate("gate$i",0)->getPreviousGate()-
>getOwnerModule();

        int modid=prev->getIndex();

        int src=getIndex();
        int n=gateSize("gate$o");
        int dest = intuniform(0,n-1);

        char msgn[20];
        sprintf(msgn, "Gradient from %d-to-%d", src, dest);

        Gradient *msg = new Gradient(msgn);
        msg->setNodeid(i);
        msg->setSourceInt(modid);



        // print data
        int node=msg->getNodeid();
        int source=msg->getSourceInt();
        EV<<"Gradient Message"<<endl;
        EV<<"Module is: "<<node<<endl;
        EV<<"Module send Interest Source is: "<<source<<endl;
        return msg;

    }
    return msg;
}
```

```
void Node::SendGradient(Gradient *grmsg)
{

    EV<<"Sending to next gradient ok"<<endl;
    finished=false;

    if (gate("gate$o",0)->isConnected()==true)
    {
    if (hasGate("gate$o"))
        {
    int n=gateSize("gate$o");
    int k=n-1;
for (int i=0; i<k; i++)
    {
        if  (gate("gate$o",k)->isConnected()==true)
        {
        Gradient *grcopy=grmsg->dup();
        send(grcopy,"gate$o",i);

        numGrSent++;
        numGrTotal++;
        finished=false;
        }
        else
        {
   EV<<"new gate reconnected for gradient transmission"<<endl;
            finished=false;
        }
```

```cpp
        } // for
} // if has Gate(gate$o)

else if (hasGate("out") && gate("out")->isConnected()==true)
            {
        Gradient *grcopy=grmsg->dup();
        send(grcopy,"out");
        finished=false;
        numGrSent++;
        numGrTotal++;
            }

    } // if gate,0 is connected()

    // if all nodes around source are dead stop simulation
    else if
((state[6][1]==4)||(state[10][1]==4)||(state[3][1]==4)||(state[4][1]==4))
    {
        finished=true;
        EV<<"No routing any more!"<<endl;
        EV<<"Nodes Around Source are Dead"<<endl;
        EV<<"Stop Network Communication"<<endl;
        EV<<"---------------------------------------"<<endl;
        endSimulation();
    }
} // end of function
```

b. Exploratory Data

```cpp
bool Node::AcceptGrad(Gradient *grmsg,int eventid)
{
    const char *className;
    className=grmsg->getClassName();

    if ((strcmp(className,"Gradient"))==0)
            {
        int n=gateSize("gate$o");
        int k=intuniform(0,n-1);

// if module accepted Gradient, continue forward it

    Gradient *gradcopy=grmsg->dup();
    send(gradcopy,"gate$o",k);
  // check if you have the same event id with Event Table
 int i=getIndex();
EV<<"Gradient reached at module "<<i<<endl;
EV<<EventTable[i][0]<<" Node has eventid  " <<EventTable[i][1]<<endl;

// Make comparison to create exploratory data
        if ((EventTable[i][0]==i)&&(EventTable[i][1]==eventid))
        {
//start sense data environment and send it to those you have gradients
            EV<<"Can make exploratory data"<<endl;
            EV<<"Module"<<i<<"has same data on Module
"<<EventTable[i][0]<<"with eventid "<<eventid<<endl;
            return true;
        }
```

Appendix A: Source Code of Node Module 159

```
                else
                    {
                        EV<<"No matching fields"<<endl;
                        return false;
                    }

        } // if strcmp is gradient message
    return false;

}
```

```
void Node::ExData(Gradient *grmsg,int i)
{
    if (grmsg!=NULL)
    {
        //get values of gradient message
        int node=grmsg->getNodeid();
        int src=grmsg->getSourceInt();
        EV<<"Gradient has following INFO:"<<endl;
        EV<<"Node Having Match is "<<node<<endl;
        EV<<"Has gradient to Module: "<<src<<endl;

    // make a type message cMessage and send it to the src(neighbor that
    // points through gradient
        cMessage *exploratoryData=new cMessage();
// Stage 4: If there is a match on Eventid and Gradient
// send to node you have gradient for

        cModule *target=getParentModule()->getSubmodule("Node",src);
        int n=gateSize("gate$o");
        int k=n-1;

        if (hasGate("gate$o")&&(gate("gate$o",k)->isConnected()==true))
            {
          send(exploratoryData,"gate$o",k);
            cDisplayString& dispStr=getDisplayString();
            dispStr.parse("i=device/pocketpc,pink");
            }
        else if (hasGate("out"))
            {
            send(exploratoryData,"out");

            cDisplayString& dispStr=getDisplayString();
            dispStr.parse("i=device/pocketpc,pink");
            }

        int targetid=target->getIndex();
        matchid=targetid;
      EV<<"Exploratory has been forwarded to "<<targetid<<endl;
            } // if msg !=NULL
}
```

Part 4: Optimal Path Establishment

    a.   Path Reinforcement

```
OptimalPath *Node::CreateOptPath(Gradient *grmsg,int matchid,int eventid)
```

Appendix A: Source Code of Node Module   160

```
{
    //this is a function to create the optimal path for module index
    //Initialize message

      int src = getIndex();   // our module index
      int Targetid;
      int Targetindex;
// make the target of sending message for optimal path
// set the target module be the one that have match on eventid
// matchid is equal targetid of function ExData, the one that Gradient came

  cModule *target=getParentModule()->getSubmodule("Node",matchid);

      Targetid=target->getId();
      Targetindex=target->getIndex();

// we will use targetindex for optimal path
      int optnode=0;
      char optname[20];
      sprintf(optname, "Optimal Path" );


        // Create message object and set source and destination field.
        OptimalPath *optmsg=new OptimalPath(optname);
        optmsg->setSrcid(src);
        optmsg->setEventid(eventid);
        optmsg->setSinkid(15);
        optmsg->setSourceid(14);
        optmsg->setTargetid(Targetid);
        optmsg->setTargetindex(Targetindex);

        int targetindex=optmsg->getTargetindex();

      optnode=targetindex;
      bestnode=ReturnOptimalNode(optnode);

       //process to build optimal path table
      RecordPath(optmsg);

return optmsg;
}
```

b. Optimal Path Establishment

```
void Node::RecordPath(OptimalPath *optmsg)
{

    // use a queue to insert the optimal path, created in the end of trans-
mitting
    // exploratory data and print this node in Genv of Omnet++

    cQueue *queuemsg=new cQueue("queuemsg");
    OptimalPath *optimal=new OptimalPath("optimal");
    cPacket *DPkt=new cPacket();
      queuemsg->setTakeOwnership(1);

        if (optmsg!=NULL)
        {
            queuemsg->insert(optmsg);
```

Appendix A: Source Code of Node Module   161

```
                }
    if ((queuemsg->isEmpty())!=true)
     {
        EV<<"queue msg is not empty-optimal path"<<endl;

        optimal=(OptimalPath *)queuemsg->pop();
        int bestpath=optimal->getTargetindex();
        EV<<"-------------------------------------"<<endl;
        EV<<"Print path : Next best node is :"<<bestpath<<endl;
        EV<<"-------------------------------------"<<endl;
        cModule *target=getParentModule()->getSubmodule("Node",bestpath);
        SendPacket(DPkt,target);
     }
}
```

    c.   Data Packets Forwarding through path

```
void Node::SendPacket(cPacket *DataPkt,cModule *target)
{
    // count the average end-to-end delay
    // Difference of Time (Arrive at Sink)-Time (Send from Source)
    // Divide with total packets arrived successfully at Sink

    simtime_t creat_time;
    simtime_t arrive_time;
    simtime_t eedelay;
    int modid;
    int targetid;
    modid=getIndex();
    targetid=target->getIndex();

    if ((DataPkt!=NULL))
    {
        cPacket *DataPcp=DataPkt->dup();
        creat_time=DataPcp->getCreationTime();

 EV<<"Packet created and Forwarded from [ "<<modid<<" ] at
"<<creat_time.dbl();

        sendDirect(DataPcp,target,"gate$o",0);
        cModule* arrivedmod=DataPcp->getArrivalModule();
        cModule* sendermod=DataPcp->getSenderModule();


        if (arrivedmod!=sendermod)
        {
            succPackets++;
            EV<<"Data Packet delivered ok!"<<endl;
            arrive_time=DataPcp->getArrivalTime();
            eedelay=arrive_time-creat_time;
EV<<"Packet End to End Delay to [ "<<targetid<<" ] is
"<<eedelay.dbl()<<endl;
            endToendDelayVec.record(eedelay);
        }
        else
        {
            lostPackets++;
            arrive_time=DataPcp->getArrivalTime();
            eedelay=arrive_time-creat_time;
```

Appendix A: Source Code of Node Module   162

```
            EV<<"Packet Lost End to End Delay to [ "<<targetid<<" ] is
"<<eedelay.dbl()<<endl;

        }

        numPackets++;

        if (succPackets!=0)
        {
            double delaydb=eedelay.dbl();

        avgeedelay=delaydb/succPackets;
        EV<<"Average delay is :"<<avgeedelay<<endl;
        AvgDelayVec.record(avgeedelay);

        }
        packetlossratio=lostPackets/(lostPackets+succPackets);
        packetloss=(round)(packetlossratio*100.0)/100.0;
        /////record scalar vectors for statistics////////////////
        packetlossVec.record(packetloss);
        numPacketsVec.record(numPackets);
        lostPacketsVec.record(lostPackets);
        succPacketsVec.record(succPackets);

    }
}
```

Part 5: Recording Statistics

    a.   End of Simulation-Calling Finish() class

```
void Node::finish()
{
     int n=getIndex();


    endsimtime=simTime();


    // This function is called by OMNeT++ at the end of the simulation.
    EV << "Sent:     " << numSent << endl;
    EV << "Received: " << numReceived << endl;
    EV<<"  Residual Energy of Module with Index [ "<<n<<"] is : "<<ener-
gy[n][1]<<endl;
    EV <<" Hops: " << hopcount << endl;
    EV<<" Path Loss dB is: "<<pathloss_db<<endl;
    EV<<" Interest Messages Sent are:"<<numInterestSent<<endl;
    EV<<" Interest Messages Received are:"<<numInterestReceived<<endl;
    EV<<" Total Messages Sent are:"<<numtotalSent<<endl;
    EV<<" Total Messages Received are:"<<numtotalReceived<<endl;
    EV<<" Simulation end time is: "<<endsimtime<<endl;
    EV<<" Total Energy Spent in milli Joule (mJ) is:
"<<energyspentunit<<endl;

    EV<<"---------------------------------------------------------"<<endl;
    EV<<"Total Interest Messages: "<<numIntTotal<<endl;
    EV<<"Total Ex-Data Messages: "<<numExDataTotal<<endl;
```

```cpp
    EV<<"Total Data Messages: "<<numDataTotal<<endl;
    EV<<"Total Optimal Path Messages: "<<numOptTotal<<endl;
    EV<<"--------------------------------------------------------"<<endl;
    EV<<"Count the statistics"<<endl;
    EV<<"_____"<<endl;
    EV<<"Total Data Messages in Network: "<<TotalData<<endl;
    EV<<"Total Alive Data Messages: "<<TotalAliveMsgData<<endl;

    EV<<"Total Interest Messages: "<<TotalInt<<endl;
    EV<<"Total Alive Interest Messages: "<<TotalAliveMsgInt<<endl;

    EV<<"Total Gradients Messages in Network: "<<TotalGr<<endl;
    EV<<"Total Alive Gradients Messages: "<<TotalAliveGr<<endl;

    EV<<"Total Optimal Path Messages: "<<TotalOpt<<endl;
    EV<<"Total Alive Optimal Path Messages: "<<TotalAliveOpt<<endl;

    EV<<"Total Alive Ex-Data Messages: "<<TotalAliveMsgExData<<endl;
    EV<<"Total Ex-Data Messages: "<<TotalExData<<endl;
    EV<<"Total Data packets are: "<<numPackets<<endl;
    EV<<"Total Succesfully Delivered Data Packets are:
"<<succPackets<<endl;
    EV<<"Total Lost Packets are: "<<lostPackets<<endl;
    EV<<"End to End Delay is:"<<avgeedelay<<endl;
    EV<<"Packet Loss Ratio of Module ["<<n<<" ]is: "<<packetloss<<"or
"<<packetloss*100<<" %"<<endl;

EV<<"_____"<<endl;


    energylast=energy[n][1];
    energylastVec.record(energylast);


    recordScalar("Packet Loss",packetloss);
    recordScalar("Number of Packets",numPackets);
    recordScalar("Number of Succesfully Delivered Packets",succPackets);
    recordScalar("Number of Lost Delivered Packets",lostPackets);
    recordScalar("Messages Sent", numSent);
    recordScalar("Messages Received", numReceived);
    recordScalar("Total Messages Send ",msgsend);
    recordScalar("Total Messages",msgTotal);
    recordScalar("Hop Counts",hopcount);
    recordScalar("Residual Energy",residualenergy);
    recordScalar("Energy at the End of Simulation",energylast);
    recordScalar("Path loss in dB",pathloss_db);
    recordScalar("Messages Interest Sent", numInterestSent);
    recordScalar("Messages Interest Received", numInterestReceived);
    recordScalar("Total Messages Sent ",numtotalSent);
    recordScalar("Total Messages Received",numtotalReceived);
    recordScalar("Energy Spent Unit", energyspentunit);
    //////////////////////////////////////////////////////////////
    recordScalar("Total Data messages Sent",numDataSent);
    recordScalar("Total Data messages Received",numDataRec);
    recordScalar("Total Data messages ",numDataTotal);

    recordScalar("Total Ex-Data messages Sent",numExDataSent);
    recordScalar("Total Ex-Data messages Received",numExDataRec);
    recordScalar("Total Ex-Data messages ",numExDataTotal);
```

```
    recordScalar("Total Opt-Data messages Sent",numOptSent);
    recordScalar("Total Opt-Data messages Received",numOptRec);
    recordScalar("Total Opt-Data messages ",numDataTotal);

    recordScalar("Total Gradient messages Sent",numGrSent);
    recordScalar("Total Gradient messages Received",numGrRec);
    recordScalar("Total Gradient messages ",numGrTotal);

    recordScalar("Total Interest messages Sent",numIntSent);
    recordScalar("Total Interest messages Received",numIntRec);
    recordScalar("Total Interest messages ",numIntTotal);
    ///////////////////////////////////////////////////////////////
    recordScalar("Total Data Count",TotalData);
    recordScalar("Total Data Alive",TotalAliveMsgData);
    recordScalar("Total Int Count",TotalInt);
    recordScalar("Total Int Alive",TotalAliveMsgInt);
    recordScalar("Total Gradient Count",TotalGr);
    recordScalar("Total Gradient Alive",TotalAliveGr);
    recordScalar("Total Optimal Path Count",TotalOpt);
    recordScalar("Total Optimal Path Alive",TotalAliveOpt);
    recordScalar("Total Ex-Data Count",TotalExData);
    recordScalar("Total Ex-Data Alive",TotalAliveMsgExData);
    recordScalar("End-To-End Delay",avgeedelay);
    recordScalar("Average Delay Results",avgeedelay);



    EV<<"--------------------------------------------------------"<<endl;

    ///////////////////////////////////////////////////////////////

    histogram.collect(hopcount);
    histogramenergy.collect(residualenergy);



    StatNumSent.record();
    StatNumReceived.record();
    StatHopCount.record();
    StatMsgSend.record();
    ///////////////////////////////////

    histogramsent.collect(numSent);
    histogramrec.collect(numReceived);

    hopCountStats.recordAs("hop count");


    histogram.recordAs("Hop Counts");
    histogramsent.recordAs("Number of Total Messages Sent");
    histogramrec.recordAs("Numbers of Messages Received");
    histogramenergy.recordAs("Energy Remaining in Module");


}
```

II. Source Module

Part 1 : Handling Different Type Messages

Handle Message() class

```cpp
void Source::handleMessage(cMessage *msg)
{

    const char *classN;
        classN=msg->getClassName();

        // record sink for statistical values-number of messages
            cModule *source=getParentModule()->getSubmodule("Source");
            snapshot(source);

        if ((strcmp(classN,"NodeMsg"))==0)
        {
    NodeMsg *ttmsg = check_and_cast<NodeMsg *>(msg);

    if (ttmsg->getDestination()==getIndex())
            {

        // Message arrived
          int hopcount = ttmsg->getHopCount();
            // send a signal
          emit(arrivalSignal, hopcount);
EV << "Message " << ttmsg << " arrived after " << hopcount << " hops.\n";

                numReceived++;
                numDataTotal++;
            numData=ttmsg->getLiveMessageCount();
            numDataVec.record(numData);
                    delete ttmsg;
                    bubble("ARRIVED, starting new one!");

        // Generate another one.
            EV << "Generating another message: ";
            NodeMsg *newmsg = generateMessage();
            forwardMessage(newmsg);

            numDataTotal++
            numSent++;
            numDataTotalVec.record(numDataTotal);
            numData=ttmsg->getLiveMessageCount();
            numDataVec.record(numData);


            if (ev.isGUI())
             updateDisplay();
                }
         else // if Msg didn't arrived at source
            {
        // We need to forward the message.
          forwardMessage(ttmsg);
                } // if ttmsg!=Destination
        } // if msg is NodeMsg type

        else if ((strcmp(classN,"InterestMsg"))==0)
```

```cpp
      {
InterestMsg *intmsg = check_and_cast<InterestMsg *>(msg);
              SendInterest(intmsg);
          bool arrived;
          numIntSent++;
          numIntRec++;
          numIntTotal++;
      numInt=intmsg->getLiveMessageCount();
       numIntVec.record(numInt);


          arrived=InterestAccepted(intmsg);
           numIntRec++;
           numIntTotal++;

           bubble("Got Interest");
             if (arrived==true)
               {
                 EV<<"ok!"<<endl;
                          }
          else if ((strcmp(classN,"cMessage"))==0)
           {

if ((SourceAcceptEx(ExData)==true)||(SourceAcceptEx(ExDatadup)==true)
{
   cModule *sender=msg->getSenderModule();
      int senderindex=sender->getIndex();
     EV<<"Exploratory Data came from Node[ "<<senderindex<<"]"<<endl;

                              numDataTotal++;
                              numExData=msg->getLiveMessageCount();
                              numExDataVec.record(numExData);

      cModule *arrivalmod=msg->getArrivalModule();
      int arrivalid=arrivalmod->getId();
                // id of sink is 15
EV<<"Arrival must equals (14) is:  "<<arrivalid<<" sink id"<<endl;

EV<<"Source can start reinforcement"<<endl;
// change icon of source, source now can send optimal path reinforecment

// colorize sink to antenna tower (accept  last gradient)
      cDisplayString& dispStr=getDisplayString();
      dispStr.parse("i=device/antennatower");
               cMessage *ExData = check_and_cast<cMessage *>(msg);

             int exdataid=ExData->getId();
             EV<<"the id of exdata came to source is: "<<exdataid<<endl;
             EV<<"Source got Exploratory Data from Sink"<<endl;
             numDataTotal++;
            }
          else {
  EV<<"Exploratory  Data didn't arrived yet"<<endl;
            }
            }

   else   {
EV<<"Messages Didn't Arrived yet! at Source!"<<endl;
```

Appendix A: Source Code of Sink Module   167

```
              }
          }  // if strcmp InterestMsg

} // end of handleMessage
```

b. Generating Interest Message

```cpp
InterestMsg *Source::generateIntMessage()
{
   // Produce source and destination addresses.
   int src = getIndex();   // our module index
   int n = size();         // module vector size
   int dest = intuniform(0,n-2);
   if (dest>=src) dest++;

   char msgname[20];
   sprintf(msgname, "Interest-%d-to-%d", src, dest);
   char msgintname[20];
      sprintf(msgintname, "Interest from %d-to-%d", src, dest);

      // Create message object and set source and destination field.
      InterestMsg *intmsg2 = new InterestMsg(msgintname);
      intmsg2->setSource(src);
      intmsg2->setDestination(dest);
      intmsg2->setHopCount(intmsg2->getHopCount()+1);
      intmsg2->setSinkid(15);
      intmsg2->setEventid(2);


      return intmsg2;
}
```

III. Sink Module

Part 1: Accept Exploratory Data and Start Path Reinforcement

    a.   HandleMessage() Class- Accept Exploratory Data

```
.....(source code similar with HandleMessage() of Source Module(Part II))

else if (((strcmp(classN,"cMessage"))==0))
     {
             if (SinkAcceptExData(ExData)==true)
               {

                 // start print INFO for ExData
                 cModule *sender=msg->getSenderModule();
                 int senderindex=sender->getIndex();
EV<<"Exploratory Data came from Node[ "<<senderindex<<"]"<<endl;



                 cModule *arrivalmod=msg->getArrivalModule();
                 int arrivalid=arrivalmod->getId();

                 // id of sink is 15
EV<<"Arrival must equals (15) is:  "<<arrivalid<<" sink id"<<endl;
EV<<"Sink can start reinforcement"<<endl;

                 numExData=msg->getLiveMessageCount();
                 numExDataVec.record(numExData);


                 // Sink has accepted exporatory
                 // colorize sink to antenna tower (accept  last gradient)
                   cDisplayString& dispStr=getDisplayString();
                    dispStr.parse("i=device/antennatower");
                    bubble("I got Exploratory Data!");
                    bubble("Path Reinforcement");

                  SendSinkExData(ExData);
                  } //end of SinkAcceptEx
       }
```

    b.   Accept Exploratory data

```
// This function determine the last module that ExData reaches Sink.
// Sink now will transmit to last module (from which it got ExData) the Op-
timal Path Message.
//This is the reinforcement message for optimal Path.
bool Sink::SinkAcceptExData(cMessage *ExData)
{
    if ((ExData!=NULL))
    {
```

```
cModule *lastmodule=gate("in")->getPreviousGate()->getOwnerModule();
 EV<<"The last module ExMessage to Sink is:"<<lastmodule<<endl;
        cModule *sink=getParentModule()->getSubmodule("Sink",-1);
        int sinkid=sink->getId();
        int i=getId();
        if (i==sinkid)
        {
EV<<"Gradient reached sink! Now Sink can send reinforcement"<<endl;
            return  true;
        }
        else
            return false;
    } // end of if
        return false;
}
```