August 2016

An Energy Efficient, Load Balancing, and Reliable Routing Protocol for Wireless Sensor Networks

Kamil Samara

University of Wisconsin-Milwaukee

AN ENERGY EFFICIENT, LOAD BALANCING, AND RELIABLE ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Kamil Samara

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin – Milwaukee

August 2016

ABSTRACT

AN ENERGY EFFICIENT, LOAD BALANCING, AND RELIABLE ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Kamil Samara

The University of Wisconsin-Milwaukee, 2016
Under the Supervision of Professor Hossein Hosseini

The Internet of Things (IoT) is shaping the future of Computer Networks and Computing in general, and it is gaining ground very rapidly. The whole idea has originated from the pervasive presence of a variety of things or objects equipped with the internet connectivity. These devices are becoming cheap and ubiquitous, at the same time more powerful and smaller with a variety of onboard sensors. All these factors with the availability of unique addressing, provided by the IPv6, has made these devices capable of collaborating with each other to accomplish common tasks.

Mobile AdHoc Networks (MANETS) and Wireless Sensor Networks (WSN) in particular play a major role in the backbone of IoT.

Routing in Wireless Sensor Networks (WSN) has been a challenging task for researchers in the last several years because the conventional routing algorithms, such as the ones used in IP-based networks, are not well suited for WSNs because these conventional routing algorithms heavily rely on large routing tables that need to be updated periodically. The size of a WSN could range from hundreds to tens of thousands of nodes, which will make routing tables' size very large. Managing large routing tables is not feasible in WSNs due to the limitations of resources.

The directed diffusion algorithm is a well-known routing algorithm for Wireless Sensor Networks (WSNs). The directed diffusion algorithm saves energy by sending data packets hop by hop and by enforcing paths to avoid flooding. The directed diffusion algorithm does not attempt to find the best or healthier paths (healthier paths are paths that use less total energy than others and avoid critical nodes). Hence the directed diffusion algorithm could be improved by enforcing the use of healthier paths, which will result in less power consumption.

We propose an efficient routing protocol for WSNs that gives preference to the healthier paths based on the criteria of the total energy available on the path, the path length, and the avoidance of critical nodes. This preference is achieved by collecting information about the available paths and then using non-incremental machine learning to enforce path(s) that meet our criteria.

In addition to preferring healthier paths, our protocol provides Quality of Service (QoS) features through the implementation of differentiated services, where packets are classified as critical, urgent, and normal, as defined later in this work. Based on this classification, different packets are assigned different priority and resources. This process results in higher reliability for the delivery of data, and shorter delivery delay for the urgent and critical packets.

This research includes the implementation of our protocol using a Castalia Simulator. Our simulation compares the performance of our protocol with that of the directed diffusion algorithm. The comparison was made on the following aspects:

- Energy consumption

- Reliable delivery

- Load balancing

- Network lifetime

- Quality of service

Simulation results did not point out a significant difference in performance between the proposed protocol and the directed diffusion algorithm in smaller networks. However, when the network's size started to increase the results showed better performance by the proposed protocol.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

ACKNOWLEDGMENTS

First, I thank Allah almighty for granting me the chance to complete my PhD studies.

I would also like to acknowledge and appreciate the efforts and guidance of my advisor, Professor Hossein Hosseini, and the members of my Ph.D. committee: Professors K. Vairavan, Ethan Munson, Adel Nasiri, and Hamid Seifoddini.

I would like to express my special thanks to my mother-in-law for her help.

Most importantly, I would like to express my greatest appreciation to my wife Hana for her support and for her belief in me.

1 Introduction

1.1 Problem Statement

The Internet of Things (IoT) is shaping the future of networks and computing in general. With broadband connectivity becoming cheap and ubiquitous, devices are becoming more powerful and smaller, with a variety of onboard sensors; physical objects are becoming part of the internet. This phenomenon will open the door to various application domains ranging from Green-IT and energy efficiency to military applications.

Mobile AdHoc Networks (MANETS) and Wireless Sensor Networks (WSN) play a major role in the backbone of IoT [1].

Routing in Wireless Sensor Networks (WSN) has been a challenging task for researchers in the last several years. It is a challenge because the conventional routing algorithms, like ones used in IP-based networks, are not suited for WSNs because these conventional routing algorithms rely on routing tables. WSNs' size could range from hundreds to tens of thousands of nodes, thus making the size of routing tables very large. Managing large routing tables is not feasible in WSNs due the hardware limitations.

Although WSNs are considered a subcategory of wireless Ad Hoc networks, the routing techniques used in AdHoc networks are not suited for WSNs. Routing techniques in Ad Hoc networks assume global knowledge of the whole network, so each packet sent will contain the full path from the source to destination. Embedding full path in the packet

header will make data packets larger in size, which will result in the consumption of more energy [2].

Energy limitation is the main constraint in WSNs. WSNs are operated by a finite energy source (usually batteries), so any routing algorithm to be used in WSNs should take this in consideration, besides the hardware limitation of WSNs' nodes. Energy aware routing algorithms have proven to reduce power consumption substantially.

1.2   Research Questions

In this research we will address the following question: Can we improve on the existing routing techniques to achieve more power conservation while maintaining load balancing and quality of service?

The remaining sections of this dissertation are organized as follows. The first section describes the Internet of Things, a background section that defines WSNs and explains the challenges and applications of WSNs. The next section discusses solar energy and how it can be used to charge mobile devices such as WSNs' nodes. The network layer section will briefly review the previous work on the routing level with some focus on the Directed Diffusion algorithm. The dissertation concludes with a conclusions and future work sections.

2 Internet of things

Due to the improvements in technology, the availability of networks and the internet, and the low cost of communication devices, access to the internet is available to more people with smaller devices at lower rates. Individuals can access the internet through their handheld devices such as phones, tablets, notebooks, or any device with processing power and connectivity. All of these small devices are equipped with sensors and actuators that give them the ability to interact with the surrounding environment [3].

On the other hand, physical objects are fitted with tags that make it possible to identify these objects in addition the availability of IP addresses after IPv6 was introduced. This combination makes it possible to link the physical world to cyberspace through handheld devices, thus making the Internet of Thing (IoT) a new reality [3, 4].

The "things" in the IoT include a wide range of physical elements. These could include portable personal objects such as smartphones and tablets. The IoT also includes elements from the environment fitted with tags that can be connected to the internet. Each element can provide data and perform services if needed [3].

Figure 1 shows the general components of the IoT system. Things could be identified through scanning their tag IDs, thus communicating the location of the thing. Networked things fitted with sensors and an actuator can interact with the environment, sending data to higher services. Smart things can sense activities and collect data, linking them to the IoT. Middleware handles the communication between networked things and the

application layer, enabling application and service development to utilize data as received from things. Being connected to the cloud provides the capability to add intelligence, resulting in better services [5].
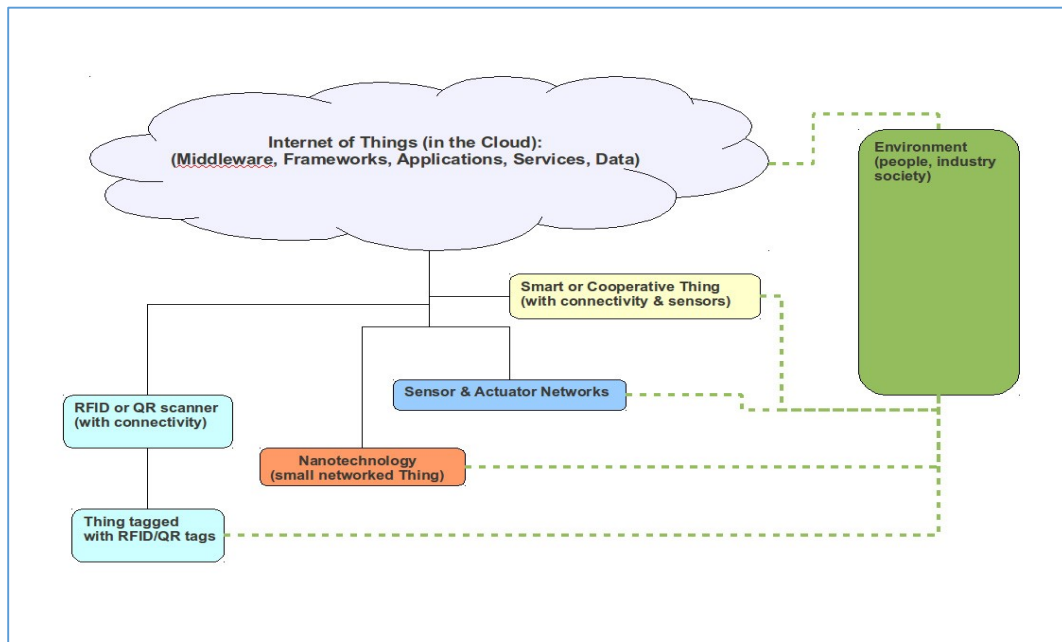
IoT will have a high impact on several aspects of our lives. On the individual level, some examples could be e-health, smart houses, and e-learning. On the business level, a few examples include automation and industrial manufacturing, logistics, business/process management, and intelligent transportation of people and goods [4].

2.1   MANETs and WSNs in IoT

Cheap and easy to deploy wireless sensor networks (WSNs) and the widespread use of mobile ad hoc networks (MANETs) open the way to a wide range of applications that could be implemented through IoT [1].

4

Wireless Sensor networks can be used to track objects, movement, and collect data from the environment. With these capabilities, wireless sensor networks can bridge the gap between the physical world and the digital world. This bridge will bring intelligence to wireless sensor networks applications [1, 4].

3 Wireless Sensor Networks

3.1 Motivation

The need to monitor civil infrastructures (like tunnels) and natural resources (like forests) motivated the design and implementation of distributed wireless sensor networks [6, 7].

With the technological advances in the last few years especially in the fields of printed circuits, microelectromechanical systems (MEMS), and wireless communications made the use of sensor networks a reality.

3.2 Definitions and Background

3.2.1 Sensing and Sensors

Sensing is the process of collecting data from the physical world (like temperature) by using sensors. Figure 2 shows the details of the sensing process.

A sensor is a device that captures a physical phenomenon and converts it into electrical signals which then is passed to a controller to be processed [6].
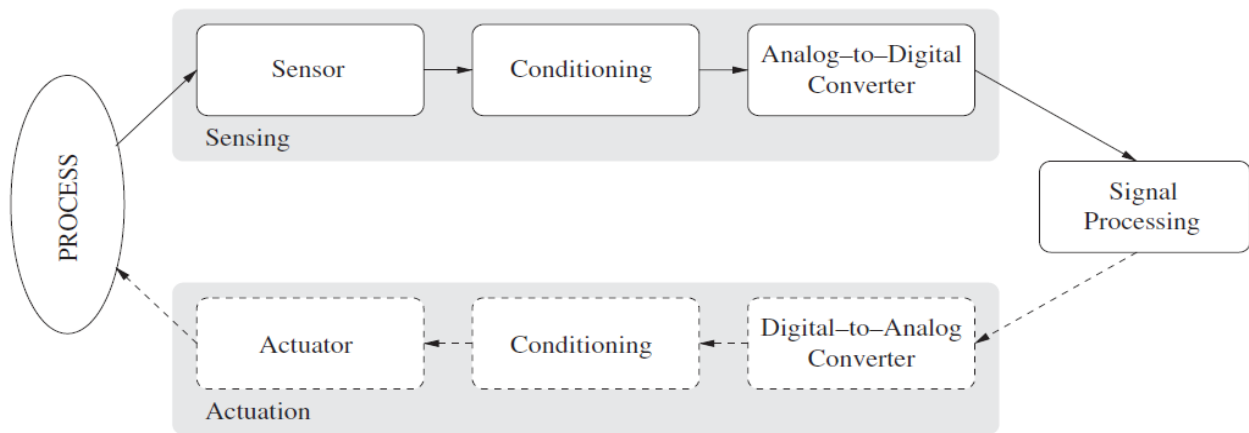
Figure 2: Sensing Process (diagram taken from [12, page 4])

3.2.2 Wireless Sensor Networks

Wireless Sensor Networks are a collection of hundreds or thousands of wireless sensor nodes that are often deployed in remote areas, whose job is to collect data wirelessly and deliver it to the base station. Each node consists of a sensing component, processor, communication, and storage components [8].
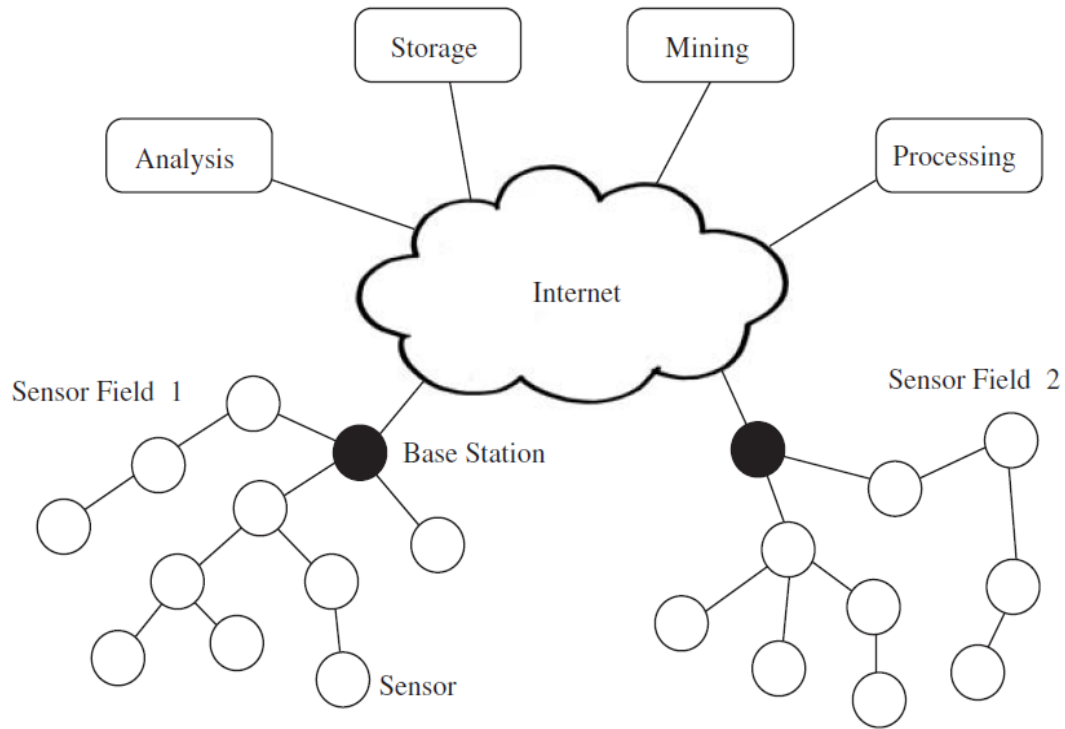
Figure 3: Wireless Sensor Network (diagram taken from [12, page 7])

3.3 Challenges and Constraints

3.3.1 Energy

Energy is the most important constraint that affects wireless sensor networks' design at all layers. Sensor nodes are usually operated with batteries. Moreover, batteries must either be replaced or charged after a certain period of use, and this is not always applicable in the case of sensor nodes because they are usually deployed in remote areas [9, 10].

3.3.2 Self-Management

As mentioned earlier, wireless sensor networks are usually deployed in remote areas, which implies that no infrastructure is available, and maintenance operation is almost impossible.

Therefore, the nodes have to adapt to these constraints and environment changes using self-management through teamwork without human interaction [10, 11].

3.3.3 Wireless Networking

Dependence on wireless transmission in WSNs introduces a distance challenge because attenuation limits the signal range. Moreover, the relationship between the transmission power and the received power is expressed using this formula:

$$P_r \propto \frac{P_t}{d^2}$$

Nodes in WSNs cannot transmit for larger distances because this transmission will consume much energy. Therefore, the best way is to divide the distance into shorter segments [9, 10, 11].
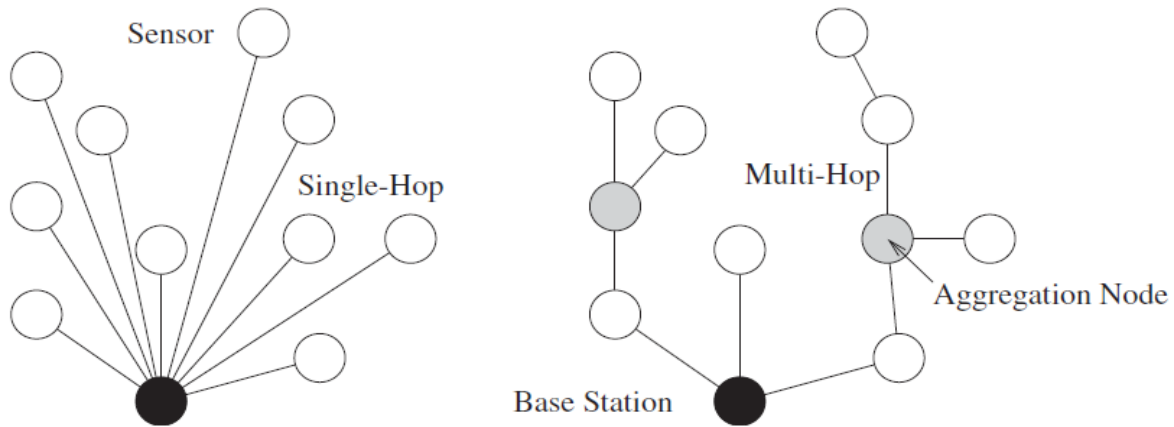
3.3.4 Design Constraints

The design goal in WSNs is to develop small, cheap, and energy efficient nodes. These design goals will limit the hardware capabilities of the nodes. For example, the nodes cannot have GPS systems that will force the designers to use alternative approaches to determining nodes' position. Also, the memory size will be modest, which will not allow storing huge routing tables, thus affecting the design of routing protocols [12].

3.4 Applications

3.4.1 Health Care

WSNs carry a great promise in improving health care services in a wide range of areas. Examples of that promise include real-time patient monitoring in hospitals, monitoring disaster areas, studying human behavior under chronic diseases, and improving the life of elderly persons through smart environments [14, 15].

3.4.2 Traffic Control

Congestion in ground transportation is one of the complex problems that are facing urban cities. WSNs can help by distributing sensing nodes that collect data about the traffic density, the directions and speeds of vehicles, and suggest alternative routes for drivers [6, 16].

3.4.3 Military Applications

Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, this attribute makes it suitable for military use. Also, the self-management capability in WSNs makes them resilient to the destruction of some nodes by enemies.

WSNs have been used in battlefield surveillance, damage assessment, biological and chemical attack detection and many other military applications [6, 14].

3.4.4 Environmental Applications

The autonomous feature of WSNs makes it a good solution for areas that are hard to reach by humans. Examples of these applications include: tracking animals, monitoring environmental conditions, detecting forest fires, and others [6, 15].

3.4.5 Home Applications

By installing WSN nodes in home appliances, they can form a network, and they can be connected to the internet so the owners can manage their home devices [13].

3.4.6 **Industrial Applications**

Wired sensor networks have long been used in industry. They primarily have been used for control and process automation. A huge disadvantage of wired networks is the high cost of the infrastructure as well as upgrading costs. WSNs can provide a good alternative for wired networks in industrial plants. Also, they could provide an alternative for human monitoring in preventive maintenance [15, 16].

4 Cross-Layer Design

The internet backbone protocols TCP/IP are based on the OSI layered architecture as shown in Figure 5. This layered architecture played a crucial role in the success and stability of the internet. The OSI model divides its tasks into layers from bottom to top. Each layer performs a precise task(s). A bottom layer serves a top layer by providing services to the above layer. The uniform interface between corresponding layers is achieved by encapsulating data coming from higher layers. This isolation in the layered design led to an easier application design that sits on the top of the hierarchy [13, 19].
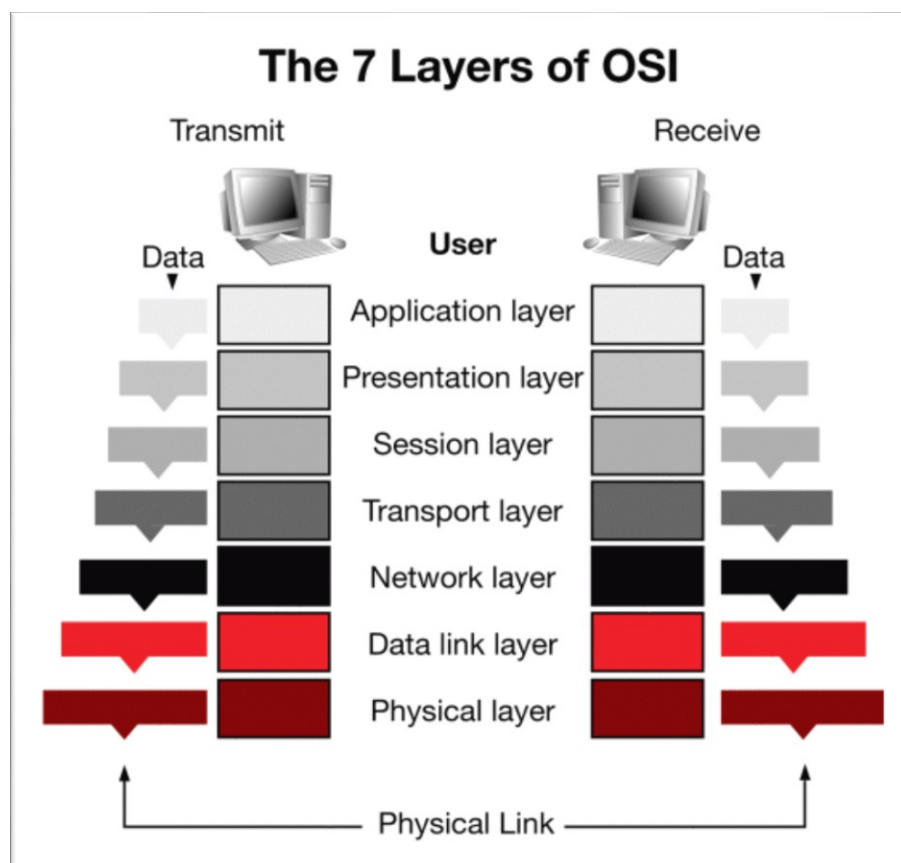
Figure 5: OSI Model (diagram taken from [66])

13

The rigid separation of layers in the OSI model causes isolation for each layer, which results in each layer having a limited view of the network. Combining this fact with the dynamic nature of WSNs will lead to the conclusion that the layered architecture is not the optimal choice for WSNs.

The dynamic nature of WSNs requires a fuller image of network status to be able to make the right decision. To obtain a more comprehensive picture of the network and adapt to changes layers need to share information. This is the essence of a cross-layer design [19].

4.1 General Motivations for Cross-Layer Design

Figure 6 shows an example of a cross-layer design framework with information exchange between layers. The link layer transmits the links' status to the MAC layer. The MAC layer, based on the information passed to it from the link layer, assigns time slots among nodes. The links' capacities are shared from the MAC layer to the network layer, which uses this information to make routing decisions that minimize congestion. Congestion control at the transport layer is optimized based on the traffic flow happening at the network layer. Packets categorization occurring in the application layer affects the resource allocation at all layers [14].
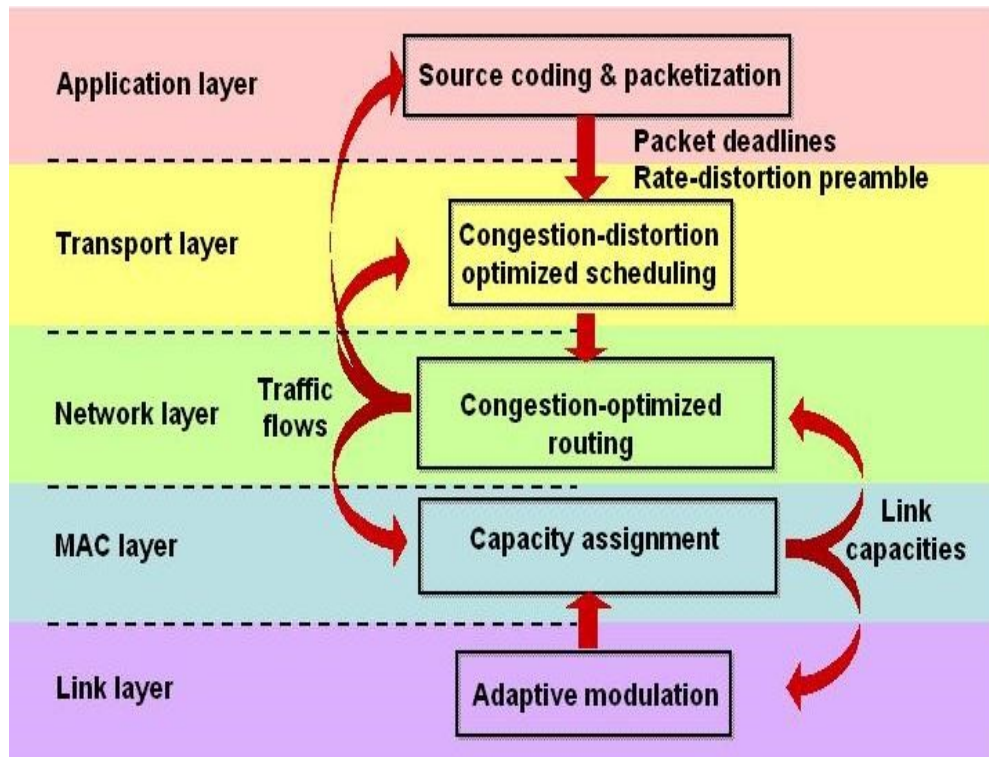
Figure 6: Cross-layer design framework (diagram taken from [67])

4.2 Proposals Involving Cross-Layer Design

There are two basic ways of violating the layered architecture:

- Creating new interfaces

- Merging adjacent layers

4.2.1 Creation of New Communication Interfaces

To be able to implement the cross-layer design, new communication interfaces need to be created. These interfaces, also called service access points (SAPs), are used to share information between layers [14]. Depending on the direction of information flow, this category can be further divided into three subcategories:

15

- Upward information flow

- Downward information flow

- Back and Forth Information Flow

4.2.1.1 *Upward Information Flow*
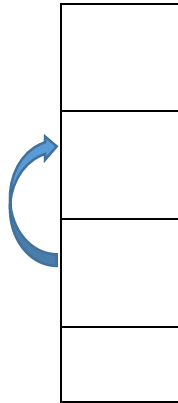
Figure 7: Upward information flow

Figure 7 shows the flow of information from a lower layer to an upper layer at run time. This new SAP from a lower layer to an upper layer results in the creation of upward information flow [14].

4.2.1.2 *Downward Information Flow*
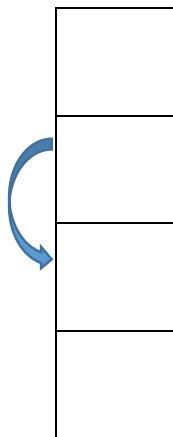
Figure 8: Downward information flow

16

Figure 8 shows the flow of information from a higher layer to a lower layer at run time. This new SAP from a higher layer to a lower layer results in the creation of a downward information flow [14].

4.2.1.3 *Back and Forth Information Flow*
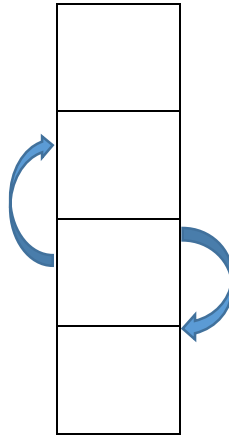
Figure 9: Back-and-forth information flow

Here, as shown in Figure 9, the flow of data is going in two directions. During the run time of the network, both the upper and lower layers will be exchanging data that will result in each layer having a broader view of the network status [14].

4.2.2 **Merging of Adjacent Layers**

Merging adjacent layers is another way of implementing the cross-layer design. Here, the services of two or more adjacent layers are merged to form a new layer called super layer. The super layer will have access to all the information the old layers had and performs the same services. This super layer will eliminate the need for sending information between layers [13].

4.3 Proposals for Implementing Cross-Layer Interactions

Depending on the cross-layer interaction, implementation can be divided into two categories:

- Direct communication between layers

- A shared database across layers

4.3.1 Direct Communication between Layers
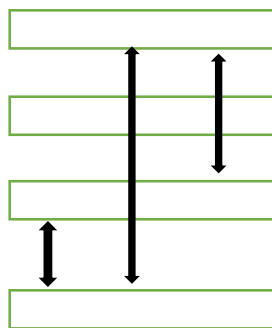
Figure 10: Direct communication between the different layers

As shown in Figure 10, in direct communication, the layers' exchange information at run time in a direct way by making data available to each other. This exchange is done by passing the needed information along with the packet header, either as part of the header or as extra information attached to the packet [13].

4.3.2 A Shared Database across Layers
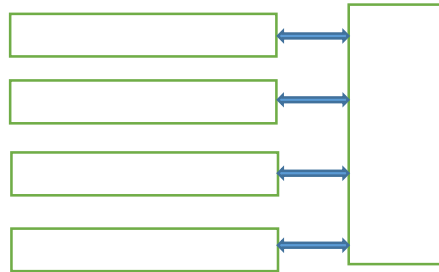
Figure 11: A shared database

As shown in Figure 11, a shared database could be used to exchange information between layers instead of direct communications. The shared database acts as a new layer that provides storage services to all other layers. Although this approach elevates the extra packets design, it adds the overhead of managing the database [13].

5 Network Layer

5.1 Routing Challenges

The nature of WSNs enforces different challenges than traditional wireless networks. This nature makes routing protocols designed for Ad-Hoc wireless networks not suitable for WSNs [6, 19].

5.1.1 Energy Consumption

Energy consumption has been the major challenge for designing routing protocols. Although the goal of traditional routing protocols is to deliver data through the shortest route, in WSNs, saving energy overcomes the importance of the shortest path [20].

5.1.2 Scalability

WSNs usually consist of a huge number of nodes. A large number of nodes makes it very expensive (resources wise) to store global network (like node ids) information at each node. Hence, fully distributed protocols, which operate on local information, need to be developed to enhance scalability [6, 20].

5.1.3 Addressing

The huge number of sensor nodes in a network makes assigning unique addresses for each node a difficult task to achieve. While unique addressing could be used locally between neighboring nodes, address-based routing protocols are not practical due to the large surcharges that come with unique addresses for each node [20, 21].

5.1.4 Robustness

Nodes in a WSN operate as routers since all routing decisions are made autonomously on the node level. The preferred routing approach in WSN is multi-hop, which results in a number of nodes being involved in delivering data packets. A single node failure will cause information loss. Nodes in a WSN are usually manufactured by using low-cost components, which make nodes prone to failure. This makes robustness an important design feature. Routing protocols should try to prevent single-point failures [21].

5.1.5 Topology

The deployment of WSN nodes is usually random. Consequently, individual nodes are unaware of their initial topology within the network. So routing protocols should provide topology-awareness such that the neighborhood of each node is discovered. Further network topology changes will occur during the lifetime of the network since nodes will be switching between ON and OFF modes to save energy. Power drainage and failure are other reasons for nodes leaving the network hence causing more topology changes [22].

5.1.6 Application Centric

WSNs are usually built and deployed to serve a specific application. The application serves as an important factor in designing routing protocols. To show this point, we will explain the different behavior of routing protocols in the case of monitoring applications compared with event-based applications.

In monitoring applications, nodes send sensed data to the base station in a periodic manner. So an efficient delivery could be maintained by using static paths.

In event-based applications, on the other hand, nodes stay in the sleep mode until an event occurs. When an event occurs routes, need to be established in an on-demand manner. So new routes are used with each event [6, 22]

5.2   Routing Protocols' Classification

5.2.1   Data-centric and Flat-Architecture Protocols

As mentioned earlier, the huge number of sensor nodes makes it very hard to assign IDs to nodes. Therefore, data-centric protocols treat all nodes equally, so the focus here is the data, not the nodes.

Data here is identified by attributes; the requesting data is done by the attributes of the phenomenon. A significant advantage in data-centric protocols is that there is no need for topology management, which reduces the management overhead. Examples of data-centric protocols are Flooding, Gossiping, SPIN, and Directed Diffusion [7, 20].

5.2.2   Hierarchical Protocols

In data-centric and flat-architecture protocols the majority of the information, generated by the sensors, has to go through nodes near the sink. As the nodes' density increases, nodes close to the sink suffer from an increased load. As a result, nodes closer to the sink die faster than nodes in other parts of the network. Therefore, flat-architecture

22

protocols cause unbalanced energy consumption through the network and cause a disconnection in the network.

The disadvantages of the flat-architecture protocols can be addressed by forming a hierarchical architecture, where the nodes form clusters and the local communication between cluster members are controlled by a cluster head. Sensor nodes form clusters where the cluster heads aggregate and fuse data to conserve energy. The cluster heads can also form another layer of clusters among themselves before reaching the sink. Examples of Hierarchical Protocols are LEACH, PEGASIS, TEEN, and APTEEN [20,23].
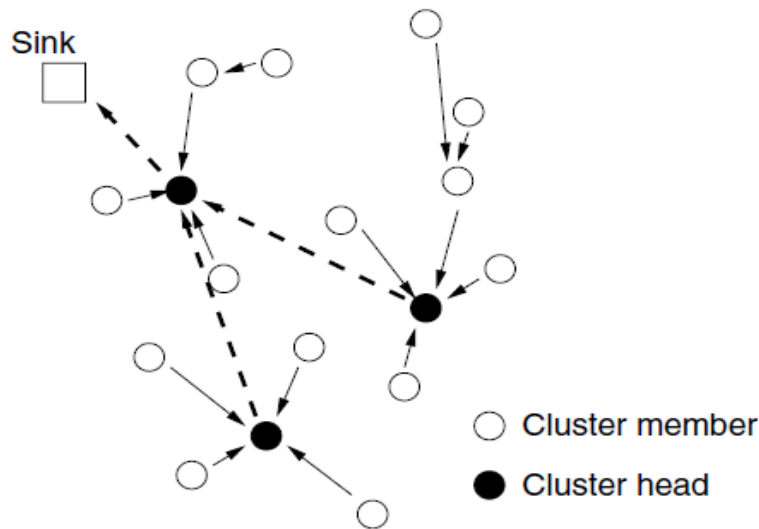
Figure 12: Hierarchical Protocols (diagram taken from [10, page 149])

5.2.3 Location Based Routing

Some applications of WSNs need location information. These applications associate the sensor reading with locations of the wireless sensor nodes. WSN nodes can be fitted

with GPS devices to provide the needed location information. However, GPS is not feasible for all types of WSNs, so some localization algorithms are being used to determine the node location.

In these cases, where the nodes' locations are available, it is logical to use this information for routing as well. Location-based protocols take advantage of the location information of each node to provide economical routing [20,24].

5.2.4 QoS (Quality of Service) Based Protocols

Many of the routing protocols explained above focus only on energy consumption in the network. Hence, route generation is performed to minimize energy consumption in the network. While energy consumption is one of the most important performance metrics in WSNs, it is not the only one.

There are applications where other performance metrics are more important than energy. In multimedia applications for example throughput and delay are more important. In these cases, the special QoS requirements must be met [7].

5.3 Directed Diffusion

Directed diffusion is another data-centric and data dissemination protocol. It is also application-aware in that data generated by sensor nodes is named by attribute-value pairs. The main idea of directed diffusion is that nodes request data by sending interests for named data. This interest dissemination sets up gradients within the network that are used to direct sensor data toward the recipient, and intermediate nodes along the data

paths can combine data from different sources to eliminate redundancy and reduce the number of transmissions [27,28].

Directed diffusion does not rely on globally valid node identifiers, but instead uses attribute-value pairs to describe a sensing task and to steer the routing process. For example, a description for a simple vehicle-tracking application could be:

type = vehicle // detect vehicle location

interval = 20 ms // send data every 20 ms

duration = 10 s // perform task for 10 s

rect = [-100,-100,200,200] // from sensors within rectangle

That is, a task description expresses a node's desire (or interest) to receive data matching the provided attributes. The data sent in response to such interests are also named in the same manner, which is, using attribute-value pairs [29].
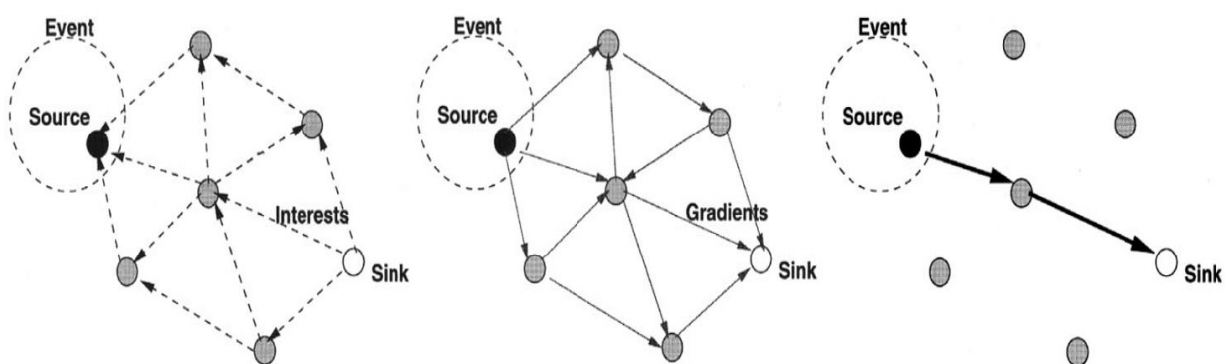
Figure 13: Directed Diffusion (diagram taken from [27, page 3])

Once an application has been described by using this naming approach, the interest must be diffused through the sensor network. This process is shown in Figure 13. A sink node periodically broadcasts an interest message to its neighbors, which continue to broadcast the message throughout the network. Each node establishes a gradient toward the sink node, where a gradient is a reply link toward the neighbor from which the interest was received. As a consequence, using interests and gradients, paths between event sources and sinks can be established. Once a source begins to transmit data, it can use multiple paths for transmission toward the sink. The sink can then reinforce one particular neighbor based on some data-driven local rule. For example, a sink could reinforce a neighbor from which the sink has received a previously unseen event. Toward this end, the sink resends the original interest message to the neighbor, which in turn reinforces one or more of its neighbors based on its own local rule.

Directed diffusion differs from SPIN in that queries (interests) are issued on demand by the sinks and not advertised by the sources as in SPIN. Based on the process of establishing gradients, all communication is neighbor-to-neighbor, removing the need for addressing schemes and allowing each node to perform aggregation and caching of sensor data, both of which can contribute to reducing energy consumption. Finally, directed diffusion is a query-based protocol, which may not be a good choice for certain sensor network applications (for example environmental monitoring applications), particularly where continuous data transmission is required [28].

5.4 Directed diffusion improvements

Several attempts have been made to improve directed diffusion after it had been initially developed. In this section, we do not claim a full review of directed diffusion improvements or modification but use a few examples to show the major trends in these modifications.

5.4.1 Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering

Directed diffusion distributes interests by performing network-wide broadcasts. Consequently, the overall performance of the protocol can be strongly improved by the efficiency of this elementary operation.

Performing network-wide broadcasts is done using the simple flooding algorithm where every node in the network forwards each new message to all of its neighbors. This algorithm is quite inefficient in wireless networks because it leads to a large number of unnecessary rebroadcasts between the neighboring nodes in the immediate area.

In this paper [30] the authors investigate the feasibility of this combination: executing directed diffusion on top of a sensor network with a topology constructed by passive clustering.

Clustering decreases the flooding cost by restricting the re-forwarding of the messages within the same cluster. Each node will only forward messages to the cluster head. For the approach to effective, the cost of forming and maintaining the clustered structure has to be lower than the energy savings from flooding messages
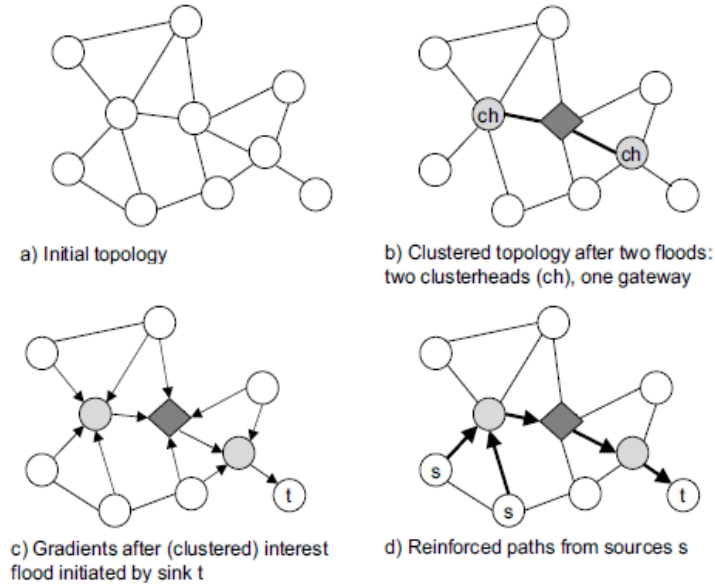
Figure 14: Establishing the clustered diffusion structure (diagram taken from [30, page 176])

In this algorithm, the construction of clusters is dynamic and is initiated by the first data message to be flooded. The advantage of this method over the traditional methods of clustering is avoiding long initial set-up periods.

5.4.2 The Study of Directed Diffusion Routing Protocol Based on Clustering for Wireless Sensor Network

Again, in this paper [31] the authors are seeking to improve directed diffusion by suppressing the energy waste caused by flooding interests. Based on the study of clustering mechanisms and algorithms, the authors are proposing a new directed diffusion routing protocol named DDBC (Directed Diffusion Based on Clustering).

To minimize the clustering overhead, DDBC adopts the passive clustering approach, the selection of cluster head takes a strategy named "first declaration wins".

In the "first declaration wins" mechanism, the first node to declare itself head will be the head of the cluster it is within.

The mechanism the cluster head node is selected makes trade-offs between the network robust and the energy-efficiency.
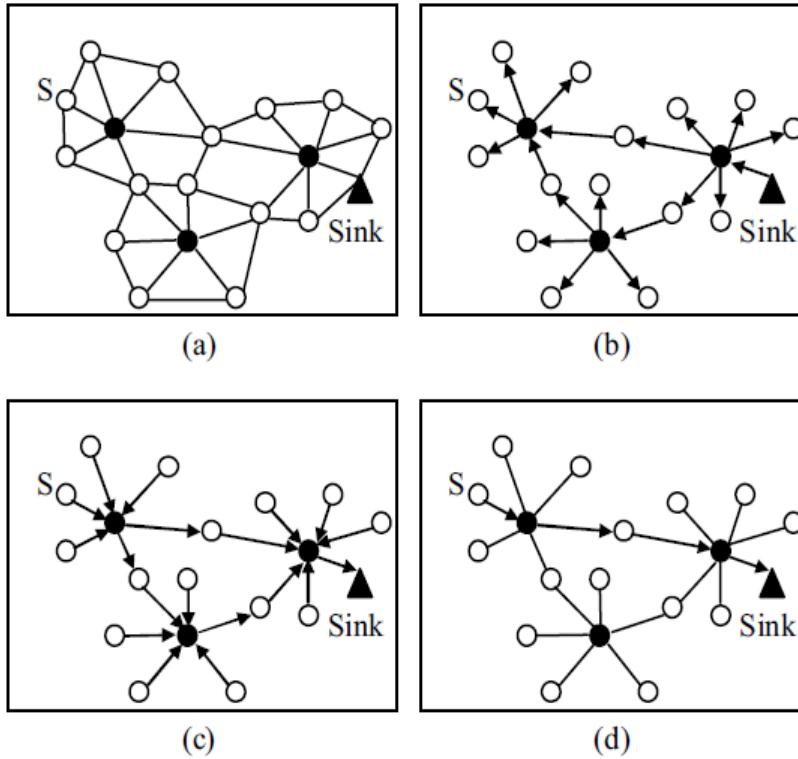
Figure 15: Establishing Clusters in DDBC (diagram taken from [31, page 5123])

Creating and maintaining routes in DDBC is achieved by adding some information about the cluster to the original packet structure.

5.4.3 An Energy-Efficient Diagonal-Based Directed Diffusion for Wireless Sensor Networks

Here [32] the authors are presenting a modified directed diffusion protocol by using a diagonal-based hexagonal-mesh scheme for a wireless sensor network. This algorithm is based on a fixed topology, namely the hexagonal-mesh. This choice was made based on the assumption that nodes have low-mobility in the network. As a result, this algorithm is not suitable for networks with mobile nodes.

To be able to build the hexagonal-mesh, sensor nodes must have a fully-functional Global Position System (GPS) receiver, to logically determine the coordinate position.
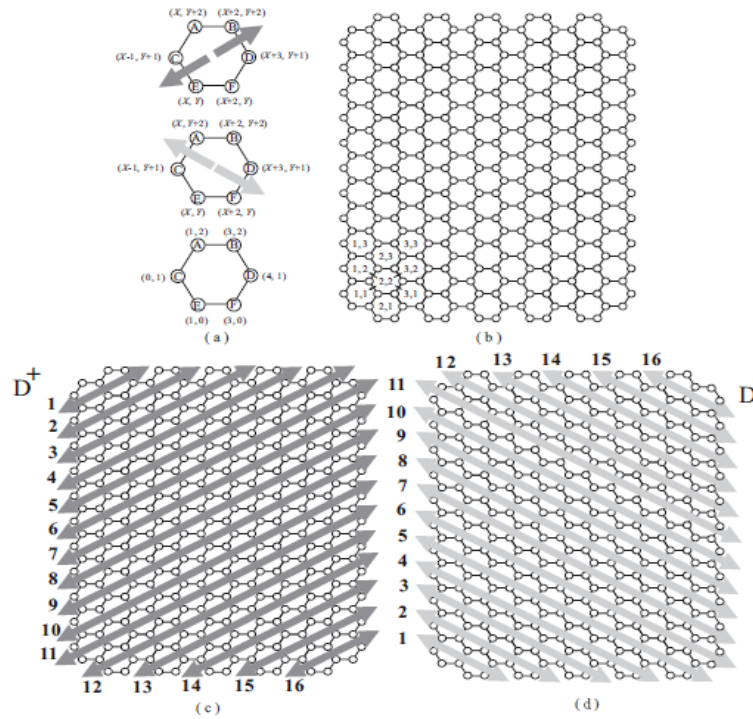
Figure 16: Example of a hexagonal mesh and diagonal paths (diagram taken from [32, page 2])

Diagonal paths are identified so that each sensor can directly connect to backbone path.

To avoid draining nodes on backbone paths, a periodic backbone-path-exchange scheme is presented for handling the per-node fairness problem.

5.4.4 A Secure and Energy-efficient Data Aggregation Protocol based on Directed Diffusion

Since flooding interests is one of the highest sources of energy waste, the authors in [33] are tackling this problem by introducing grid-based directed diffusion (GDD). Grids are constructed by self-organization of nodes using location information.
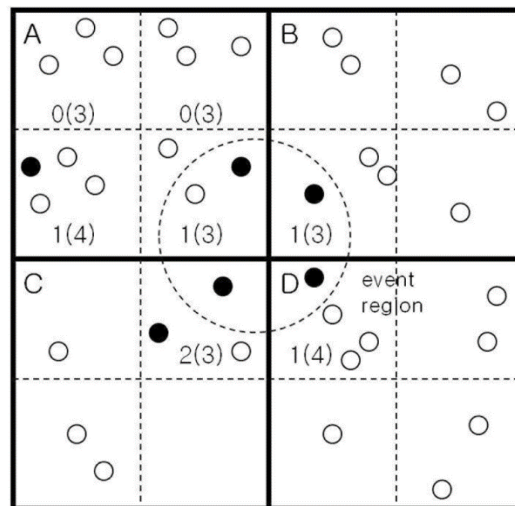
Figure 17: Grid-based Directed Diffusion (diagram taken from [33, page 2])

In GDD, the network area is first divided into fixed grids. In each grid, one grid head node is selected to transmit interest and sensing data. Broadcast overheads are reduced because the nodes receive interests and send data only through the grid head.

When the current grid head's battery power level falls below a predetermined threshold or serves for a predetermined period, it broadcasts a new election message within the grid. All the nodes then vote for a new grid head by using the ballot. This is done by replying to the new election message with its choice of candidate. The top pick from the trust table of its neighbors is selected as the grid's candidate.

5.4.5 LDDP: A Location-based Directed Diffusion Routing Protocol for Smart Home Sensor Network

In this paper [34], the authors are using the nodes' locations to propose a location-based directed diffusion routing protocol for smart home sensor network (LDDP).

The node's locations are exploited to partition the network based on rooms. This partitioning should result in fewer messages being forwarded since flooding to all the networks will be avoided.

(a) Interest diffusion stage (b) Gradient construction stage

(c) Path strengthening and data transmission stage
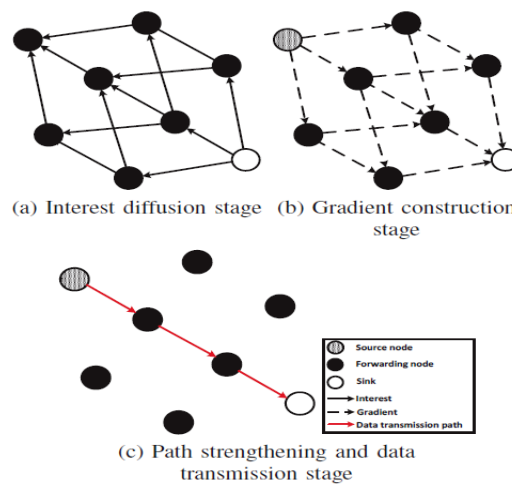
Figure 18: Directed diffusion routing process (diagram taken from [34, page 511])

LDDP divides the whole smart home sensor network into non-overlapping regions depending on the room where the sensor nodes are located. When a triggered event occurrs, the interest is diffused according to the room number. LDDP only allows nodes with the same room set to forward and receive interest mutually.

In directed diffusion, the optimal path is established hop-by-hop during the detecting process. While in LDDP, the sink node sends interest according to the room order. Sensing events are often associated with the room location and range.

5.4.6 EADD: Energy Aware Directed Diffusion for Wireless Sensor Networks

Directed diffusion prefers the fastest paths without taking into consideration the nodes' energy level. This decision mechanism will result in draining the nodes on the fastest path more quickly. It causes an unbalanced life cycle of the nodes. Consequently, we need to consider the available energy of the sensor nodes

In this paper [35], the authors propose EADD: Energy Aware Directed Diffusion for Wireless Sensor Networks. This algorithm makes forwarding decisions based on each node's available energy.

EADD gives preference to nodes with a higher energy level by assigning them a shorter response time compared with nodes with a lower energy level that will end up by choosing nodes with a higher energy level

If the nodes' energy distribution is as shown in Figure 20, to distinguish a more efficient path among the gradients is clear. The first path (S-80-60-D) Set Up delay time is shorter than that of the others.
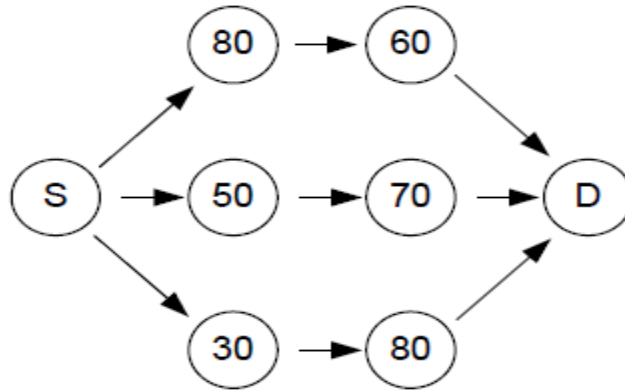
5.4.7 Energy-Aware Adaptive Directed Diffusion Algorithm of Wireless Sensor Networks

EAADD (Energy-Aware Adaptive Directed Diffusion) improves on the work done in [35] by considering the nodes' drainage history. This is done by bearing in mind the correlations of the nodes' available energy between adjacent rounds; then they use an adaptive algorithm to choose the next hop node that is more durable [36].

Since a certain node could forwards more packages than other nodes in the same round, first, the algorithm considers the correlations of the available nodes' energy between adjacent rounds; then an adaptive algorithm is used to choose the next hop node that is more durable.

This way could avoid the unbalanced energy consumption problem and save the energy of the nodes efficiently and prolong the network's lifetime.

5.4.8 Design of Gradient and Node Remaining Energy Constrained Directed Diffusion Routing for WSN

In this paper [37], the authors propose an energy-efficient routing algorithm for wireless sensor networks called Gradient and Node Remaining Energy Constrained Directed Diffusion Routing (GRE-DD)

In GRE-DD, they save energy consumption by setting a gradient diffusion depth (GDD), so the interests' propagation stops when GDD is reached.

GRE-DD helps to reduce the interest retransmission times at the interest propagation stage by setting a maximum value on the gradient diffusion depth, which will lead to less transmitted data.

At the same time, only nodes with an energy level higher than a set minimum are chosen as a gradient in the gradient setup phase.

GRE-DD help to increase the load balance and the average network lifetime by selecting nodes with a minimum remaining energy level. Using this threshold will increase the probability of selecting nodes with higher energy levels to do the transmission.

5.4.9 Analyzing Previous Work Done to Improve Directed Diffusion

In [30 - 32] all of these attempts focus on topology change to increase scalability and minimize the cost of flooding the interests. These approaches add topology management overhead and takes away the simplicity of directed diffusion.

In [33] and [34] the nodes' locations are being introduced to directed diffusion to partition the sensed area or to build geographic grids. The nodes' locations are being used to minimize interest flooding, direct path enforcing, and reduce redundant data transmission.

As in [30 - 32] the work done in [33] and [34] takes directed diffusion to another topology paradigm, which is location-based routing. Location-based routing comes with its own overhead; special devices or algorithms are used determine the node's location.

In [35] an Energy Aware Directed Diffusion Protocol (EADD) was proposed. It gives preference to nodes with a higher energy level by assigning them a shorter response time compared with nodes with a lower energy level that will end up by choosing nodes with a higher energy level.

In [36] EAADD (Energy-Aware Adaptive Directed Diffusion) improves on the work done in [35] by considering the nodes' drainage history. This is done by bearing in mind the correlations of the nodes' available energy between adjacent rounds; then they use an adaptive algorithm to choose the next hop node which is more durable.

In [37] they save energy consumption by setting a gradient diffusion depth (GRE-DD), so the interests' propagation stops when GRE-DD is reached to avoid total interest flooding. At the same time, only nodes with an energy level higher than a set minimum are chosen as a gradient in the gradient setup phase.

As can be seen in [35 - 37] the routing decisions are based on local knowledge.

6 Solar Energy

In this chapter, we present a short introduction on solar energy and its use in charging smaller devices like WSN nodes. Since WSN could be deployed in hard to reach environments, solar energy is a promising option to prolong WSN life span.

Understanding the mechanism of solar cells and their rechargeable cycles could be an important factor in improving routing decisions. A photovoltaic cell is a solar device that converts light into electrical energy through the photovoltaic reaction as shown in Figure 20. Most solar cells are made from silicon with high efficiency and low cost.

The most prominent advantages of solar cells that they do not require chemical reactions like batteries, nor they need mechanical parts like generators [40].
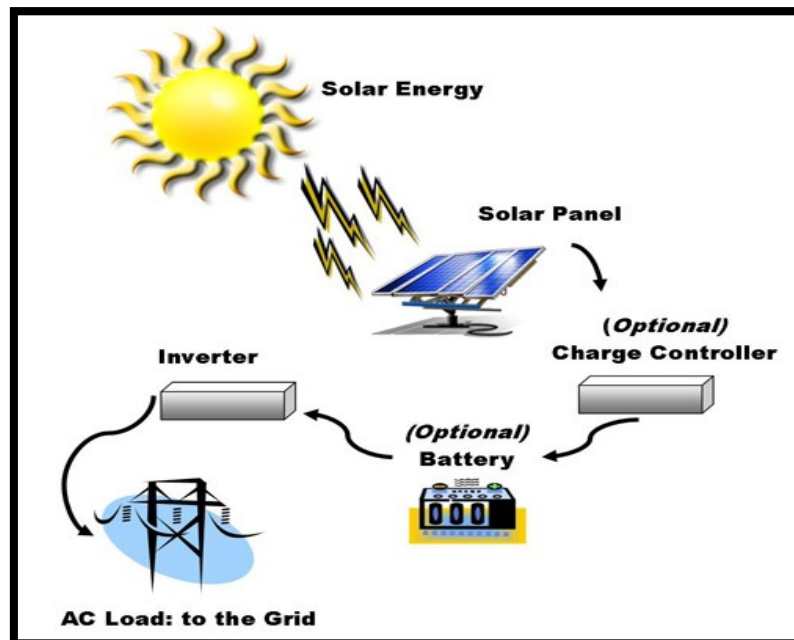
Figure 20: Solar Energy (diagram taken from [68])

37

As can be seen in Figure 21, manufacturing small devices in the size of sensor nodes became reality. A example of that reality is cell phones [41]. Additionally, the latest advances in solar cell technology allow a charging efficiency of 70%. As mentioned in the "Photonics Spectra" magazine in the May 2012 issue: "With a slightly more complex solar cell, it becomes possible to convert all colors of the light from the sun to electricity, and an efficiency of up to 70 percent is achievable" [42].

Figure 21: Portable Device Powered by Solar Energy (diagram taken from [69])

6.1 Power Harvesting

Due to the out-of-reach nature of wireless sensor networks, it is necessary to use batteries. However, the problem with using batteries is that they require regular replacement at usually hard to access places [43].

An emerging alternative is power harvesting. Power harvesting, also known as energy harvesting, is the process of capturing energy from natural resources [43]. The main disadvantage of power harvesting is the small energy it provides, but it WSNs this is not a problem considering the hardware capabilities of WSN nodes which result in low

power demand. The sources for power harvesting are many, which include solar, thermal, wind, and many others [43].

Considering the small size of WSN nodes, solar power is the optimal option. In the following section, we will focus on solar power harvesting.

In [44] the authors are providing an analysis of photovoltaic (PV) harvesting system for indoor low-power applications, which can be extended to the outdoor application as the case in WSNs.

Figure 22 show the main components of an energy harvesting system using a PV cell.
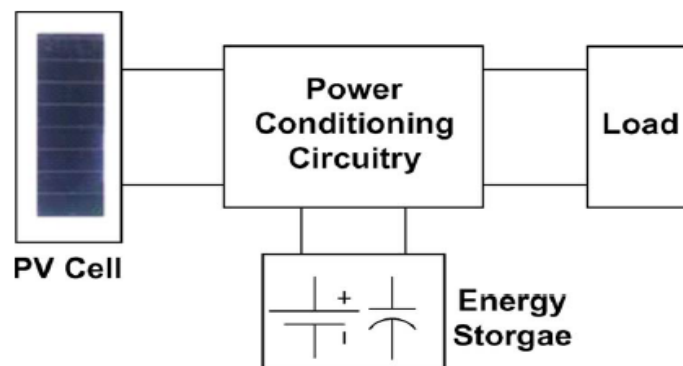
Figure 22: Configuration of the indoor energy harvesting system (diagram taken from [44, page 3])

In [44] the authors analyzed the system behavior under different loads with different illumination levels using various energy options. As a result, using a system with PV cells and the rechargeable battery is a good option to be used with low-power devices like WSN nodes.

7 Quality of Service (QoS)

QoS is the overall performance experienced by users when using a networking system. To be able to measure the quality of service, several characteristics are usually measured, such as error rates, bit rate, throughput, delay, availability, and more [45].

QoS can be categorized into two main architectures, integrated services, and differentiated services. Figure 24 gives an overall view of the difference between integrated services and differentiated services [45].
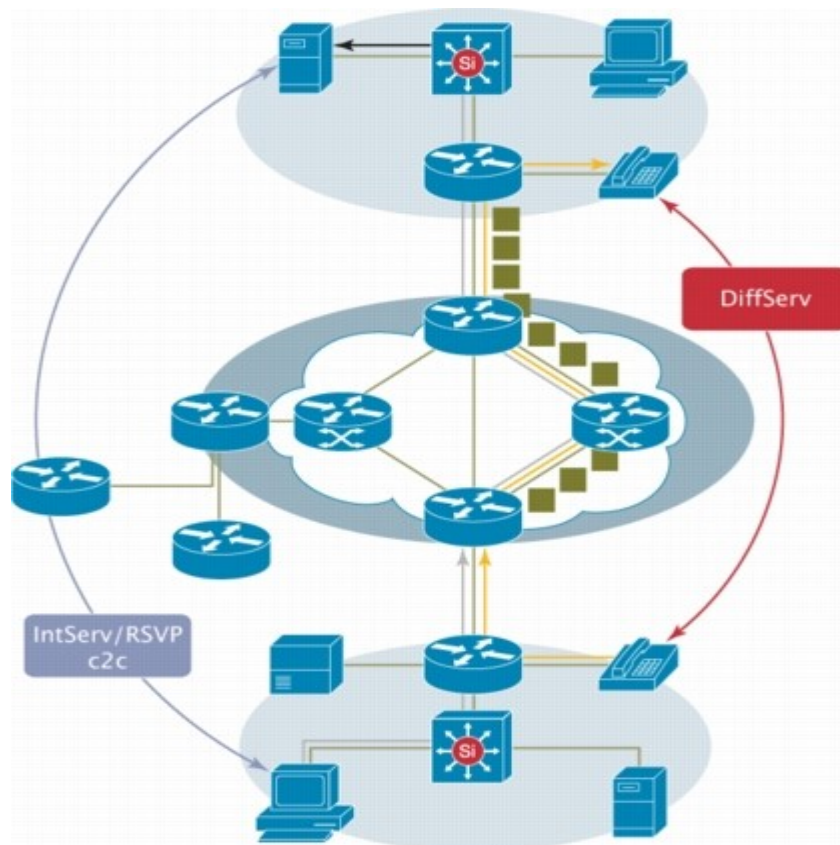
Figure 23: Integrated services Vs. Differentiated services (diagram taken from [70])

7.1   Integrated services

Integrated services or IntServ is a fine-grained QoS system, where elements are reserved all through the service path to guarantee QoS on the network.

Before a request is served all the routers between the sender and the receiver must decide if they can fulfill the requirements needed for the request. If they cannot, then the request is rejected.

If the reservation goes through, then all the request resources are dedicated to serving this request [45].

7.2   Differentiated services

Differentiated services or DiffServ is a coarse-grained QoS system. DiffServ provides QoS by classifying network traffic and providing different service according to the traffic class.

Modern networks carry different classes of data, including voice, video, and text. Each class has its own QoS needs.

DiffServ classifies packets and then routers on the path from the sender to the receiver to implement a per-hop behavior that manages each traffic class differently preferring higher-priority packets [45].

7.3 QoS in Wireless Sensor Networks

Besides resource limitations in WSN nodes, WSNs are usually deployed in unattended and harsh environments implementing crucial applications. These factors emphasize the importance of QoS in WSNs [46].

Considering the nature of WSNs, the IntServ approach is not applicable to WSNs. WSN nodes do not have sufficient resources to establish end-to-end connections and manage the information needed for these connections.

Although the QoS requirements differ in WSNs according to the network application, WSN nodes work collectively to achieve the application goals that make the DiffServ a better option to use with WSNs [46].

The most important points to be considered when designing a QoS system for WSNs are as follows:

- QoS must be integrated into all the network layers
- QoS parameters must be decided based on the WSN application
- Resources constraints must be considered

7.3.1 QoS at the Network Layer

QoS in WSNs at the network layer encompasses end-to-end reliability, which is an important requirement in WSN applications. Designing an efficient routing protocol, which uses resources efficiently, is very crucial to empowering QoS in the network.

Implementing QoS at the network layer level without the help from other layers will result in a weak design. Solutions must be cross-layer optimized [45, 46].

One aspect that network layer can improve on QoS is by minimizing the number of control messages sent [46].

Routing protocols that ensure shorter paths can increase QoS in the network. As shown in [47] each node increase on the path between the source and the destination increases the average packet loss ratio by approximately 5-10 %.

Network layers can increase reliability, which is an important QoS aspect, by enforcing multipath routing. WSNs usually have high node density, so the possibility of having more than one path between the source and the destination is high. According to [48 - 50] the delivery ration on a 14-hop path can be increased from 50% to 75% if there is a second disjoint path.

RAP [51], and SPEED [52] all use geographic forwarding (GF), in which nodes forward packets to their one-hop neighbor that is closer to the sink. This will ensure faster delivery and shorter paths. Multipath Multi-SPEED (MMSPEED) [48] also uses GF but adds the feature of multipath.

JiTS (Just-in-Time Scheduling) [53] is a network layer protocol for soft real-time packet delivery. JiTS orders packets in a forwarding queue based on their transmission time. Transmission time is calculated by multiplying the average one-hop delay by the number of hops. When a packet's transmission time is reached, it is dequeued from the queue head.

8 Proposed Protocol

This section is dedicated to describing the design of our proposed WSN routing protocol.

8.1 General Design

Our proposed WSN routing protocol can be classified as Data-Centric and is based on the famous Directed Diffusion WSN routing protocol.

Directed Diffusion is a well-known WSN routing protocol that uses on-demand queries (interests) to request data from nodes. As on-demand query protocol, Directed Diffusion does not need to keep global addresses or global network topology, which makes it a scalable protocol.

Directed Diffusion is an energy efficient protocol, and it achieves that by:

i.   Using attribute-based naming to query data by flooding these queries in the network; this saves energy since the attribute-based naming queries are much smaller than the data itself

ii.   Using gradients to direct data back to the sink instead of flooding the data in the network to reach the sink

However, Directed Diffusion has some drawbacks:

i.   It does not take into its consideration the energy level of the nodes when enforcing a certain path.

ii.   It does not implement smart decision techniques that will compare between available paths and choose the best.

iii. It does not provide any QoS features.

iv.   It does not avoid critical nodes (nodes with a very low level of energy)

v.   It does not distinguish between traffic types. Some data packets could be a higher priority than others.

vi.   Routing decisions are based on local information only.

These are some of the drawbacks that we will be trying to address in our protocol.

8.2   Similarities to Directed Diffusion

1. Both are data-centric protocols.

2. Both are query-based protocols, using flooding to send out queries.

3. Both use attribute-based naming.

4. Both use interests to query data and gradients to send them back to the source.

8.3   Differences from Directed Diffusion

1. Handling rechargeable nodes.

2. Considering the energy level of nodes when deciding return paths.

3. Collecting information about the status of available path then using smart decision techniques to help choose the best-enforced path(s).

4. Considering critical nodes

5. Implementing cross-layer optimization.

6. Classifying traffic by assigning a priority flag for more important data.

A major design change that was added to directed diffusion is of a whole new stage called the information collection stage. This stage collects the needed data about the available paths to compare them and choose the best among them.

In our case, we collect Hop Count, Total Energy, and Lowest Energy. However, these parameters could be different. The designer of the application can decide what data are needed and collect them through this stage. This choice opens the possibilities to a wide range of implementations.

In WSNs, routing is based on local information among neighboring nodes. Routing decisions are made locally; each node will select the next hop without any clue about the other nodes on the path. Although a full knowledge about the network yields better routing, that is not feasible in WSNs due to memory limitation, and due to the high traffic needed to collect data about all the nodes in the network [11].

In our work, we take a middle way between full network knowledge (holistic) and local knowledge. Aware diffusion purses a semi-holistic approach. Instead of collecting data about the whole network we only collect the needed data about the potential paths between the source (sensing node) and the destination (sink node). This means at the moment of choosing the next hop the node will have information about the potential paths leading from the node making the routing decision to the destination.

8.4 Detailed Design

In this section, we will describe the packets structure used in the proposed protocol. Mainly they are the query packet and the data packet besides other types of packets. Also, we will describe the details of route discovery and enforcement.

8.4.1 Packets Structure

In this protocol, we have four types of packets, the query packet that is flooded through the network holding the attributes of the requested data, the data packet, which is the nodes' reply to the query packet holding the data back to the sink, and the enforcing packet that is used to enforce the chosen path.

8.4.1.1 Query Packet (interest)

Interest ID	Source ID	Destination ID	Time to Live	Requested Data Attributes	Type

Figure 24: Query Packet

Interest ID: Each interest generated by the base station has a unique ID

Source ID: The local ID of the sending node

Destination ID: The local ID of the receiving node

Time to Live: Time in seconds which determines the life span of an interest

Requested Data Attributes: using the application attribute-based naming scheme to describe the requested data

47

Type: Specifies data type; 0: normal, 1: critical, 2: urgent

Interest ID	Source ID	Destination ID	Hop Count	Total Energy	Lowest Energy

Figure 25: Query Response Packet

Interest ID: Each interest generated by the base station has a unique ID

Source ID: The local ID of the sending node

Destination ID: The local ID of the receiving node

Hop Count: The total number of nodes on the path so far

Total Energy: The total of accumulated energy level of all the nodes on the path

Lowest Energy: The energy level of the node with the lowest energy level on the path

8.4.1.3 *Data Packet*

Interest ID	Source ID	Destination ID	Data

Figure 26: Data Packet

Interest ID: Each interest generated by the base station has a unique ID

Source ID: The local ID of the sending node

Destination ID: The local ID of the receiving node

Data: The data being sent back to the source

48

Interest ID	Source ID	Destination ID

Figure 27: Enforcing Packet

Interest ID: Each interest generated by the base station has a unique ID

Source ID: The local ID of the sending node

Destination ID: The local ID of the receiving node

8.4.2 Interests

An interest message is a query that determines what the operator wants. An interest contains a description of the task needed. Any task should be supported by the WSN. Typically, these tasks are collecting data about a physical phenomenon happening in the environment.

Interests are classified into the following categories:

- Normal data
- Critical data
- Urgent data

If critical data is being requested, then more than one path can be reinforced to ensure delivery. In the case of urgent data, packets holding urgent data will go through urgent data queue, which has a higher priority than critical or normal packets queues.

8.4.3 Naming Scheme

As in directed diffusion, task descriptions are named by a list of attribute-value pairs that describe a task. In general, each attribute has a value range associated with it. So the first step in designing the protocol is choosing a naming scheme for the tasks supported by the sensor network.

The attributes and their values are application dependent. So according to the application being run at the application layer, the network will be supporting different tasks with different attributes and values. However, in general, since our protocol is query-based protocol, applications will be requesting certain data. To distinguish different data types, there should be an attribute called type. Another common attribute is time to live. This attribute will be used to know when an interest expires.

8.4.4 Proposal Flow Chart

Figure 29 shows the process step by step. In the following section, we will explain each step in detail.
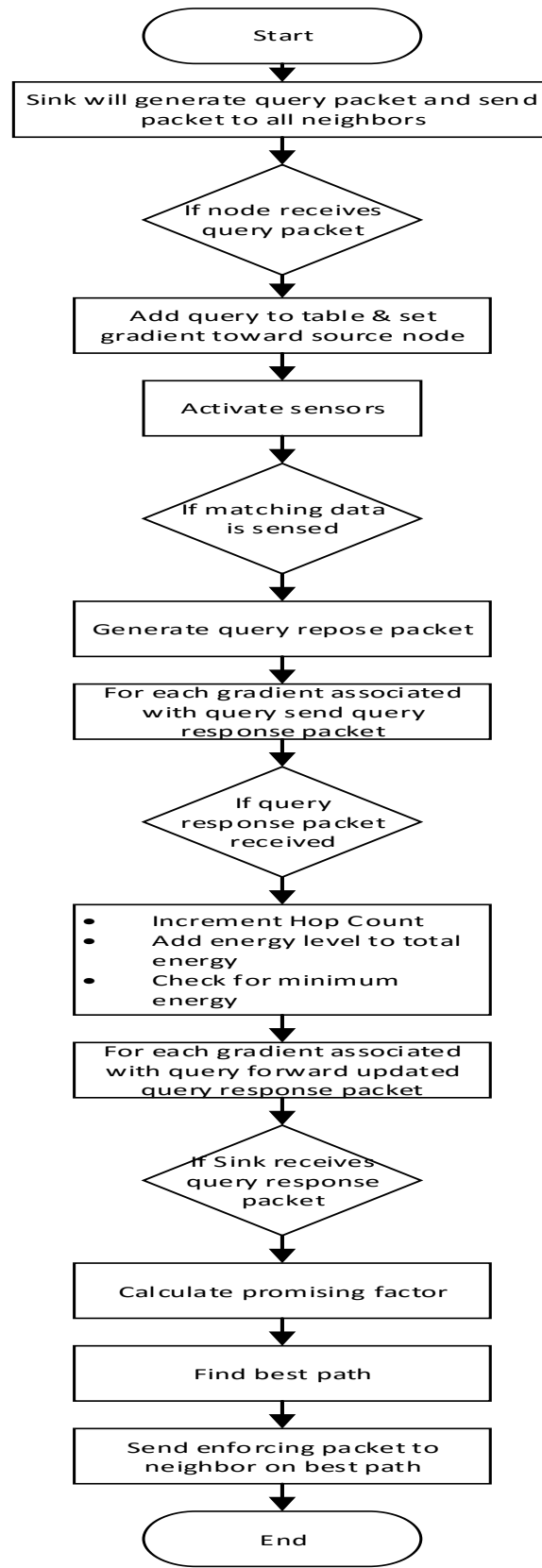
Figure 28: Proposed Protocol Flow Chart

51

8.4.5 Interests Propagation

Interests are flooded through the sensor network. For each active task, the sink will broadcast an interest message to all its neighbors. Each node that receives the interest message will also broadcast it to all its neighbors.

Every node maintains an interest cache. Each item in the cache corresponds to a distinct interest. Two interests are distinguished by the ID field.

Interest entries in the cache do not contain information about the sink, but just about the immediately previous hop. Also, identical interests are aggregated into a single entry.

When a node receives an interest, it checks to see if the interest exists in the cache. If no matching entry exists the node creates an interest entry. This entry has a single gradient (a gradient specifies a direction in which to send events) toward the neighbor from which the interest was received. If an interest entry does exists, but no gradient for the sender of the interest, the node adds a gradient with the specified value. Finally, if both an entry and a gradient do exist, the node simply updates the attribute fields if they are different.

When an interest entry has expired, the interest entry is removed from the interests' cache.

Not all received interests are resent. A node may suppress a received interest if it recently resent a matching interest.

Interest ID	Time to live	List of Gradients	List of Enforced Neighbors	Query Response Packet	Priority
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	

Table 1: Interests Table at Each Node

Figure 29 shows the pseudo code for the interests' propagation stage

Proposed Protocol – Interest Propagation Stage

Base station generates a query packet according to user request

Base station transmits query packet to all neighboring nodes

if node receives a query:

if query is not a duplicate :

        add query to table

        set a gradient toward source node

        activate sensors

        forward query packet to neighboring nodes

Figure 29: Interests Propagation Stage Algorithm

8.4.6 Data Collection Stage

This stage is a major addition to directed diffusion. In this stage, all the needed data to compare between paths are collected.

When a node has at least one active interest, the node will switch on its sensors and start sensing for the requested data. If the sensing node senses data that matches the requested data by the interest, it will generate a Query Response Packet and send a copy of it to all the gradients associated with the interest.

Forwarding nodes, on the other hand, could receive the same Query Response Packet from multiple neighbors, but it will only forward one of them.

Each Query Response Packet received by the base station will hold an information summary (Hop Count, Total Energy, and Lowest Energy) about the path it took. Based on that information the base station will choose the best path according to the criteria determined by the application designer.

Figure 30 shows the pseudo code for the data collection stage.

Proposed Protocol – Data Collection Stage

if node senses data matching any active query:

    if no enforced neighbor is available for this query:

        generate Query Response Packet

        for each gradient associated with query:

            send Query Response Packet

    else:

        generate Data Packet

        send Data Packet to enforced neighboring node

if nodes receives Query Response Packet:

    Increment Hop Count field

    add energy level to total energy level field

    if node's energy level if less than Lowest Energy field:

        set Lowest Energy field to node's energy level

    for each gradient associated with query:

        forward updated Query Response Packet

Figure 30: Data Collection Stage Algorithm

55

8.4.7 Reinforcing Paths

When the base station starts receiving Query Response Packets in the reply to an interest that was propagated earlier, it will receive the packets through multiple paths because the source node will send the Query Response Packet to all the nodes from which it received the interest propagation packet.

Using the information summary contained in the Query Response Packet, the base station can choose the best path(s) and reinforce the chosen path by using enforcing packets. The number of reinforced paths is decided based on the importance of the data being requested.
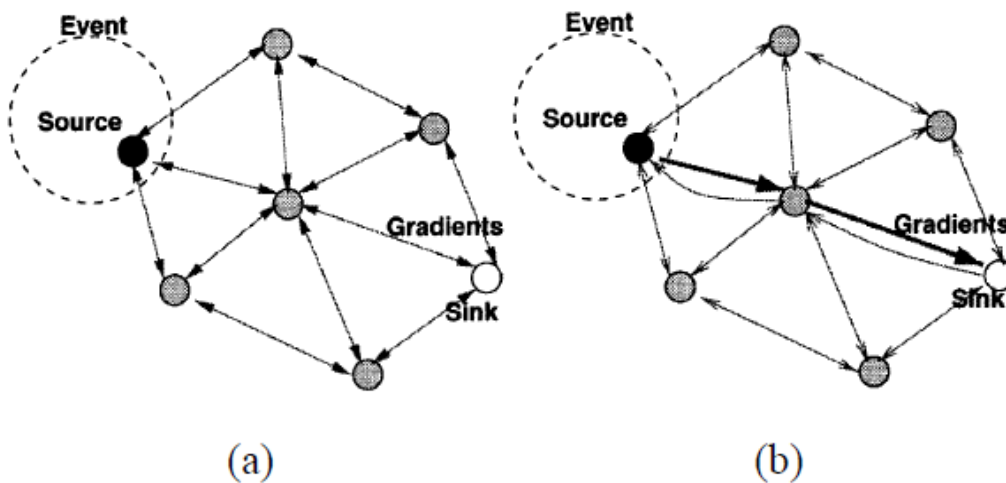
Figure 31: Enforcing Paths (a) Gradient establishment (b) Sending Enforcing Packets (diagram taken from [27, page 3])

After deciding the best path(s), the base station will send an enforcing packet to the neighboring node(s) that forwarded the Query Response Packet containing the best path(s) information. In turn, each forwarding node on the path of the enforcing packet will forward the enforcing packet to the node from which received the Query Response

56

Packet. The forwarding node will know to which node to forward the enforcing packet from the interests table (table 1) because, as shown in Table 1, each node will cache the Query Response Packet forwarded in response to the interest currently being worked on. Moreover, the cached Query Response Packet contains the source node ID as shown in Figure 26.

Figure 32 shows the pseudo code for the reinforcing paths stage.

Proposed Protocol – Reinforcing Paths Stage

if base station receives Query Response Packets:

    calculate promising factor for all available paths using the following equation

        $PF\ s{\rightarrow}d=((TE)*(LE))/((HC))$

    find path with the highest promising factor

    send an Enforcing Packet to neighboring node

if node on path receives an Enforcing Packet:

    set the gradient pointing to sending node as enforced neighbor

    calculate promising factor for all available paths using the following equation

        $PF\ s{\rightarrow}d=((TE)*(LE))/((HC))$

    find path with the highest promising factor

    send an Enforcing Packet to neighboring node

Figure 32: Reinforcing Paths Stage Algorithm

8.4.8 Data Propagation

After the reinforcing phase is completed, the source nodes know which neighboring nodes to use to forward the data packets.

Every time the source nodes sense a matching data to interest requested data they will generate a data packet and forward the data packet towards the base station using the enforced nodes listed in Table 1.

Every node along the path will do the same thing and forward the data packet through the list of enforced nodes until the data packet reaches the base station.

Each node will have three buffers: one for normal data, another for critical data, and the last one for urgent data.

This process will continue until the time in the to-live field associated with the interest becomes zero. Then this interest will be removed from the table of interests in the source node, and no more data packets will be generated in response to this interest.

Figure 33 shows the pseudo code for the data propagation stage.

Proposed Protocol – Data Propagation

if node receives a Data Packet:

      forward Data Packet to enforced neighboring node associated with this query

---
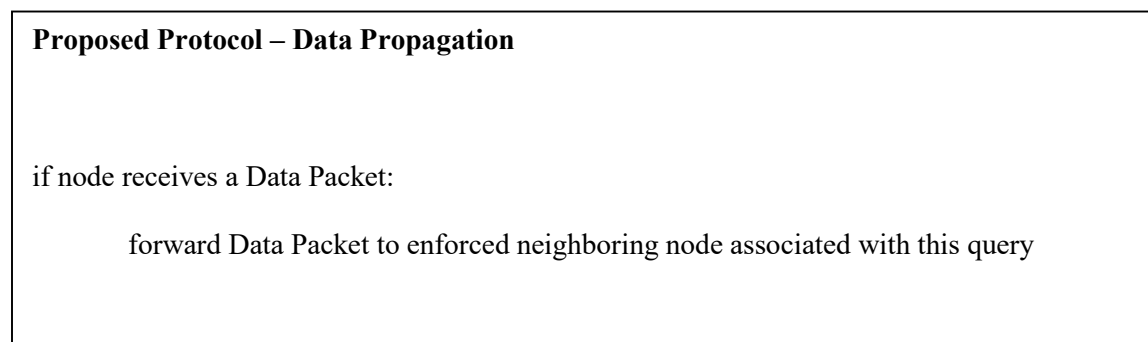
Figure 33: Data Propagation Algorithm

8.4.9 Quality of Service (OoS) Aspects

Our algorithm can provide a certain number of QoS aspects:

- Reliability: by reinforcing more than one path and by preferring healthier nodes the reliability can be increased.

- Differential Services: by classifying data packets into critical, urgent, and normal and dedicating different resources to each class, urgent packets can be delivered faster.

- Speed: by preferring shorter paths over longer ones the speed can be increased

8.4.10 Cross-layer implementation

As described in the Cross-Layer design section, the dynamic nature of wireless sensor networks motivates the violation of the layered architecture.

In our design we violated the layered architecture by creating new communication interfaces between layers in two ways:

1. **Upward information flow:** Here energy levels from the physical layer were communicated to the routing layer to help in making routing decisions.

2. **Downward information flow:** Here the data classes (normal, critical, and urgent) were communicated from the application layer to the routing layer to help in making routing decisions.

8.4.11 Choosing the Best Path

Based on the fact that transmission is the main source of energy depletion in WSNs, in our work, we decided to use non-incremental learning to avoid the continuous data collection required in incremental learning. The continuous data collection will result in more data collection packets being sent on a regular basis, and as a result, more energy consumption that outweighs the benefit of improving the learned concept [24].

In our work, the learning data are collected every time a new query is initiated, and the same learning data are used for the duration of the query lifetime.

Choosing the best path from source (s) to destination (d) is done by calculating the cost function for each path using the following formula:

Cost function:

$$\text{PF } s \rightarrow d = \frac{(TE) * (LE)}{(HC)}$$

<div align="center">Equation 1: Cost function</div>

Where:

TE: Path Total Energy Ratio

LE: Path Lowest Energy Level Ratio

HC: Path Hop Count Ratio

The components of equation 1 are described in the following sections:

8.4.11.1 *Path Total Energy Ratio*

$$\text{TE for path } s \rightarrow d = \frac{Path\ s \rightarrow d\ Total\ Energy}{\sum_{All\ Paths\ s \rightarrow d} Total\ Energy} * 100$$

Equation 2: Path Total Energy Ratio

Where:

$$Path\ s \rightarrow d\ Total\ Energy = \sum_{All\ node\ on\ Path\ s \rightarrow d} Node\ Energy\ Level$$

Equation 3: Total Energy for Path s → d

8.4.11.2 *Path Lowest Energy Level Ratio*

$$\text{LE for Path } s \rightarrow d = \frac{Path\ s \rightarrow d\ Lowest\ Energy\ Level}{\sum_{All\ Paths\ s \rightarrow d} Lowest\ Energy\ Level} * 100$$

Equation 4: Path Lowest Energy Ratio

Where:

$$Path\ s \rightarrow d\ Lowest\ Energy\ Level = \min_{All\ nodes\ on\ Path\ s \rightarrow d} (Node\ Energy\ Level)$$

Equation 5: Lowest Energy Level for Path s → d

8.4.11.3 *Path Hop Count Ratio*

$$\text{HC for Path } s \rightarrow d = \frac{Path\ s \rightarrow d\ Hop\ Count}{\sum_{All\ Paths\ s \rightarrow d} Hop\ Count} * 100$$

Equation 6: Path Hop Count Ratio

Where Hop Count is the number of nodes on path s → d

In this section, we will demonstrate three examples that show how the best path, according to our criteria, is chosen in three different scenarios.

In all the examples the source node is assumed to be node A and the sink is Node F.

8.4.12.1 *Example 1: Prefer Shortest Path*

In this example, the energy levels of all the nodes are close to each other so preference will be given to the shortest path available.

Figure 34: Sample Network 1

In Table 2 we will show the values of the collected data (total energy, lowest energy, and hop count) that was collected by the Query Response Packet about the four possible paths.

Path	Total Energy	Lowest Energy	Hop Count
ABF	23	11	2
ABCDF	45	11	4
ACDF	34	10	3
ACBF	33	10	3
Sum	135	42	12

Table 2: Data Collected For Sample Example 1

In Table 3 we show the calculations for Total Energy Ratio, Lowest Energy Level Ratio, and Hop Count Ratio for each path using equations 2, 4, and 6 and at the end the cost function for each path using equation 1.

Path	Total Energy Ratio	Lowest Energy Ratio	Hop Count Ratio	Cost function
ABF	(23/135)% = 17.03	(11/42)% = 26.19	(2/12)% =16.66	170.3 x 26.19 / 16.66 = 26.77
ABCDF	(45/135)% = 33.33	(11/42)% = 26.19	(4/12)% =33.33	33.33 x 26.19 / 33.33 = 26.19
ACDF	(34/135)% = 25.18	(10/42)% = 23.80	(3/12)% = 25	25.18 x 23.80 / 25 = 23.97
ACBF	(33/135)% = 24.44	(10/42)% = 23.80	(3/12)% = 25	24.44 x 23.80 / 25 = 23.26

Table 3: Cost Function Values for Sample Example 1

According to the cost function values that were shown in Table 3, path ABF is the chosen one because it is the shortest path.

In this example, the energy levels of all the nodes are not close to each other where node C has a high level of energy compared with the rest of the nodes so preference will be given to paths containing node C, but since node C is part of multiple paths, the shortest among them will be chosen.
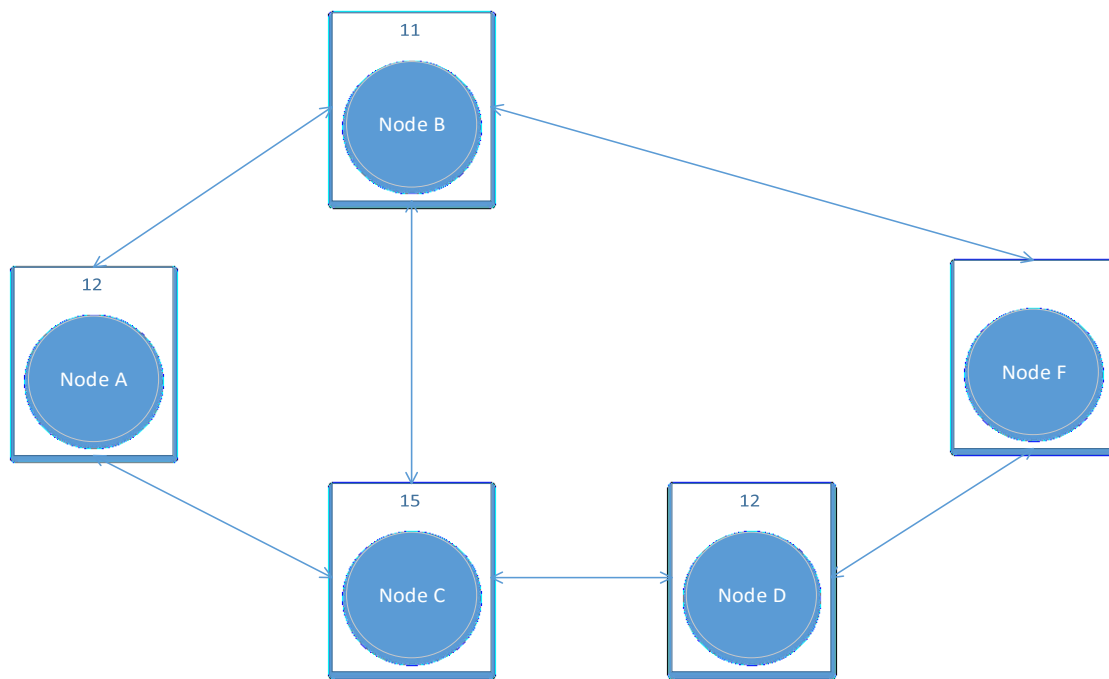
Figure 35: Sample Network 2

In Table 4 we will show the values of the collected data (Total energy, Lowest Energy, and Hop Count) that was collected by the Query Response Packet about the four possible paths.

Path	Total Energy	Lowest Energy	Hop Count
ABF	23	11	2
ABCDF	50	11	4
ACDF	39	12	3
ACBF	38	11	3
Sum	150	45	12

Table 4: Data Collected For Sample Example 2

In Table 5 we show the Total Energy Ratio, Lowest Energy Level Ratio, and Hop Count Ratio for each path using equations 2, 4, and 6 and at the end the cost function for each path using equation 1.

Path	Total Energy Ratio	Lowest Energy Ratio	Hop Count Ratio	Cost function
ABF	(23/150)% = 15.33	(11/45)% = 24.44	(2/12)% =16.66	15.33 x 24.44 / 16.66 = 22.48
ABCDF	(50/150)% = 33.33	(11/45)% = 24.44	(4/12)% =33.33	33.33 x 24.44 / 33.33 = 24.44
ACDF	(39/150)% = 26	(12/45)% = 26.66	(3/12)% = 25	26 x 26.66 / 25 = 27.72
ACBF	(38/150)% = 25.33	(11/45)% = 24.44	(3/12)% = 25	25.33 x 24.44 / 25 = 24.76

Table 5: Cost Function Values for Sample Example 2

According to the calculated cost function values that were shown in Table 5, path ACDF is the chosen one because it contains node C and shorter than other paths containing node C.

65

In this example, the energy levels of all the nodes are not close to each other where node D has a very low level of energy compared with the rest of the nodes (so it is a critical node) so preference will be given to paths not containing node D.
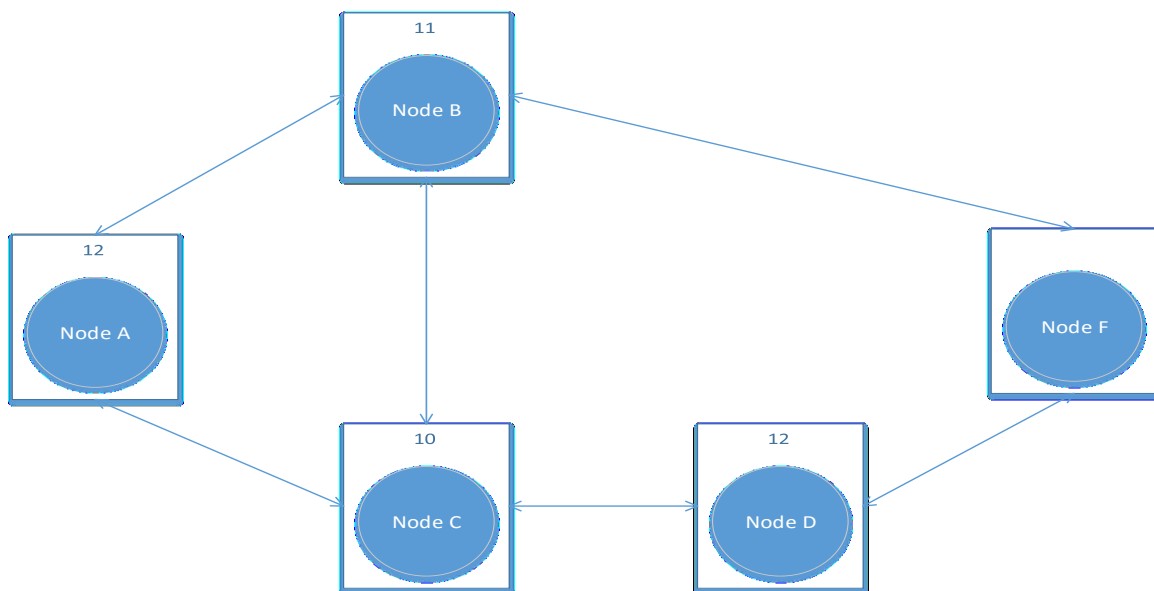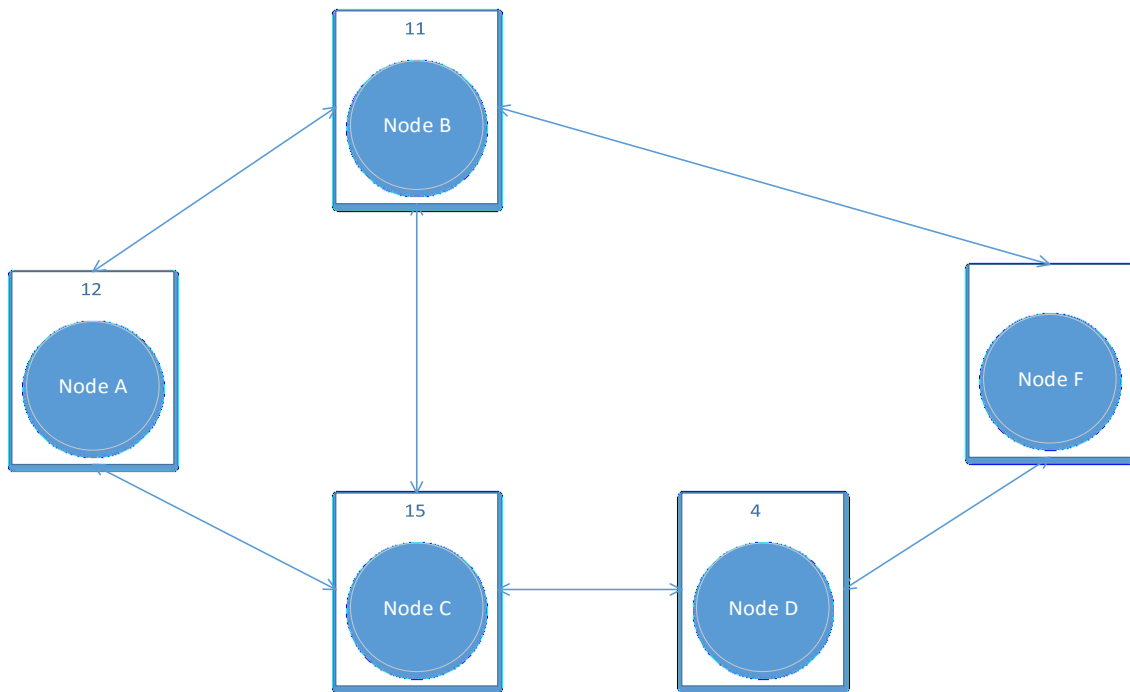

Figure 36: Sample Network 3

In Table 6 we will show the values of the collected data (Total energy, Lowest Energy, and Hop Count) that was collected by the Query Response Packet about the four possible paths.

Path	Total Energy	Lowest Energy	Hop Count
ABF	23	11	2
ABCDF	42	4	4
ACDF	31	4	3
ACBF	38	11	3
Sum	134	30	12

Table 6: Data Collected For Sample Example 3

In Table 7 we display the calculated Total Energy Ratio, Lowest Energy Level Ratio, and Hop Count Ratio for each path using equations 2, 4, and 6 and at the end the cost function for each path using equation 1.

Path	Total Energy Ratio	Lowest Energy Ratio	Hop Count Ratio	Cost function
ABF	(23/134)% = 17.16	(11/30)% = 36.66	(2/12)% =16.66	17.16 x 36.66 / 16.66 = 37.76
ABCDF	(42/134)% = 31.34	(4/30)% = 13.33	(4/12)% =33.33	31.34 x 13.33 / 33.33 = 12.53
ACDF	(31/134)% = 23.13	(4/30)% = 13.33	(3/12)% = 25	23.13 x 13.33 / 25 = 12.33
ACBF	(38/134)% = 27.53	(11/30)% = 36.66	(3/12)% = 25	27.53 x 36.66 / 25 = 40.36

Table 7: Cost Function Values for Sample Example 3

According to the calculated cost function values that were shown in Table 7, path ACBF is the chosen one because it does not contain node D and has higher energy levels than other paths not containing node D.

8.5 Power Modeling:

The power consumption in WSN is mainly caused by the transmission part. The transmission power consumption is much higher than the computation power consumption. As an illustration, consider this comparison: The energy needed to transmit 1 KB over a 100m distance is approximately equivalent to the energy necessary to carry out 3 million instructions at a speed of 100 million instructions per second (MIPS) [56].

Also, the power consumption in the ideal state is very low compared with the power consumption in the active state. That is why we will focus on the power consumption in the active state when the node is transmitting or receiving [57].

$$P_{TR} = P_T(d) + P_R$$

Equation 7: Transmission and Receiving Power

Where:

P_{TR}: the power to transmit and receive

$P_T(d)$: the power consumed by the transmitter to transmit over a distance d

P_R: the power consumed by the receiver

The power consumed at transmission can be expressed by this formula:

$$P_T(d) = P_{elec} + P_{amp}(d)$$

Equation 8: Transmission Power

Where:

P_{elec}: the power consumed by the DSP circuit and the Digital/Analog converter

P_{amp}(d): the power consumed by the signal amplifier to transmit for a range of d

The power consumed at the receiver can be expressed by this formula:

$$P_R = P_{elec}$$

8.5.1 Multi-Hop Vs. Single-Hop

P_T (d) >>> P_R because P_{amp}(d) is much higher than P_{elec} so that is why in our protocol we chose to use a multi-hop transmission model compared with a single-hop transmission model [56].

In multi-hop transmission, the power consumption can be calculated as:

$$P_{TR} = \sum_{i=1}^{n}[2 * P_{elec} + P_{amp}(d)]$$

Where n is the number of nodes on the path from the source to the destination

And if we are transmitting a packet of N bits the formula becomes:

$$P_{TR} = N \sum_{i=1}^{n}[2 * P_{elec} + P_{amp}(d)]$$

8.5.1.1 Dynamic Voltage-Frequency Scaling (DVFS)

P_{elec} [57] can be expressed as:

$$P_{elec} = P_{dynamic} + P_{static}$$

Equation 12: Electric Circuit Power Consumption

Where:

$P_{dynamic}$: the dynamic power consumption

P_{static}: the static power consumption

Now P_{static} [57] can be expressed as:

$$P_{static} = I * V$$

Equation 13: Static Power Consumption

Where:

I: current

V: voltage

Now $P_{dynamic}$ [57] can be expressed as:

$$P_{dynamic} = \tfrac{1}{2} * C * V^2 * F$$

Equation 14: Dynamic Power Consumption

Where:

C: capacitor

F: Frequency

From the above formula, we can notice that we can reduce $P_{dynamic}$ by reducing V or F. This technique is called Dynamic Voltage-Frequency Scaling (DVFS)

By using DVFS we can lower the voltage and frequency being used in the sensor node. Lowering the CPU frequency will make the processing speed slower and lowering the signals' voltage will make the transmission prone to more errors and will reduce the data transmission rate as shown in the following formula:

$$R = W * \log_2 \left(1 + \frac{S}{N}\right)$$

<div align="center">

Equation 15: Data transmission rate

</div>

Where:

R: Data transmission rate

S: Signal Level

N: Noise Level

By reducing the voltage, level of the signal (S) the S/N ratio will decrease, causing the data rate to decrease according to Shannon's equation. Therefore, we are sacrificing speed and reliability to maximize the lifespan of nodes.

9 Simulation and Results

The following sections are dedicated to describing the simulation process and its results. The description will include the simulation environment, the simulation process, and the results of comparing the performance of directed diffusion with the proposed protocol.

Our goal in this study is to verify the performance of our proposed protocol. To do this, we compared its performance with directed diffusion. Directed diffusion is a well-known routing protocol for WSNs. Our choice to compare the proposed protocol with directed diffusion was based on the following reasons:

1. The fact that our proposed protocol is a major modification to directed diffusion.
2. Directed diffusion is a well-established protocol, and it is widely used for performance comparison.
3. Both protocols fall under the same category of flat structure routing protocols

During the simulation process, several performance aspects were compared. To try to achieve the most fairness in the comparison process between the two protocols, we used the same environment for both protocols with the same parameters in each experiment. This method will assure that both protocols go through the same conditions when collecting the simulation results.

9.1   Simulation Environment

9.1.1   **Simulator**

To evaluate the performance of our proposed routing protocol we implemented it using a Castalia simulator which is a dedicated Wireless Sensor Network Simulator. It is used as an add-on Framework with OMNeT++. OMNeT++ is a general network simulator.

Castalia is highly parametric which makes it suitable for simulating a wide range of applications in different platforms. Castalia implements sensor nodes as compost modules where each module consist of sub-modules. For example, the communication is a module that consist of routing, MAC, and physical sub-modules [62].

One of the biggest advantages of Castalia is the realistic modeling of the wireless and radio channels. Resulting in a realistic node behavior especially interacting with the wireless medium. This feature makes Castalia an attractive option for researchers who wants to test their protocols.

It was developed in C++ at the National ICT Australia [62].

9.2   Simulation Parameters

In Figure 37 we show a sample of the Castalia parameter file that was used in the simulation experiments.

73

```
[General]

# =========================================================
# Always include the main Castalia.ini file
# =========================================================

include ../Parameters/Castalia.ini

sim-time-limit = 300s


SN.field_x = 60          # meters
SN.field_y = 60          # meters

SN.numNodes = 49
SN.deployment = "7x7"


SN.node[*].Communication.Radio.mode = "IDEAL"
SN.node[*].communication.radio.collisionModel = 0

SN.node[*].Communication.Radio.RadioParametersFile = "../Parameters/Radio/CC2420.txt"
SN.node[*].Communication.MACProtocolName = "TMAC"

SN.node[*].ApplicationName = "proposedProtocl"
SN.node[0].Application.isSink = true
SN.node[*].Application.collectTraceInfo = true
SN.node[*].Application.sampleInterval = 1
SN.node[*].Application.sendAggregatedDataInterval = 1
SN.node[*].Application.sensingNodes = " 48 "
SN.node[*].Application.queryTimeToLive = 2
SN.node[*].Application.queryInterval = 10

SN.node[*].Communication.RoutingProtocolName = "proposedProtoclRouting"

SN.node[*].Communication.Routing.queryResponseTimeToLive = 2
SN.node[*].Communication.Routing.numberOfPathsEnforced = 2
SN.node[*].Communication.Routing.collectTraceInfo = true


SN.wirelessChannel.printDebugInfo = true


# =========================================================
[Config DD]

SN.node[*].Communication.RoutingProtocolName = "directedDiffusion"
```

Figure 37: Castalia parameters File

9.2.1 Comparison Aspects

Our simulation compares the performance of our protocol with the one of directed diffusion. The comparison was made on the following aspects:

1. Energy consumption

2. Reliable delivery

74

3. Load balancing

4. Network lifetime

5. Quality of Service

9.2.1.1 *Energy consumption*

A main focus of the simulation was to compare the total energy consumption between the two protocols in an attempt to support the concept that our proposed routing protocol will result in less energy consumption than directed diffusion.

9.2.1.2 *Reliable delivery*

To be able to support the concept that our protocol provides a higher level of reliable delivery the simulation compares the total number of packets delivered to the sink node between the two protocols.

9.2.1.3 *Load balancing*

A comparison was made of the standard deviation of energy consumption for all the nodes between the two protocols to support the concept that our protocol provides a higher level of load balancing.

9.2.1.4 *Network lifetime*

Network lifetime is an important parameter in WSNs. Network lifetime is quantified by two metrics, namely, the time for the first node death and the time when the sink node stops receiving data packets from the sensing nodes [55].

Priority delivery was compared by sending the same number of packets once as normal flow and a second time as priority flow. The percentage of delivered packets was then used to compare the two cases.

9.2.2 MAC Layer

Castalia has implemented four different Medium Access Control (MAC) modules that are described below.

Tunable MAC

This module can be used for experimental usage especially when some proposals of duty cycle MAC protocols are tested. There are some parameters to adjust the fraction of time or the exact time duration that the node continues to listen. The experiments with CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) also enable parameters that influence carrier sensing [64].

T-MAC and S-MAC

A module simulating protocols T-MAC and its predecessor S-MAC can also be used in Castalia. These protocols are aimed at keeping the energy consumption as low as possible. Since the authors of Castalia did not find all the practical implementation details in the proposal, some protocol properties are adjustable using parameters [64].

IEEE 802.15.4 MAC

This module implements the core functionality of the MAC part of the IEEE 802.15.4 standard, but some features are not implemented. The implementation is aimed at BAN. Among the available functionalities, there are CSMA-CA, Direct data transfer mode and Guaranteed time slots (GTS - a form of TDMA). Indirect transfer mode, and non-beacon or multi-hop personal area networks (PAN), which can be utilized in WSN, are not implemented [64].

Baseline BAN MAC

This module is an implementation of IEEE 802.15.6 draft proposal for a standard in BAN MAC. Since Body Area Networks are out of the scope of this thesis, this module is not described in detail and the readers are referred to [64].

In our implementation, we decided to use T-MAC for its saving energy properties.

9.3 Simulation Results

In this section, we will demonstrate the Castalia simulation results obtained for comparing the performance of directed diffusion with our proposed protocol.

As mentioned earlier the comparison was made on four aspects:

1. Energy consumption
2. Reliable delivery
3. Load balancing
4. Network lifetime
5. Quality of Service

This section will contain five subsections. Each subsection is dedicated to one aspect.

9.3.1 Energy Consumption

To evaluate the performance of our protocol against directed diffusion we have simulated both directed diffusion and the proposed protocol using the same simulation parameters shown in Table 8.

Parameter Name	Value
Simulation Time	300 sec
X axis	40 – 180 meters
Y axis	40 – 180 meters
Number of Nodes	25- 256
Sink node	Node 0
Radio Type	CC2420
MAC Protocol	TMAC

Table 8: Simulation Parameters for Energy Consumption Comparison

For each network size, the simulation experiment was run five times against both directed diffusion and the proposed protocol. The results of the average of the five runs were used to represent the values presented in the energy consumption aspect comparison.

As shown in Table 8, the sink node was always node zero. All the other nodes were assumed to be normal sensing nodes.

The application layer will send a new interest every 10 seconds. The sensing nodes will send a data packet every 1 second.

The simulation focus was to compare the total energy consumption between the two protocols in quest to support that our proposed routing protocol will result in less energy consumption.

The simulation was performed with a different number of nodes ranging from 25 to 256 to support the observation that energy conservation will occur regardless of networks size.

Figure 38 shows the simulation results comparing the total energy consumption for both directed diffusion and the proposed protocol.
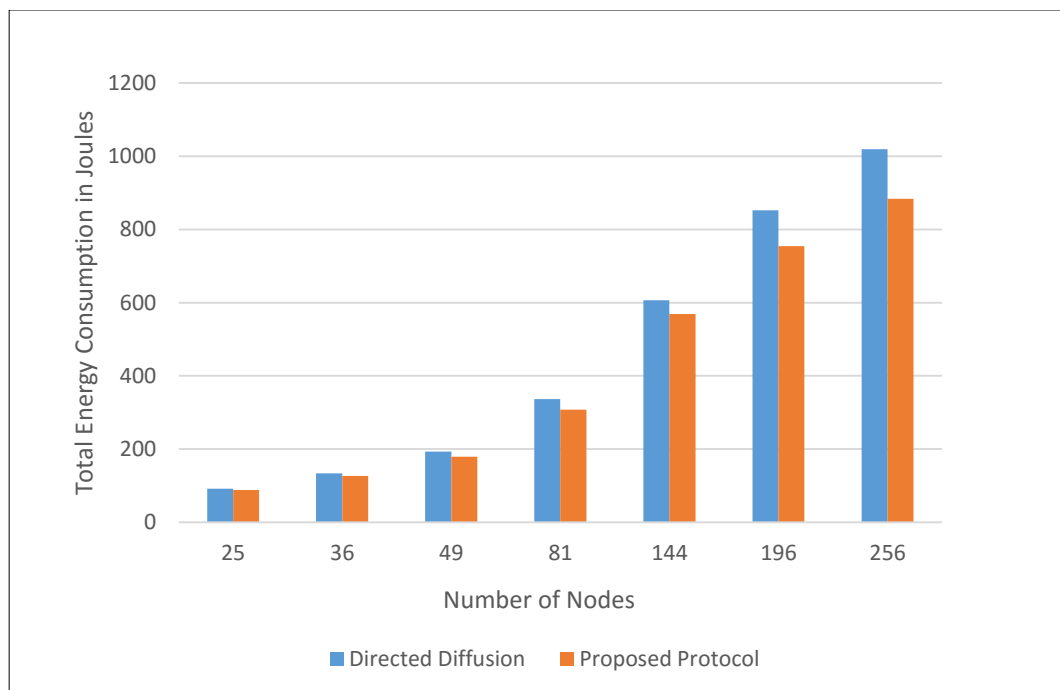
Figure 38: Comparing Total Energy Consumption

Actually, as shown in Figure 39 the difference in the total energy consumption was increasing as the number of nodes increased because the paths tend to be longer in bigger networks and assuring the healthier and shorter paths will decrease energy consumption.
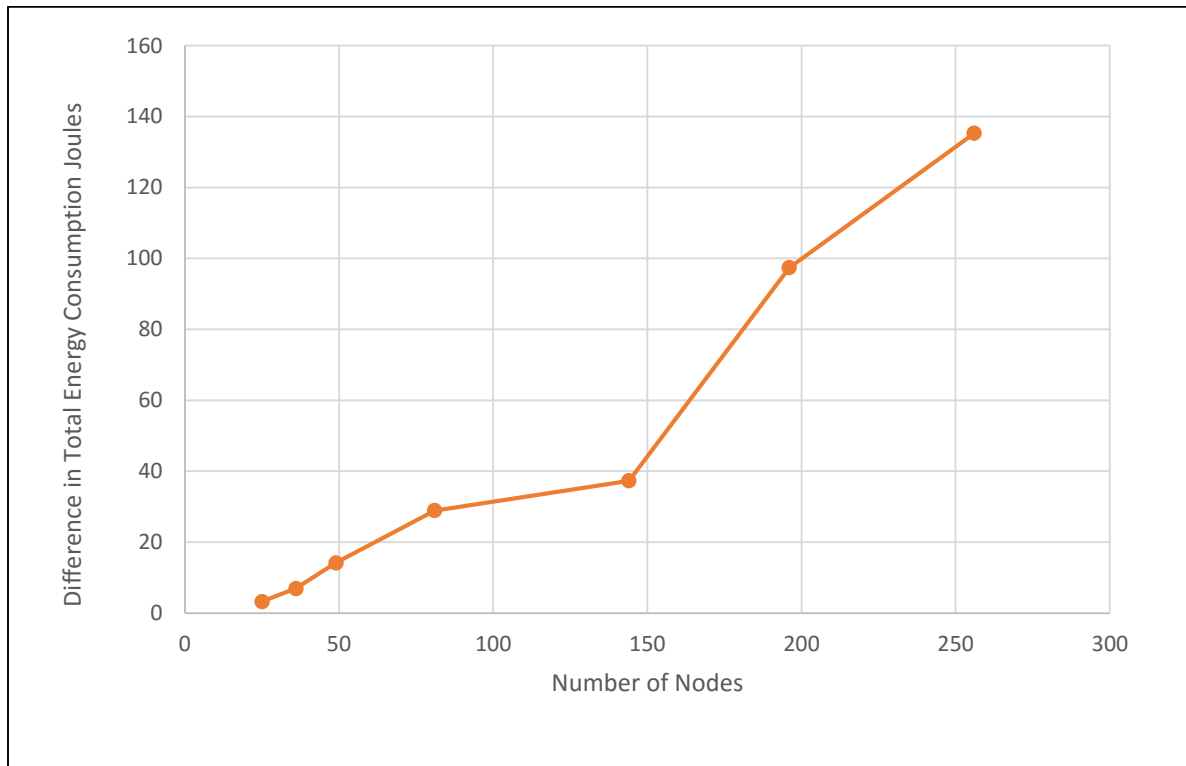
Figure 39: Difference in Total Energy Consumption

From this set of simulation experiments, it is concluded that the proposed protocol energy consumption is less than directed diffusion, and the energy consumption difference increases as the network size increases.

9.3.2 Reliable delivery

To evaluate the performance of our protocol against directed diffusion we have simulated both directed diffusion and the proposed protocol using the same simulation parameters shown in Table 9.

Parameter Name	Value
Simulation Time	300 sec
X axis	60 – 90 meters
Y axis	60 – 90 meters
Number of Nodes	25- 225
Sink node	Node 0
Radio Type	CC2420
MAC Protocol	TMAC

Table 9: Simulation Parameters for Reliable Delivery Comparison

For each network size, the simulation experiment was run five times against both directed diffusion and the proposed protocol. The results of the average of the five runs were used to represent the values presented in the energy consumption aspect comparison.

As shown in Table 9, the sink node was always node zero. All the other nodes were assumed to be normal sensing nodes.

The application layer will send a new interest every 10 seconds. The sensing nodes will send a data packet every 1 second.

The simulation focus was to compare the total number of delivered packets to the sink node between the two protocols in quest to support that our proposed routing protocol will have higher reliable delivery.

The simulation was performed with a different number of nodes ranging from 25 to 225 to support the observation that more packets will be delivered regardless of networks size.

Figure 40 shows the simulation results comparing the total number of packets delivered to the sink node for both directed diffusion and the proposed protocol.
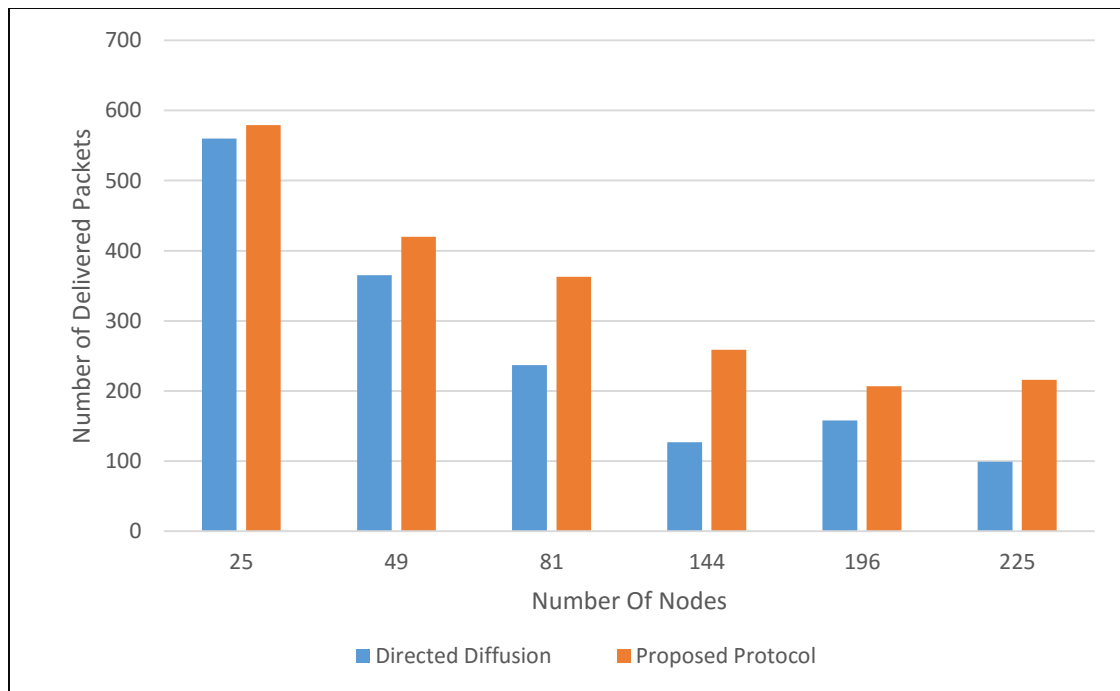
Figure 40: Comparing Total Number of Packets Delivered to Sink Node

Also, as shown in Figure 41 the difference in the total number of delivered packets was always higher and had a tendency to increase as the number of nodes increased.
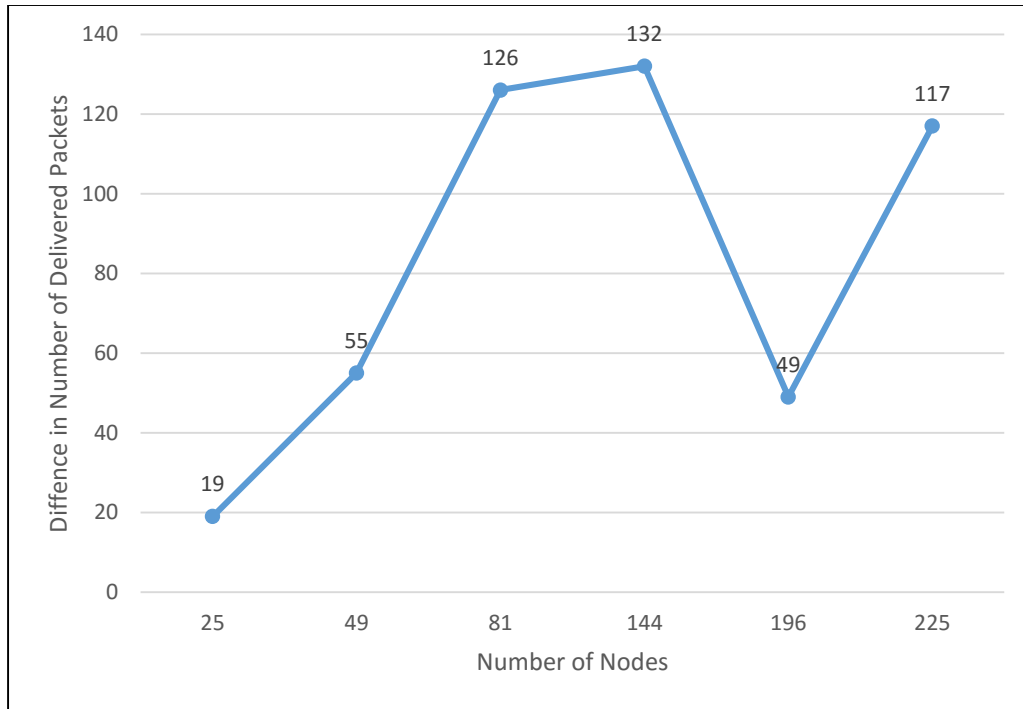
Figure 41: Difference in Total Packets Delivered

From this set of simulation experiments, it is concluded that the proposed protocol packets delivery has higher reliability than directed diffusion.

9.3.3 Load Balancing

To evaluate the performance of our protocol against directed diffusion we have simulated both directed diffusion and the proposed protocol using the same simulation parameters shown in Table 10.

Parameter Name	Value
Simulation Time	300 sec
X axis	60 – 180 meters
Y axis	60 – 180 meters
Number of Nodes	25- 225
Sink node	Node 0
Radio Type	CC2420
MAC Protocol	TMAC

Table 10: Simulation Parameters for Load Balancing Comparison

For each network size, the simulation experiment was run five times against both directed diffusion and the proposed protocol. The results of the average of the five runs were used to represent the values presented in the energy consumption aspect comparison.

As shown in Table 10, the sink node was always node zero. All the other nodes were assumed to be normal sensing nodes.

The application layer will send a new interest every 10 seconds. The sensing nodes will send a data packet every second.

The simulation focus was to compare the load balancing between the two protocols in an effort to support the observation that our proposed routing protocol will have better

load balance between the nodes. To achieve this, we calculated the standard deviation of the nodes' energy after the simulation period finished.

The simulation was performed with a different number of nodes ranging from 25 to 225 to support the concept that our protocol will balance the load more efficiently than directed diffusion regardless of the size.

Figure 42 shows the simulation results comparing the standard deviation of energy levels for all nodes for both directed diffusion and the proposed protocol.
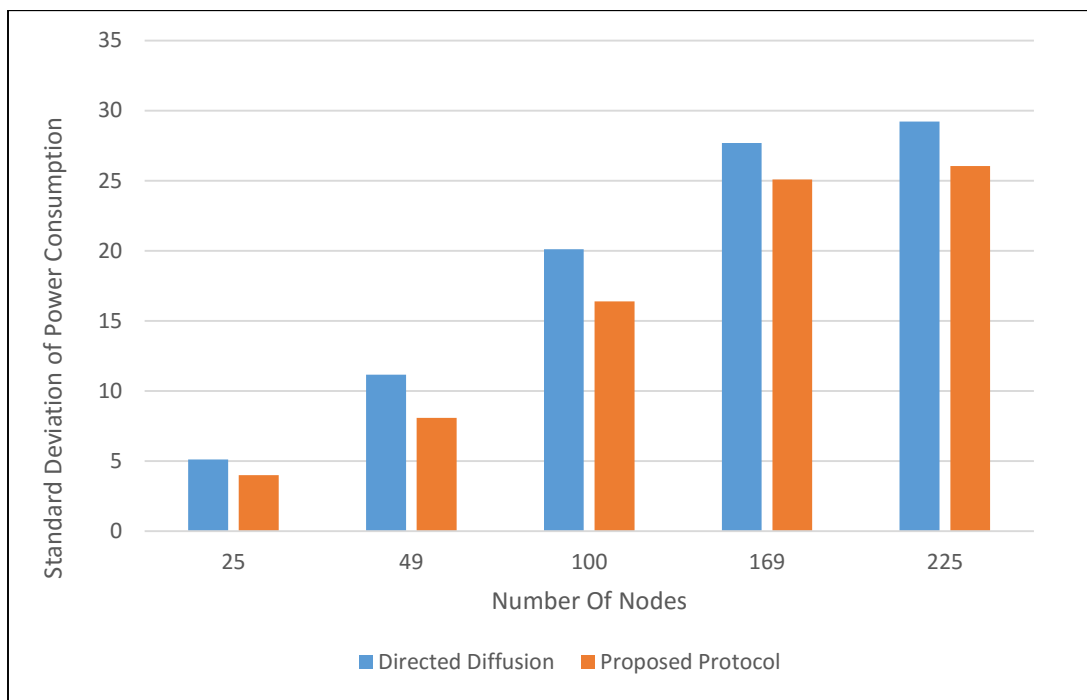
Figure 42: Comparing Standard Deviation of Power Consumption

Also, as shown in Figure 43 the difference in the standard deviation of energy levels of the nodes tends to increase as the number of nodes increased.
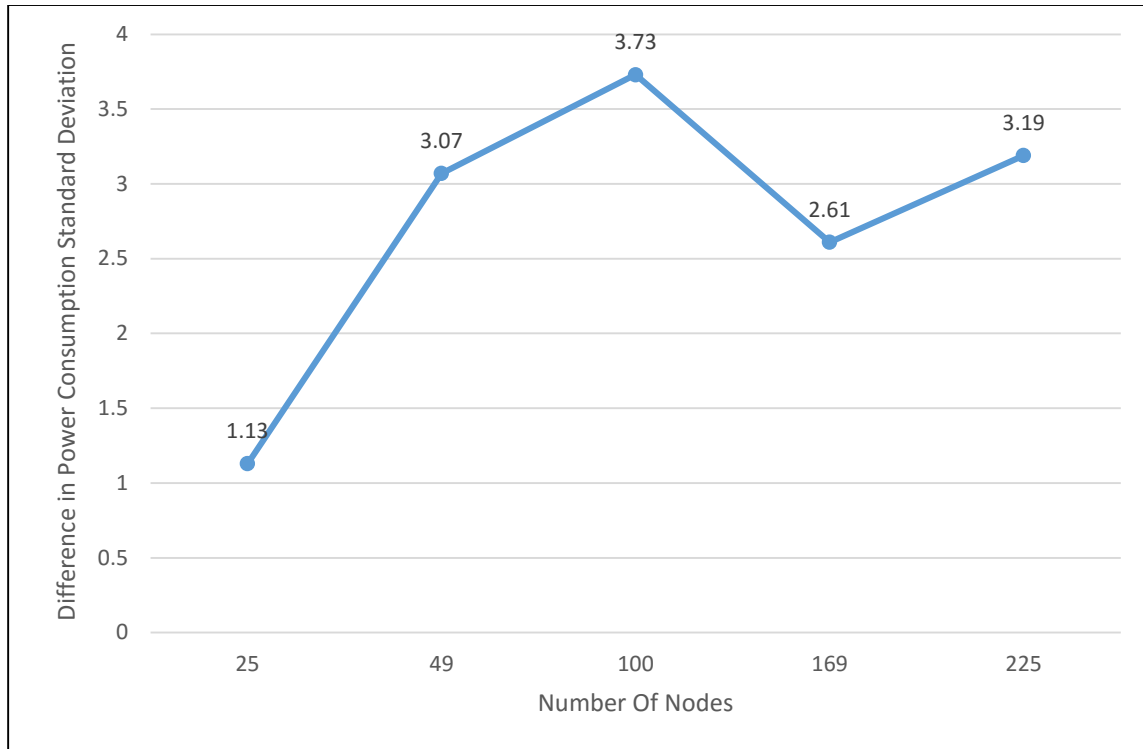
Figure 43: Difference in Power Consumption Standard Deviation

From this set of simulation experiments it is concluded that although the difference was not high, the proposed protocol balanced energy consumption among nodes more equally than directed diffusion.

9.3.4 Network lifetime

To evaluate the performance of our protocol against directed diffusion we have simulated both directed diffusion and the proposed protocol using the same simulation parameters shown in Table 11.

Parameter Name	Value
Simulation Time	3600 sec
X axis	60 – 180 meters
Y axis	60 – 180 meters
Number of Nodes	25- 225
Sink node	Node 0
Radio Type	CC2420
MAC Protocol	TMAC

Table 11: Simulation Parameters for Life Time Comparison

For each network size, the simulation experiment was run five times against both directed diffusion and the proposed protocol. The results of the average of the five runs were used to represent the values presented in comparison of the energy consumption.

As shown in Table 11, the sink node was always node zero. All the other nodes were assumed to be normal sensing nodes.

The simulation runtime in these experiments was increased to 3600 seconds, so the nodes drained completely from energy, and we could reach a state when the network was dead. By dead, we mean the sink stopped receiving any more date packets.

The application layer sent a new interest every 100 seconds. It was assumed that all the sensing nodes would sense data that matched the data requested in the interest. This was done to maximize the load on the network.

To be able to measure the network lifetime, two aspects were observed. The first aspect was the time when the sink node stopped receiving any more packets and the second aspect was the time when the first node died.

The simulation was performed with a different number of nodes ranging from 25 to 225 to support the observation that our protocol will increase network lifetime regardless of the size.

Figure 44 shows the simulation results comparing the network lifetime for both directed diffusion and the proposed protocol.
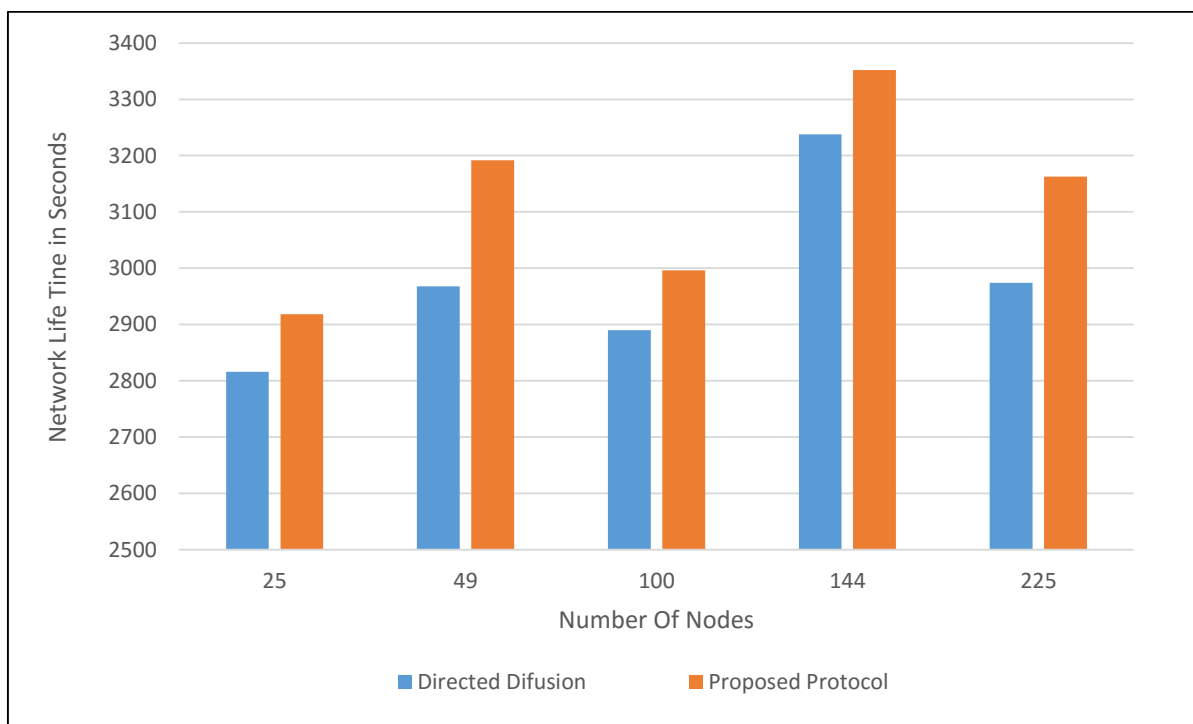
Figure 44: Comparing Network Life Time

Figure 45 shows the simulation results comparing the time the first node died for both directed diffusion and the proposed protocol.
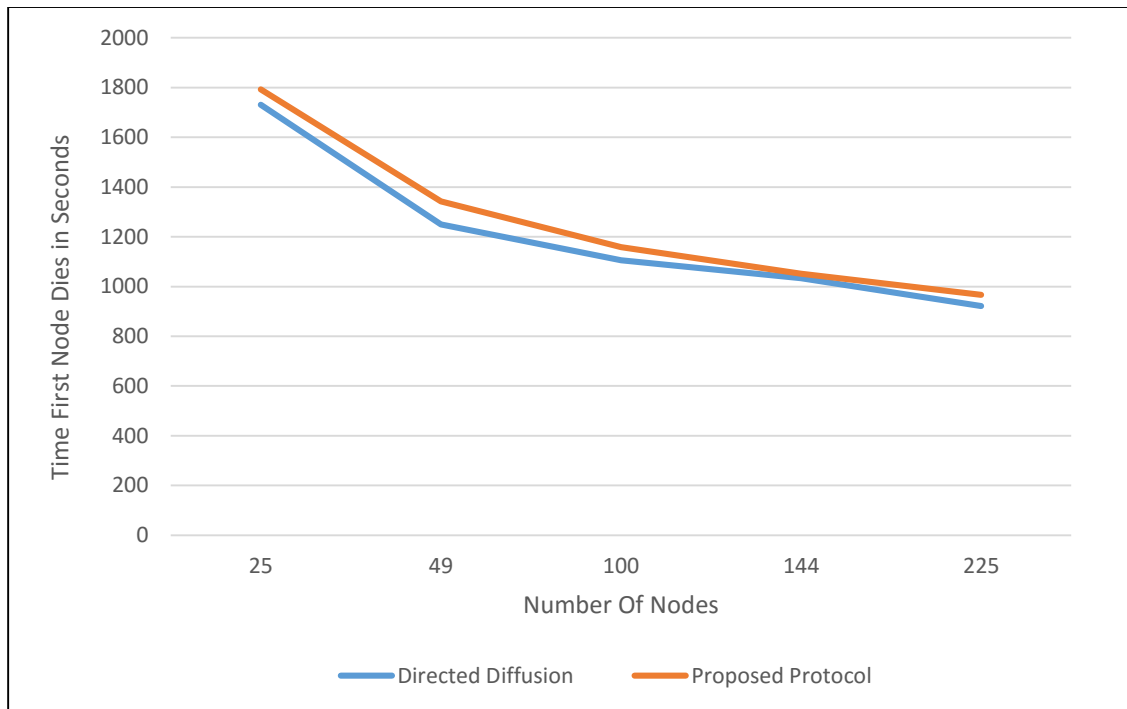
Figure 45: Comparing First Node Death Time

From this set of simulation experiments, it is concluded that the proposed protocol will extend the network lifetime more than directed diffusion.

9.3.5 Quality of Service

This experiment was different from the others because the comparison was made between the proposed protocol and itself. This was done to show that improvements in critical packets delivery were due to QoS features added to the proposed protocol, not due to another improvement.

Since we are featuring two aspects of QoS in our protocol, multipath, and differentiated services, we have here two sets of experiments. The first section will show results for comparing packets delivery when critical packets are transmitted over two paths instead

of one. Section two will show results for comparing packets' arrival delay and packets

delivery when differentiated services are implemented.

Experiments shown in the QoS section were conducted using the parameters shown in

Table 12.

Parameter Name	Value
Simulation Time	450 sec
X axis	60 – 180 meters
Y axis	60 – 180 meters
Number of Nodes	25- 225
Sink node	Node 0
Radio Type	CC2420
MAC Protocol	TMAC

Table 12: Simulation Parameters for Critical Packets Delivery

For each network size, the simulation experiment was run five times against both

directed diffusion and the proposed protocol. The results of the average of the five runs

were used to represent the values presented in the energy consumption aspect

comparison.

As shown in Table 12, the sink node was always node zero. All the other nodes were

assumed to be normal sensing nodes.

The application layer sends a new interest every 10 seconds. The sensing nodes sends a data packet every second.

9.3.5.1 *Multipath*

In this experiment we decided to have only one sensing node, which was always node the with number = (network size -1), so it will easier for us to count exactly how many packets were sent. The number of packets sent was 300.

Also, we increased the interference level by changing the following parameter in the Castalia parameters file:

SN.node[*].communication.radio.collisionModel = 1

Changing the parameter value to 1 will increase the collision level to the highest.

The number of paths enforced in case of critical data was 2. This was done by changing the following parameter value to 2:

SN.node[*].Communication.Routing.numberOfPathsEnforced = 2

Two paths will be enforced, only, if the interest is marked as critical by setting the priority field in the interest packet to 1.

The simulation focus was to compare the total number of delivered packets to the sink node between the normal packets and critical packets.

The simulation was performed with a different number of nodes ranging from 25 to 225 to support the observation that energy conservation will occur regardless of networks size.

Figure 46 shows the simulation results comparing the total number of packets delivered between normal data and critical data.
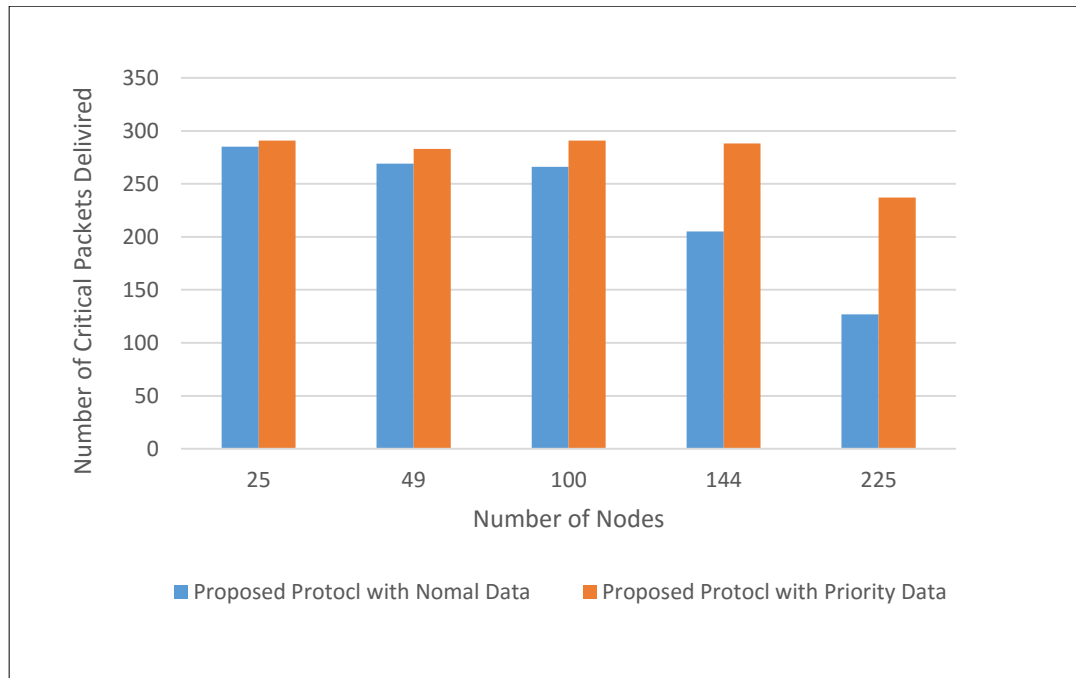
Figure 46: Comparing Total Number of Packets Delivered Based on Flow Type

Figure 47 and Table 13 shows the difference in the total number of delivered packets was always higher in the case of critical data. While the difference was small in smaller networks, it increased in bigger networks because hops are higher, and the possibility of data loss is higher.

Network Size	Normal Data Delivery	Critical Data Delivery	Different in Delivery	Improvement Rate (%)
25	285	291	6	2%
49	269	283	14	4.6%
100	266	291	25	8.3%
144	205	288	83	27.6 %
225	127	237	110	36.6%

Table 13: Improvement Rate of Critical Delivery
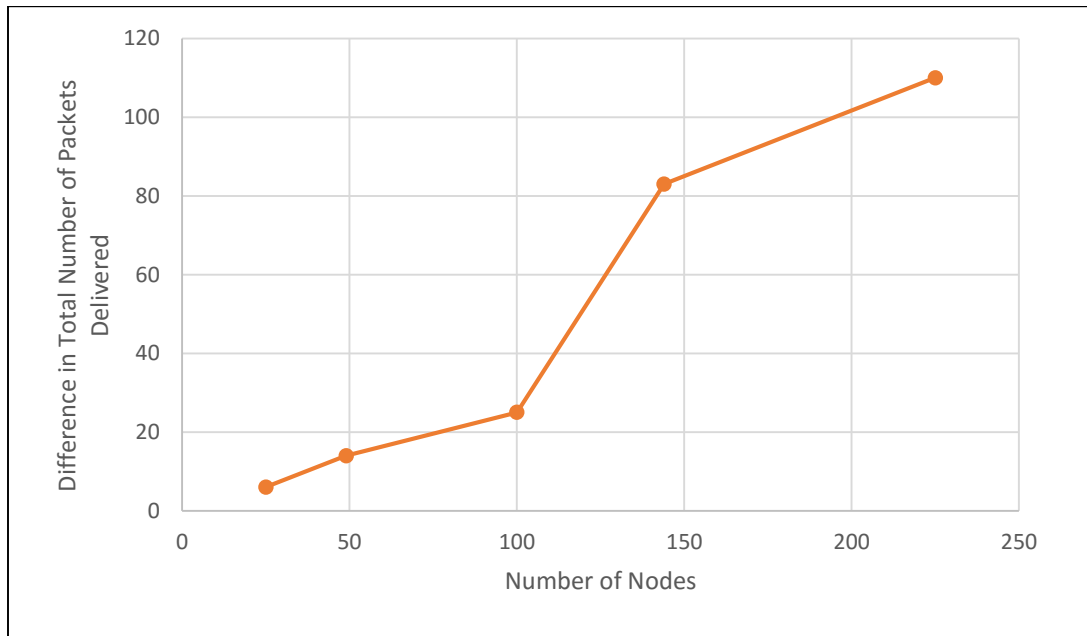
Figure 47: Difference In Total Number of packets Delivered in Critical Delivery

From this set of simulation experiments, it is concluded that the proposed protocol can increase the reliable delivery of critical packets by enforcing more paths.

Differentiated services

In this experiment, we are comparing the number of packets delivered and the average arrival delay of urgent packets before and after adding the differentiated services aspect to our proposed protocol. The simulation will run twice. The first time all packets will go through the same buffer regardless of their class (single service). In the second time, each packet class will go through a dedicated buffer (differentiated services).

Packets in the urgent buffer are given the highest priority, then critical packets, and lastly normal packets are transmitted. To measure the average arrival delay, the time of each packet arrival was recorded; then the difference between each two successive packets was obtained and added up and then divided by the number of packets delivered.

Figure 48 shows the simulation results comparing the total number of urgent packets delivered once using single service and again using differentiated services.
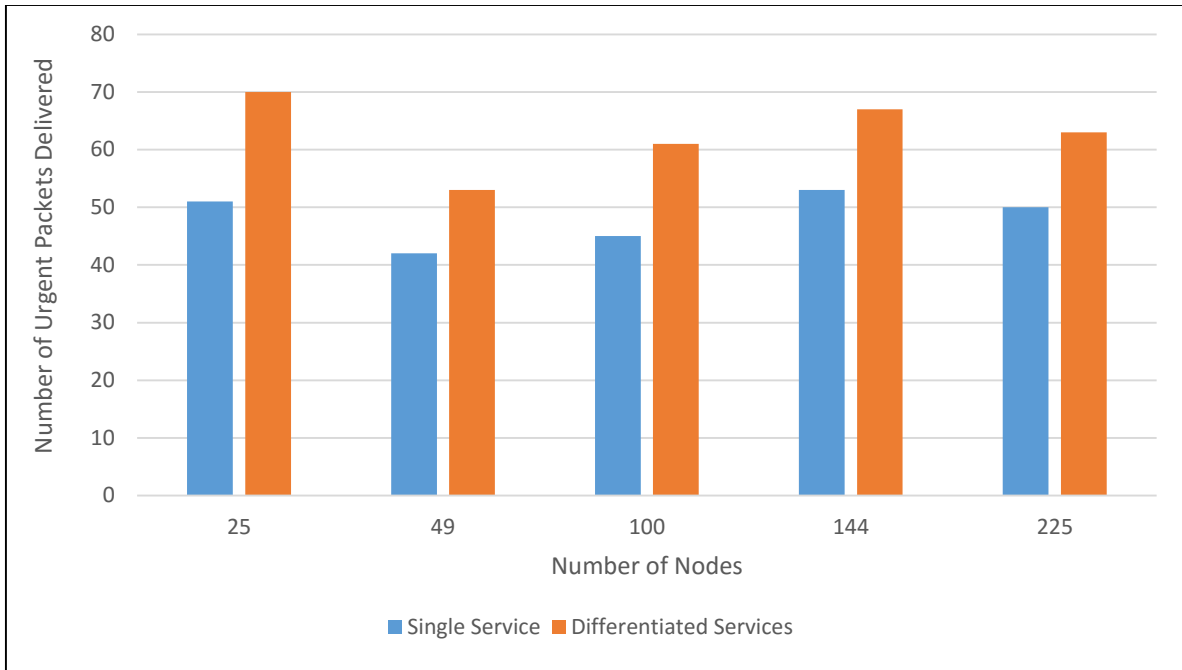
Figure 48: Comparing Number of Urgent Packets Delivered to Sink Node

Figure 49 shows the simulation results comparing the average arrival delay of urgent packets once using single service and again using differentiated services.
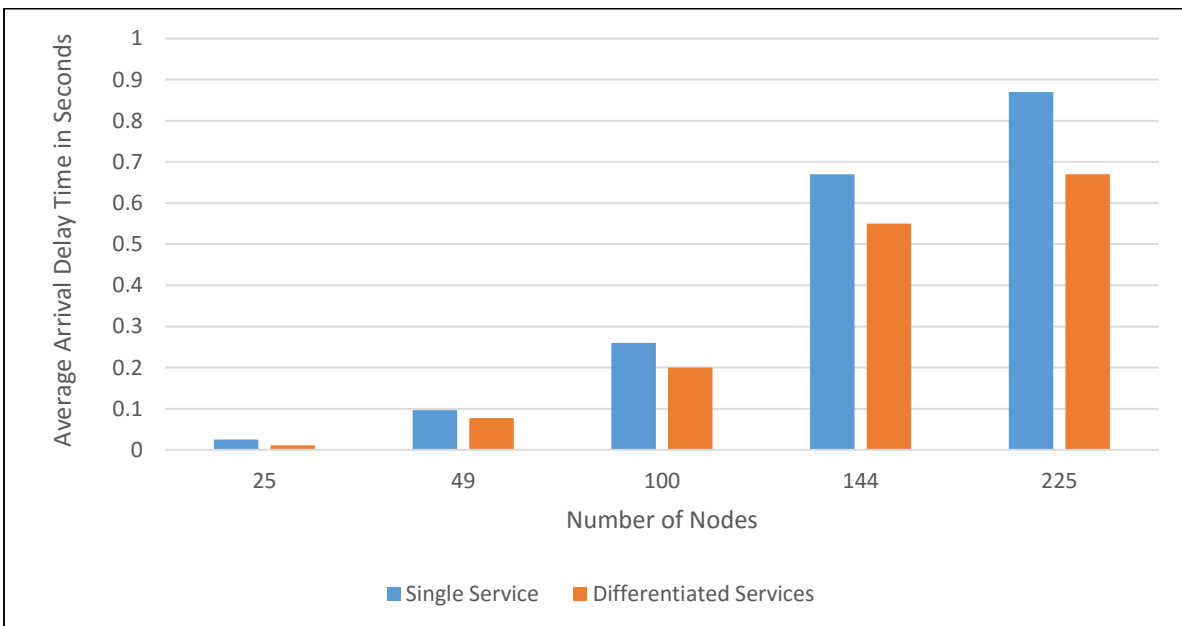
Figure 49: Comparing Average Arrival Delay of Urgent Packets

From this set of simulation experiments, it is concluded that using differentiated services can increase the QoS level by increasing the packets throughput and decreasing the delay of a certain class of data over other classes. This is crucial because not all data classes have the same needs.

10 Conclusion

Designing routing algorithms for Wireless Sensor Networks (WSNs) is a challenging task due to the nature of WSNs. WSNs nodes are generally limited in energy and computation power beside the instability of wireless links.

In this research, a routing protocol was presented that falls under the flat-structure category. The protocol uses a non-incremental machine learning technique to give preference to healthier and shorter paths, which will result in less energy consumption, more reliable delivery of data, better load balancing, and longer network life.

In addition to preferring healthier paths, our protocol provides QoS features through implementing differentiated services, where packets are classified as critical, urgent, and normal. Based on this classification, different packets are assigned different priorities and resources. This results in a more reliable delivery and less delay for urgent and critical packets.

The proposed protocol achieves its objectives by collecting data about the available paths through a special data collection stage. The collected data are used to choose the best path using a mathematical formula based on merit criteria.

In this work, we collected data about the total energy on the path, path hop count, and the lowest energy node on the path to set up a working experiment with different data sets.

We compared our protocol with a well-known flat-structure routing protocol called Directed Diffusion. The Directed Diffusion algorithm uses directed routing by using gradients to point back to the sink node and uses a naming scheme to minimize data flooding. However, Directed Diffusion algorithm does not put any efforts to be aware of the network status.

A possible drawback in our proposed protocol is an increase in the first packet arrival time. This delay is caused by adding the collecting data stage. The Directed Diffusion algorithm starts delivering packets as soon as the path is enforced, unlike our protocol where data must be collected before a path is enforced.

Another drawback could result in the case of queries with long life span. In this case, the chosen path could, after a certain time, no longer be the best path because of the topology changes that could result in the network topology.

In the simulation, the performance of our protocol was compared with the Directed Diffusion algorithm through five aspects: Energy Consumption, Reliable Delivery, Load Balancing, Network Lifetime, and Quality of Service. The simulation was conducted on different network sizes.

The simulation results did not point out a significant difference in performance between the proposed protocol and the Directed Diffusion algorithm in smaller networks. However, when the network size started to increase the results showed better performance by the proposed protocol.

Furthermore, we presented an aware diffusion routing protocol that presented a new paradigm by adding a new stage for collecting data about the status of paths and by storing critical data on multiple paths as a means of their reliable delivery to the sink node.

11 Future Work

The proposed routing protocol was designed to collect data about the paths' status and then make a decision about which path to enforce using the collected data. In our specific design; we collected data about the total energy of the path, the path hop count, and the lowest energy level of a node on the path. More experimentation could be done with different data, such as the delay, for example, and see how this could affect the performance.

Another possible future work is using weights in the cost function equations for the different data collected to give higher or lower preference to certain data types over others.

A promising improvement could be exploiting the charge and discharge cycles caused by solar energy harvesting. We can utilize these charging cycles to make better routing decisions. For example, nodes in charging status could be avoided until they reach a certain energy level before putting them in action again.

Cross-layer optimization continues to be a promising area for performance enhancement. The more data are exchanged between layers the better the overall performance. A more possible data exchange could be done as follows:

- MAC layer buffer status can be reported to the routing layer so that congested nodes are avoided.
- When the battery level is below a threshold level at a node, the routing layer can send a control message to the physical layer to lower the clock frequency. Although this will decrease the processing speed it will also decrease dynamic

the power consumption as shown in equation14, and eventually increase the node's lifetime.

To enhance providing QoS, the differentiated services could be implemented at the MAC layer level as well. The same packets classification can be used, where packets are classified into critical, urgent, and normal. Then, urgent and critical packets can be served faster by assigning them shorter interframe spacing (SIFS, PIFS, and DIFS) compared with normal packets

12 References

1.   Paolo Bellavista, G.C., *Convergence of MANET and WSN in IoT Urban Scenarios.* IEEE Sensors Journal, 2013. Vol. 13. pp. 3558 - 3567

2.   Pantazis N.A. , V.D.D., *A survey on power control issues in wireless sensor networks.* IEEE Communications Surveys & Tutorials, 2007. pp. 86 - 107.

3.   Louis Coetzee, J.E., *The Internet of Things - promise for the future- An introduction*, IST-Africa 2011 Conference Proceedings. 2011. pp. 1 - 9

4.   Atzori, L., A. Iera, and G. Morabito, *The Internet of Things: A survey.* Computer Networks, 2010. pp. 2787-2805.

5.   Adrian McEwen, H.C., *Designing the Internet of Things*. 2013: Wiley.

6.   Kazem Sohraby, D.M., Taieb Znati, *Wireless Sensor Networks Technology, Protocols, and Applications*. 2007: Wiley.

7.   Nikolaos A. Pantazis, S.A.N.a.D.D.V., *Energy-Efficient Routing Protocols in WirelessSensor Networks: A Survey.* IEEE Communications Surveys & Tutorials, 2012. pp. 551 - 591.

8.   Dr. Geoff V Merret, D.Y.K.T., *Wireless Sensor Networks: Application-Centric Design*. 2010: InTech.

9.   Alnuaimi, M., Sallabi, F. ; Shuaib, K., *A survey of Wireless Multimedia Sensor Networks challenges and solutions,* IEEE International Conference on Innovations in Information Technology (IIT). 2011,: Abu Dhabi pp. 191 - 196

10. Ian F. Akyildiz, M.C.V., *Wireless Sensor Networks*. 2010: John Wiley & Sons Ltd.

11. Holger Karl, A.W., *Protocols and Architectures for Wireless Sensor Networks*. 2005: John Wiley and Sons.

12. Waltenegus Dargie, C.P., *Fundamentals Of Wireless Sensor Networks Theory And Practice*. 2010, United Kingdom: Wiley.

13. Amiya Nayak, I.S., *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*. 2010: Wiley-Interscience.

14. Mahalik, N.P., *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*. 2007: Springer.

15. Long Zhaohua, L.J., Jiang Guiquan, *Survey of Routing Protocols in Wireless Sensor Networks Based on Applications.* 2010 2nd International Conference on Industrial and Information Systems (IIS), 2010. pp. 381-385.

16. Mohammad Ilyas, I.M., *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. 1 ed. 2004: CRC Press.

17. P. Jiang, Y.W., J. Wang, X. Shen, and A. Xue. *A Study of Routing Protocols in Wireless Sensor Networks*. Proceedings of the 6th World Congress on Intelligent Control and Automation. 2006. Dalian, China. pp. 266 - 270

18. Zwierzykowski, B.M.a.P., *Survey of Simulators for Wireless Sensor Networks.* International Journal of Grid and Distributed Computing, 2012: pp. 23-50.

19.   Subir Kumar Sarkar, T.G.B., C. Puttamadappa, *Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications*. 2 ed. 2013: CRC Press.

20.   Al-Karaki, J.N., Kamal, A.E., *Routing techniques in wireless sensor networks: a survey.* IEEE Wireless Communications, 2004. pp. 6 - 28.

21.   Kemal Akkaya, M.Y., *A survey on routing protocols for wireless sensor networks*, in *Ad Hoc Networks 3*. 2005, Elsevier. pp. 325-349.

22.   Md. Atiqur Rahman, S.A., Md. I1eas Pramanik, Md. Ferdous Rahman. *A Survey on Energy Efficient Routing Techniques in Wireless Sensor Network*, 15th International Conference on Advanced Communication Technology (ICACT), 2013, pp. 200 - 205

23.   Shio Kumar Singh, M.P.S., D K Singh, *Routing Protocols in Wireless Sensor Networks –A Survey.* International Journal of Computer Science & Engineering Survey, 2010. pp. 63-83.

24.   Matin, M., *Wireless Sensor Networks - Technology and Protocols*. 2012: InTech.

25.   Sungjin Ahn, D.Kim, *Proactive context-aware sensor networks*, EWSN'06 Proceedings of the Third European conference on Wireless Sensor Networks, 2006, pp. 38 - 53

26.   M. Dubois, M.A., and P. Stenstrom, *Parallel Computer Organization and Design*. 2012: Cambridge Publisher.

27.   Chalermek Intanagonwiwat, R.G., Deborah Estrin, John Heidemann, Fabio Silva, *Directed Diffusion for Wireless Sensor Networking.* IEEE/ACM Transactions On Networking, 2003. pp. 2-16.

28.   Kazem Sohraby, D.M., Taieb Znati, *Wireless-Sensor-Networks-Technology-Protocols-and-Application*. 2007: Wiley.

29.   Ghane, M., Rajabzadeh, A., *Remaining-energy based routing protocol for wireless sensor network*, 15th CSI International Symposium on Computer Architecture and Digital Systems (CADS). 2010, IEEE: Tehran. pp. 67 - 73.

30.   V. Handziski, A. Kopke, H, Karl, C. Frank, W. Drytkiewicz, "*Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering*," EWSN 2004, LNCS 2920, pp. 172–187, 2004.

31.   X. Liu, F. Li, H. Kuang, X. Wu, *The Study of Directed Diffusion Routing Protocol Based on Clustering for Wireless Sensor Network*, Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress, Vol. 1, pp. 5120-5124, 2006.

32.   Y.-S. Chen, Y.-W. Nian, J.-P. Sheu, *An energy-efficient Diagonal-Based Directed Diffusion for Wireless Senor Networks*, Parallel and Distributed Systems, 2002. Proceedings of Ninth International Conference, pp. 445- 450, 2002.

33.   Bi Jiana, Xu, E, *A Secure and Energy-efficient Data Aggregation Protocol based on Directed Diffusion*, International Symposium on Information Science and Engineering (ISISE), 2012, pp. 437 - 440

34. D. Cheng, Y. Song , Y. Mao , X. Wang , *LDDP: A Location-Based Directed Diffusion Routing Protocol For Smart Home Sensor Netwok*, 2nd International Conference on Systems and Informatics (ICSAI), 2014, pp. 510 - 514

35. Jisul Choe, Keecheon Kim, "*EADD: Energy Aware Directed Diffusion for Wireless Sensor Networks*", International Symposium an Parallel and Distributed Processing with Applications, 2008

36. Hao Qi, F. Wang, B. Wang ,*Energy aware adaptive directed diffusion algorithm of wireless sensor network*, International Conference on Computer Science and Service System (CSSS), 2011, pp. 385 - 388

37. Li Zhiyu, S. Haoshan, *Design of Gradient and Node Remaining Energy Constrained Directed Diffusion Routing for WSN*, International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2007. pp. 2600 - 2603

38. Asokan, R., *A Review of Quality of Service (QoS) Routing Protocols for Mobile Ad hoc Networks*, in *ICWCSC 2010*. 2010, IEEE: Chennai. p. 1 - 6.

39. Changle Li , H.Z., Binbin Hao and Jiandong Li, *A Survey on Routing Protocols for Large-Scale Wireless Sensor Networks.* sensors, 2011: pp. 3498-3526.

40. Fonash, S.J., *Solar Cell*. Britannica.

41. Perry, T., *Self-Charging Cell Phone Screens Coming Soon*, in *IEEE Spectrum*. 2013.

42.   Atwater, H., *Full Spectrum Boosts Solar Cell Power*, Photonics Spectra. 2012, Photonics Media. p. 39.

43.   S. Sudevalayam and P. Kulkarni, *Energy Harvesting Sensor Nodes: Survey and Implications*, IEEE Communications Surveys & Tutorials, VOL. 13, NO. 3, third quarter 2011, pp. 443 - 461

44.   A. Nasiri; S. A. Zabalawi; G. Mandic, *Indoor Power Harvesting Using Photovoltaic Cells for Low-Power Applications,* IEEE Transactions on Industrial Electronics, Volume:56 , Issue: 11, pp. 4502 – 4509

45.   J. Mbowe, G. Oreku, *Quality of Service in Wireless Sensor Networks*, Wireless Sensor Network, 2014, 6, pp. 19-26

46.   B. Bhuyan, H. Sarma, N. Sarma, A. Kar, R. Mall, *Quality of Service (QoS) Provisions in Wireless Sensor Networks and Related Challenges*, Wireless Sensor Network, 2010, Vol. 2, pp. 861-868

47.   N. Ota, D. Hooks, P.Wright, D. Auslander, and T. Peffer., *Application to Demand Response Energy Pricing*, Proceedings of the First international conference on Embedded Networked Sensor Systems, November 05–07, 2003

48.   E. Felemban, C. Lee, and E. Ekici. *MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks*, IEEE Transitions on Mobile Computing, vol. 5, no. 6, pp. 738–754, 2006

49.   D. Ganesan et al. *Highly resilient, energy efficient multipath routing in wireless sensor networks*, Mobile Computing and Communications Review, vol. 5, no. 4, pp. 11–25, 2002

50.   X. Huang and Y. Fang. Multiconstrained QoS multipath routing in wireless sensor networks, Wireless Networks Journal, vol. 14, no. 4, pp. 465–478, 2007

51.   C. Lu et al., *RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks*, In Proceedings of the Eighth Real-Time and Embedded Technology and Applications Symposium, IEEE CS Press, Los Alamitos, CA, 2002, pp. 55 - 66

52.   T. He, J.A. Stankovic, C. Lu, and T.F. Abdelzaher. *SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks*, In Proceedings of International Conference on Distributed Computing Systems (ICDCS '03), Providence, RI, May 2003, pp. 46 - 55

53.   K. Liu, N. Abu-Ghazaleh, and K.D. Kang. *JiTS: Just-in-Time Scheduling for Real-Time Sensor Data Dissemination*. In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06),Washington, DC, IEEE Computer Society, Silver Spring, MD, 2006, pp. 42–46

54.   Shebli, F.D., I. ; M'foubat, A.O. ; Rivenq, A. ; Rouvaen, J.M., *Minimizing energy consumption within wireless sensors networks using optimal transmission range*

between nodes in *IEEE International Conference on Signal Processing and Communications, 2007. ICSPC 2007*. 2007: Dubai. pp. 105 - 108

55.   Shamsan Saleh, A.M.A., Borhanuddin Mohd ; Rasid, Mohd Fadlee A ; Ismail, Alyani, *A Self-Optimizing Scheme for Energy Balanced Routing in Wireless Sensor Networks Using SensorAnt.* Sensors (Basel, Switzerland), 2012. pp. 11307-11333.

56.   Qin Wang; Hempstead, M.W.Y., *A Realistic Power Consumption Model for Wireless Sensor Network Devices.* IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. pp. 286 - 295

57.   Patterson, J.L.H.D.A., *Computer Architecture, Fifth Edition: A Quantitative Approach*. 4 ed. 2006: Morgan Kaufmann.

58.   Milica Pejanović đurišić, Z.T., Goran Dimić, Veljko Milutinović, *A Survey of Military Applications of Wireless Sensor Networks.* Mediterranean Conference on Embedded Computing MECO, 2012, pp. 196 - 199

59.   Sudip Misra, I.W., Subhas Chandra Misra, *Guide to Wireless Sensor Networks*. 2009: Springer.

60.   Forster, A., *Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey*, in IEEE ISSNIP, 2007,. pp. 365 - 370.

61.   Sameer Tilak, K.C., Nael B. Abu-Ghazaleh, and Tony Fountain, *Dynamic Resource Discovery for Sensor Networks*, in EUC Workshops 2005. 2005. pp. 785–796.

62.   *Castalia Offical Web Site*. Available from: https://castalia.forge.nicta.com.au/index.php/en/.

63.   N. H. Mak ; Sch. of Comput., Nat. Univ. of Singapore, Singapore ; W. K. G. Seah, *How Long is the Lifetime of a Wireless Sensor Network?,* International Conference on Advanced Information Networking and Applications, 2009. AINA '09, pp. 763 - 770

64.   Boulis, A., *Castalia User's Manual*. 2011. https://castalia.forge.nicta.com.au/index.php/en/.

65.   Stehl´ık, M., *Comparison of Simulators for Wireless Sensor Networks*, in *Faculty of Informatics*. 2011, Masaryk University: Brno, Czech Republic.

66.   OSI Model, 2015. http://realitypod.com/2012/03/osi-model/

67.   A. Goldsmith, B. Girod, *Cross-Layer Design of Ad-hoc Wireless Networks for Real-Time Media*, 2015. http://msw3.stanford.edu/~zhuxq/adhoc/

68.   NASA Solar Energy Information, 2015. http://pics-about-space.com/nasa-solar-energy-information?p=2#

69.   A. Kansal; J. Hsu; S. Zahedi; V. Raghunathan; J. Friedman, *Low-Power Platforms: Mote Heliomote Energy Harvesting System*, 2015.

http://research.cens.ucla.edu/projects/2006/Systems_Infrastructure/Mote/default.htm

70.   DiffServ -- The Scalable End-to-End QoS Model.

http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.html

13 Curriculum Vitae

Kamil Samara

Education

B.Sc., Al-Balqa Applied University, Jordan, May 2000

Major: Computer Engineering

MPhil, De Montfort University, UK, May 2004

Major: Computer Science

PhD, University of Wisconsin-Milwaukee, May 2016

Dissertation Title: Energy Efficient, Load Balancing, And Reliable Routing Protocol for Wireless Sensor Networks

Work Experience

8/15 – still   Visiting Faculty

Illinois Wesleyan University, Bloomington, IL

2/07 – 05/15   Teaching Assistant

University of Wisconsin - Milwaukee

10/04 – 8/06   Full-time Lecturer

Philadelphia University, Jordan

8/00 – 2/03   Analyst/Developer

Mid Information Technology, Jordan

14 Publications

Journals:

1) Kamil Samara, Hossein Hosseini, "Aware Diffusion: A Low Power and Reliable Delivery Routing Protocol for Wireless Sensor Networks", International Journal of Research in Wireless Systems (IJRWS), December 2015 (Accepted)

2) Kamil Samara, Hossein Hosseini, "Aware Diffusion: A Semi-Holistic Routing Protocol for Wireless Sensor Networks", Wireless Sensor Network Journal, Vol.8 No.3, March 2016, pp. 37-49

Conferences:

1) Kamil Samara, Hossein Hosseini, "A Routing Protocol for Wireless Sensor Networks with Reliable Delivery of Data", The 2015 IEEE International Conference on Internet of Things (iThings2015), Sydney, Australia, December 2015 , pp. 632 – 635

2) Kamil Samara, Hossein Hosseini, "A Routing Protocol for Wireless Sensor Networks with Low Power Consumption", CATA 2016 Las Vegas, Nevada, April 2016 (Accepted)

3) Kamil Samara, Hossein Hosseini, "A Routing Protocol for Wireless Sensor Networks with Load Balancing", 2016 IEEE International Conference On Electro/Information Technology (EIT 2016), Grand Forks, North Dakota, May 2016 (Accepted)