A NOVEL ENERGY EFFICIENT DISTRIBUTED STORAGE SYSTEM FOR DATA SURVIVABILITY IN WIRELESS SENSOR NETWORKS

BY

MOHAMMED ABDULLAH ASIRI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

APRIL 2019

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MOHAMMED ABDULLAH ASIRI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis Committee

Prof. Tarek R. Sheltami (Adviser)

Dr. Louai Al-Awami (Member)

Dr. Anas A. Al-Roubaiey (Member)

Dr. Ahmad Al-Mulhem
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

29/4/19
Date

To my parents and my wife

ACKNOWLEDGMENTS

First and foremost, I am grateful to Allah for the good health and wellbeing that were necessary to complete this thesis. I would like to express my deepest gratitude to my advisor, Professor Tarek Rahil Sheltami, for his continuous support and guidance of my master study and related research. Also, I would like to thank him for his insightful comments, patience, and motivation. I also would like to thank the rest of my thesis committee for their comments, suggestions, and encouragement. Special thanks go to Dr. Louai Al-Awami for his insightful comments and the different questions which incented me to enrich my research experience from various perspectives. I would like to thank the administration at KFUPM and Najran University for their support at all times. Special thanks go to the College of Computer Science and Information Systems at Najran University for their encouragement and assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DDSSs	Distributed Data Storage Systems
DEC-DS	Decentralized Erasure Code-based Data Survivability
DEC-EaD	Decentralized Erasure Code Encode-and-Disseminate
DEC-EaF	Decentralized Erasure Code Encode-and-Forward
DSwEE	Data Survivability with Energy Efficiency
MAC	Media Access Control
RLNC	Random Linear Network Coding
UDP	User Datagram Protocol
WSNs	Wireless Sensor Networks

THESIS ABSTRACT

NAME:	Mohammed Abdullah Asiri
TITLE OF STUDY:	A Novel Energy Efficient Distributed Storage System for Data Survivability in Wireless Sensor Networks
MAJOR FIELD:	Computer Networks
DATE OF DEGREE:	April 2019

Achieving data reliability in Wireless Sensor Networks (WSNs) applications deployed in a harsh environment is challenging due to resource constraints of sensor nodes. In this regard, Decentralized Erasure Code-based Data Survivability (DEC-DS) for WSNs utilizing Distributed Data Storage Systems (DDSSs) had introduced the notion of data Survivability for WSNs. The fundamental framework of DEC-DS allows for building an erasure code-based storage over WSN according to redundancy in hardware and data, in order to tolerate a given erasure rate. Furthermore, two approaches: Decentralized Erasure Code Encode-and-Disseminate (DEC-EaD) and Decentralized Erasure Code Encode-and-Forward (DEC-EaF) had been proposed to enhance DEC-DS in terms of energy consumption during establishing the code by utilizing Random Linear Network Coding (RLNC). Although these systems have good performance, they have not been *tested under realistic network settings. In this thesis, we conduct an evaluation of these systems over a Zigbee MAC protocol 802.15.4 using OMNeT++ simulation tool with the INET framework. Moreover, we introduce a new efficient energy dissemination approach called Data Survivability with Energy Efficiency (DSwEE) to implement the DEC-DS solution by making use of RLNC. The results confirm that DEC-DS can achieve great reliability for WSNs deployed under harsh work conditions. Moreover, results show that DSwEE achieves better performance than DEC-EaD in terms of energy consumption during building the code.*

ملخص الرسالة

الاسم: محمد عبدالله يحي عسيري

عنوان الدراسة: نظام تخزين جديد وموزع يتميز بالكفاءة في استخدام الطاقة لضمان بقاء البيانات في شبكات الاستشعار اللاسلكية

التخصص: شبكات الحاسب الآلي

تاريخ الدرجة العلمية: ابريل - 2019

يعتبر تحقيق موثوقية البيانات في تطبيقات شبكات الاستشعار اللاسلكية (WSNs) التي يتم نشرها في ظروف عمل صعبة و بيئة قاسية تحديًا بسبب الموارد المتاحة والتي هي محدودة . وفي هذا الصدد ، تظهر قابلية بقاء البيانات المبنية على أساس كود المحو اللامركزي DEC-DS لشبكات الاستشعار اللاسلكية WSNs والتي تستخدم أنظمة التخزين الموزعة (DSSs) لتقديم موثوقية البيانات لشبكات WSNs . يتيح الإطار الأساسي لـ DEC-DS ببناء تخزين storage قائم على أساس كود المحو Code Erasure بواسطة أجهزة التحسس و الاستشعار في الشبكة وفقًا لعدد التكرار والزيادة المطلوبة والمحددة في الأجهزة والبيانات data للتسامح مع معدل محو أو عطل محدد. وعلاوة على ذلك ، تم اقتراح طريقتان هما DEC-EaD و DEC-EaF لتحسين DEC-DS من ناحية استهلاك الطاقة أثناء إنشاء الكود من خلال استخدام تشفير الشبكة الخطية العشوائية(RLNC) . وعلى الرغم من أن هذه الأنظمة تعد بأداء جيد ، إلا أنه لم يتم اختبارها تحت إعدادات الشبكة العملية. في هذا العمل ، قمنا بإجراء تقييم شامل لهذه الأنظمة باستخدام بروتوكول 802.15.4 MAC Zigbee باستخدام أداة المحاكاة OMNet++ مع إطار عمل لمحاكاة برتوكولات الشبكات اللاسلكية Framwork INET . وعلاوة على ذلك ، فإننا نقدم طريقة جديدة (DSwEE) لنشر البيانات مع توفير كبير للطاقة لتنفيذ حل DEC-DS وذلك بالاستفادة منRLNC . وأظهرت النتائج أن مخططات DEC-DS تحقق موثوقية كبيرة لشبكات WSNs خاصة التي يتم نشرها في ظروف عمل صعبة و بيئة قاسية . كما أن النتائج توضح أن الطريقة DSwEE استطاعت التفوق على الآخرين من حيث استهلاك الطاقة أثناء إنشاء الكود.

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks (WSNs)

Wireless Sensor Network is a network infrastructure comprising of specially fabricated nodes that has the ability to instrument, observe and react to events and phenomena in a specified environment [2]. Each node in the network is composed of sensing unit, computing (processing) unit, memory, and wireless communication elements. The environment of WSN operation can be the physical world, a biological system or an information technology framework etc. WSNs in general are made up of four basic components [2]:

1. An assembly of distributed or localized sensors.
2. An interconnecting wireless network.
3. A data collector or base station (central point) of information clustering.
4. A group of computing resources at the central point or beyond to handle data collection, event trending, status querying and data mining.

Sensors are deployed in specialized domains called sensor fields. These venues of deployment can vary from large fields like battlefields or a river bed to small fields like a human organ.

1.1.1 Node Architecture

A sensor node has a processing module with embedded processing capabilities and an internal memory for temporal storage of sensed or relayed data. Also embedded on a node are sensors collecting data of radio, acoustic, chemical, magnetic, optical or infrared phenomena. A node has wireless communication unit for inter-node or node to sink communication. A node also has a power source coupled with a power processing unit to provide the required power to node components. Based on application, nodes may also have position or location knowledge through GPS or some local positioning technologies. A typical node architecture is depicted in Figure 1.1.

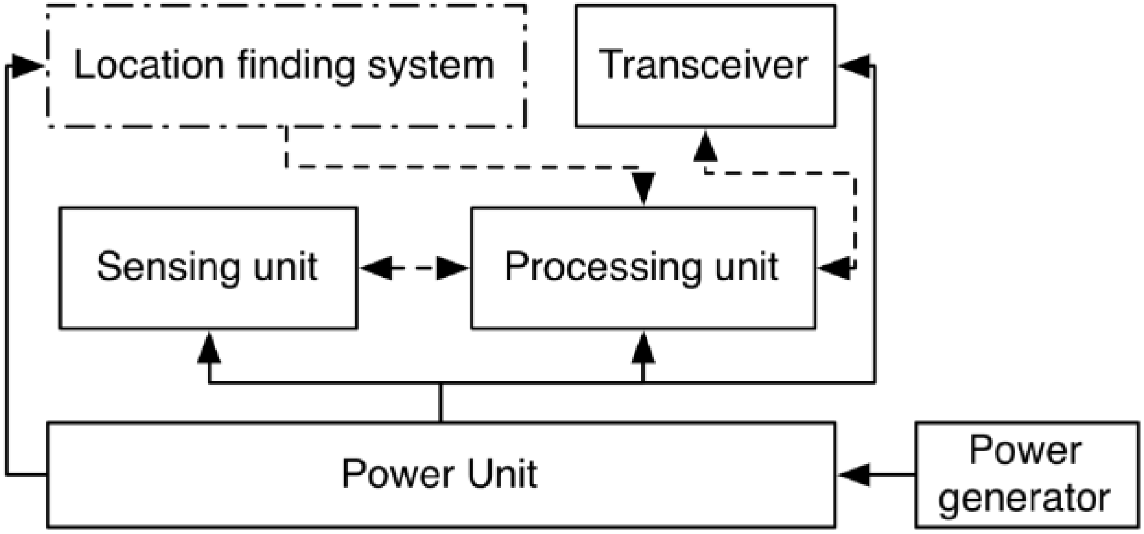


Figure 1.1: Typical node architecture

The processor subsystem is the focal component in a WSN node and the decision of a processor indicates the trade-off between adaptability and productivity which is identified with vitality and execution. The processors have numerous parts which include: microcontrollers, computerized signal processors, application-particular coordinated circuits, and field programmable gate arrays (FPGA). The sensing component is composed of larger than one analogue sensors. These sensing equipments are analogue or digital systems for reading the sensor values. Most of such sensors contain their locally fashioned analog-to-digital converter (ADC) which is able to directly interface sensors with the processing unit using a standard chip-to-chip protocol. Several microcontrollers/processors are composed of one or more internal ADCs to connect to analogue devices. Recent microcontrollers integrate flash storage, RAM, ADC, and digital I/O onto a single integrated circuit. In choosing a microcontroller family, several factors have to be taken into consideration such as energy consumption, support for peripherals, voltage requirements, cost, and number of external components required.

The communication component of a node is typically connected to the processing unit using the serial port interface (SPI) bus. The communication subsystem is the most energy intensive component and its operation has to be managed so as to conserve power. Several of the market available transceivers give a controlling functionality to alternate the transceiver between different operation levels such as active, idle and sleep state.

The power subsystem gives a supply of direct current (DC) power to all node units and their components. This unit is composed of the energy storage component, voltage regulator, and sometimes the energy scavenging component. Node power is usually obtained from a battery-pack. Moreover, other components could be used in generating power for the sensor nodes so as to extend the lifetime of the network such as solar energy storage qualities.

1.1.2 Network Architecture

Each sensor node deployed in a sensor field has the capability to collect, analyze and route field data to a designated sink point. Some nodes in the field may be built to perform specialized functions in addition to the above. A typical WSN node arrangement is as shown in Figure 1.2. Wireless sensor networks are normally deployed in different network topologies where the choice of a topology normally depends on the application. Typical topologies are grid, random, ring,linear and star.

1.2 Distributed Data Storage Systems

DDSSs form a crucial part of most adopted technologies of networks. Companies and website and Internet service providers employ DDSSs for storing and handling large amount of data, the so-called *big data*. Key merits of DDSSs in networks are robustness to faults and the ability to be scaled in or out. Robustness to faults and scalability are normally attained by replicating data over several network nodes. Although replicating data is simple, this strategy is generally in effective in terms of storage requirements and energy consumption. Hence, coding-based DDSSs present a more attractive choice as compared to replication due to their performance in terms of the previously metrics.

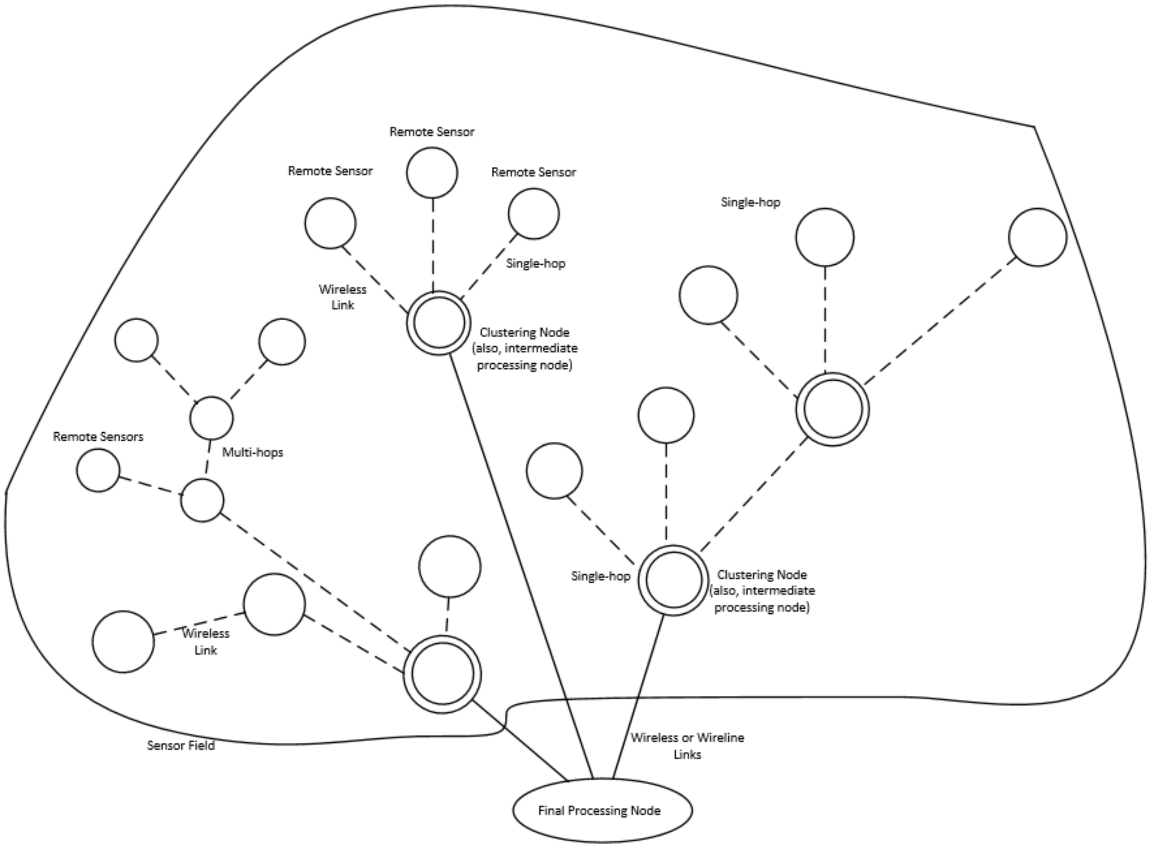


Figure 1.2: Typical sensor network arrangement

To fashion a storage system, two methods of redundancy are normally adopted to obtain a reliable data storage: replication and coding. In methods where coding is employed, several merits can be harnessed over methods employing replication by trading-off a little cost in processing. Compared to methods using codes, methods of replication normally needs huge memory requirements on each network node. That is, to get a similar reliable DDSSs based on replication normally demands a high level of redundancy than as compared to methods using codes. Factually, using similar redundant strategies, methods using codes can attain an order of magnitude in reliability far larger than methods using replication [3]. Further, replication based methods additionally are required to track the locations of unequal data, which requires sophisticated protocols for information gathering.

Also, some analytical works indicate that on average, the number of data blocks required to remake a entirely finished data set from a method using replication distributed storage is higher compared to that required when employing coding based distributed storage [4]. Coding-methods have further been reported to require lower storage requirements and simpler methods of gathering data [3].

DDSSs are also classified using the positions of the origin or source relative to the encoded-data, which can be either centralized or decentralized. Centralized here is used to refers to the case where data exists in the same physical location. The term decentralized, on the other hand, means data existing in a myriad of physical locations.

1.3 Topologies

Due to the instrumentation and monitoring nature of WSNs, the positions of the nodes with respect to each other and a gateway in the sensor field is a crucial criteria that determines performance metrics like network power consumption which determines network coverage and lifetime [5]. For example the network topology is a key factor in time synchronization since the relative position of a node to its neighbors affect the paths chosen in synchronization algorithms for time update. Several topologies exist for WSNs. Common amongst these topologies are the star, ring, grid, random and linear topologies. Figure 1.3 presents a general picture of these topologies for a four node network.

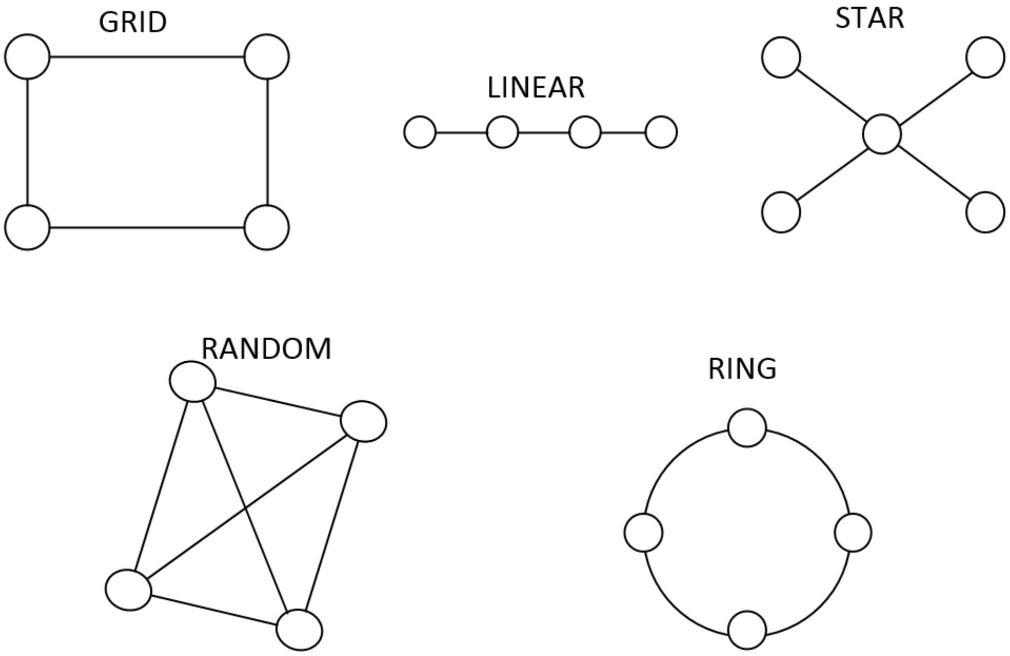


Figure 1.3: Typical WSN topologies

1.4 WSN Communication Protocols

In general, sensor nodes store and/or process the data that is collected from the sensor field. The sensed data is then transmitted to a base station or a sink node in a centralized network, or can go under processing rather than transmitting directly to the base station as in distributed networks. Several types of communication channels such as microwave, radio links and satellite links can be used for transmitting and extracting information obtained from the wireless sensor networks.

When discussing WSN specifications and solutions, it is helpful to understand the structure of communication protocol stacks. A protocol stack defines a set of layers, where each layer is a collection of related functions. A layer offers services to the layer above it, and uses services from the layer below. The most common communication stack model is the seven-layered OSI-Model. For WSNs, a simplified version of the OSI model is used, where the Presentation Layer and the Session Layer are not defined [12]. Note that not all WSN standards define the Transport Layer either.

1.4.1 IEEE 802.15.4

The IEEE 802.15.4 [6] was initially released in 2003 and updated in 2006. The standard comprises four different physical layers (PHYs), three in the 868/915 MHz band and one in the 2.4 GHz band. A total of 27 channels are defined, numbered from 0-26. Channel 0 is in the 868 MHz band, Channels 1-10 are in the 915 MHz band and channels 11-26 are in the 2.4 GHz band. In the 2.4 GHz band the channel width is 5 MHz and the channel spacing is 2 MHz. As the 868 MHz (Europe) and 915 MHz (US) bands have limited geographical availability due to various national rules and regulations, most industrial applications uses the globally available 2.4 GHz band.

1.4.2 WirelessHART

WirelessHART is a part of the HART Field Communication Specification, Revision 7.0 [6], which was ratified in September 2007. WirelessHART enables wireless transmission of HART messages, and was the first standard to be released which specifically targets industrial applications. WirelessHART was approved as IEC standard 62591 in 2010. WirelessHART is based on the IEEE 802.15.4 PHY and Media Access Control (MAC), although the MAC has been modified to allow for frequency hopping.

Furthermore, WirelessHART only operates in the 2.4 GHz band, so global availability is allowed . For the channel access method with a full mesh network topology, TDMA with frequency hopping is used.

WirelessHART offers self-configuring and self-healing multi-hop communication.

1.4.3 ISA100.11a

The ISA100 standards committee of ISA aims to deliver a family of standards for wireless systems for industrial automation. ISA100.11a [11] was the first standard to emerge, being ratified in 2009 and updated in 2011. For secure and reliable wireless communication, SA100.11a is designed and that helps in non-critical monitoring and control applications. ISA100.11a is based on the IEEE 802.15.4 PHY and MAC, but the MAC has been adopted to allow for frequency hopping and extended security mechanisms. ISA100.11a only defines operation in the 2.4 GHz band. TDMA with frequency hopping is used as the channel access mechanism. Both routing and non-routing devices is supported in ISA100.11a where network topologies can be either star, star-mesh or full mesh depending on the configuration and capabilities of the devices in the network. IPv6 traffic and routing is also integrated supported in the network layer .

1.4.4 ZigBee / ZigBee PRO / ZigBee IP

The ZigBee specifications, were first introduced in 2004 and were further upgraded in 2006 and 2007. This wireless communication protocol has a low data rate and energy consumption and was designed by the ZigBee Alliance [7]. In the specifications of

Zigbee, the application and network layers are defined on top of the physical and Mediun Access Control (MAC) layers of IEEE 802.15.4-2003, and it's main target applications are smart grid, home automation and consumer electronics applications. Since ZigBee specification uses the physical and MAC layers of the IEEE 802.15.4, they have the same modulation techniques, bandwidth and channel configurations [8]. A ZigBee network communicates using the same, user defined channel throughout in its operation lifetime. This gives it a vulnerability in terms of interference from nearby networks communicating on a similar frequency and also susceptible to noise from different signal sources in its sensor field. Hence Zigbee is not usually employed for applications requiring robust communication protocols like in harsh industrial sensor fields [9]. To deal with this issue, the ZigBee Alliance upgraded the first variant to the ZigBee PRO specification [8] in 2007. ZigBee PRO is designed such that, it will perform effectively in industrial settings with added features like, improved security and agility towards channel noise and network frequency interference. In this protocol, the phenomena of frequency agility is added where the entire network has the capability of altering its operation channel frequency when it encounters large amounts of noise and/or interference. Despite these additions and upgrades, ZigBee has not seen a wide industrial adoption. The ZigBee Alliance pronounced in April 2009 to incorporate standards from the Internet Engineering Task Force (IETF) into future ZigBee releases, thereby opening up for IP-based communication in ZigBee networks. Of special interest for the ZigBee Alliance is the 6loWPAN working group which has created a Request for Comments (RFC4944) investigating the transmission of IPv6 packets over IEEE 802.15.4 networks. This work resulted in the ratification of the ZigBee IP specification in February 2013 [10]. Although Zigbee is not popular in industrial applications, it is widely used in small electronic equipments, in small scale commercial applications and in scientific experimental inquiry. Most wireless sensor nodes use Zigbee protocol or it's variants for communication in the sensor field. Zigbee is also widely used in network architectures and simulators to test the performance of newly designed or proposed schemes of WSN network establishment and management services such as routing, time synchronization, localization and congestion control protocols.

1.5 Wireless Sensor Network Simulators

Wireless Sensor Networks (WSNs) are being employed in many critical applications such as intrusion detection, object tracking, industrial/home automation, smart structures and several others. The development of a WSN system requires that the design concepts be first checked and optimized using simulation [11].

There are two types of the simulation environment for WSNs, an adaptive development or a new development. A new development is created according to specific characteristics of sensors from the beginning. In contrast, adaptive development has simulation environments that were created for specific purposes before the technology of WSNs existed. Later, they were extended to support the functionality of wireless and adapted for WSNs[12]. Several simulation tools have been developed recently to specifically address WSNs such as NS2, NS3, Contiki, Cooja and Castalia [13], varying from extensions of existing tools to application-specific simulators.

These tools differ in design purposes, architecture, and applications abstraction level although they have some collective objectives, [11].

According to the level of complexity, simulators can be categorized into three major divisions :

- Algorithm level,
- Packet level, and
- Instruction level.

Among many types of research, the network-related research simulations have very popular. Several simulators have been developed to implement and study algorithms for wireless networks. Some are designed for specific purposes while others have general goals. They differ in the level of complexity and features. They support certain hardware and communication layers assumptions, and provide a set of tools for deployment scenarios, modeling, analysis, and visualization. Classical simulation tools include NS-2/3, OPNET, Contiki, OMNeT++, J-Sim, and TOSSIM [12][14][13].

1.5.1 OMNeT++

OMNeT++ is a discrete event simulation environment. It has been primarily designed for simulation of communication networks as a primary application area. However, it has also been successfully used in other areas such as the simulation of queuing networks, architectures of hardware, etc.,and that was a result of its generic and flexible architecture [15][16].

OMNeT++ provides a component architecture for models. C++ is used to program these components (modules) and a high-level language (NED) is used to assemble larger components. The accessibility of the models is easy and availability for free. This modular architecture for OMNeT and its simulation kernel makes it easily to embed into applications. OMNeT is also provided by GUI. Although OMNeT++ is a simulation framework (not a network simulator itself), it has currently achieved widespread popularity among the scientific community as a network simulation platform. It is also used in industrial settings. It has gained a large user community [15] [17].

1.5.2 INET Framwork

OMNeT++ is an object-oriented modular discrete network simulation framework. Although it is a powerful tool for WSNs , it does not have network protocols like IP or TCP. So, several external frameworks with models of main computer network simulation are combined with OMNeT++. INET is the most commonly framework used with OMNeT++. Researchers in communication networks are provided with all kinds of network layers, protocols, and technologies that are available in INET. INET helps in creating and validating new protocols or evaluating the performance of new algorithms. The Internet stack models, wired and wireless link layer protocols, a set of mobility models for the node movement simulations and many other protocols and components are embedded in INET. INET has been used as a base to develop other simulation frameworks for other areas such as vehicular networks and LTE. [18] [19] [20] [21].

1.6 Motivation

Nodes in WSNs are deployed in complex environments, and are, most often, supplied with energy using batteries. These batteries are a problem due to their limited supply. Achieving reliability by preserving the data sensed by sensor nodes in WSNs is a critical requirement especially when nodes deployed under harsh conditions. Traditionally replication data methods have been employed to achieve data reliability, but without consideration of WSNs limited resources in terms of memory and energy consumption. Replication data approaches require more redundancy and complicated operation to track data and collect them [3]. Coding methods have been suggested to solve these problem [22]. Key amongst these methods are decentralized codes where storage has been shown to be robust and efficient [4]. The authors in [23], [24] proposed a decentralized erasure based method for data storage in wireless sensor networks, which is characterized by a simple and decentralized building of the target code. Furthermore, to achieve data reliability for WSNs applications, distributed data storage systems (DDSSs) are utilized by increasing storage devices and replicating data packet over these devices and if some of them fail, the data collector can gather data from remaining devices that survive. For the same purpose, The proposed schemes in [23], [24] designs DDSSs that introduce data survivability notion by linking a redundancy value required to be tolerated with the expected maximum of erasures or failure. DEC-DS is proposed to optimize the required redundancy to achieve the data survivability. While DEC-DS achieves the data survivability, it needs enhancements in terms of energy consumption. Decentralized Erasure Code Encode-and-Disseminate (DEC-EaD) and Decentralized Erasure Code Encode-and-Forward (DEC-EaF) schemes were proposed to solve this energy issue by exploiting the Random Linear Coding opportunities where relay nodes participate in building the code. DEC-EaD scheme is a completely decentralized solution that uses a rotor-router random walk model to disseminate data. DEC-EaD was shown to be much better than DEC-EaF in terms of energy consumption.

In this regard, in this work, we introduce a new decentralized dissemination scheme for energy efficiency by exploiting RLNC in the shortest path routing setup for enhancing DEC-DS.

Although a somewhat detailed theoretical and simulation-based performance evaluation is carried out in this work [23], [24], evaluations are done only on simulation frameworks where system conditions are minimal. A comprehensive performance evaluation of the DEC-DS, DEC-EaD schemes presented in [23], [24] and our a new proposed scheme are carried out where communication is done using the widely adopted Zigbee MAC protocol IEEE 802.15.4. Further, the evaluations are performed on extensive simulations where many practical network settings are present. The performance metrics are the code survivability and the energy consumption required to construct the code.

1.7 Objectives

The main objectives of the study are to:

1. Conduct a comprehensive literature review on the DEC-based DSSs for data survivability in WSNs.

2. Conduct a thorough analysis of Decentralized Erasure Code (DEC) based schemes with Zigbee MAC protocol 802.15.4 using the OMNeT++ simulation tool with the INET framework.

3. Developing a new efficient energy dissemination scheme to disseminate a source data and utilize coding opportunity to save energy of nodes during building code.

1.8 Contributions

Distributed Storage Systems using DEC-DS are shown to ensure data survivability in a WSN at a smaller redundancy cost and reduced energy consumption in comparison to original DEC. Although these systems promise a good performance, they have not been tested using practical network settings. The main contributions of our thesis are:

1. Testing DEC-DS and DEC-DaD systems using a network simulator over Zigbee MAC protocol IEEE 802.15.4. We evalute the performance in terms of two metrics: code survivability and energy consumption.

2. Implementing a new energy efficient dissemination scheme, called ”Data Survivability with Energy Efficiency (DSwEE)”, that uses the shortest path in disseminating source data in a decentralized fashion with energy efficiency by utilizing network coding while routing.

3. Evaluating the performance of DSwEE against both of DEC-DS and DEC-EaD.

1.9 Thesis Outline

We organize the outline of this dissertation as follows. Chapter 2 introduces the DEC-based DDSSs for data survivability in WSNs. Furthermore, a survey of a myriad of studies on developing several variants of the original DEC system is conducted in this chapter. In Chapter 3, we introduce a new efficient energy dissemination technique for building the DEC-DS taking advantage of the concept of the network coding called Data Survivability with Energy Efficiency (DSwEE). This chapter presents how DSwEE works and illustrates with a detailed example step by step. Next, Chapter 4 presents a performance evaluation of the proposed DSwEE with the previous schemes DEC-DS and DEC-EaD in terms of code survivability and energy consumption under WSNs MAC layer 802.15.4 using the OMNeT++ simulation. Finally, conclusion and future work are discussed in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we introduce the DEC-based DDSSs for data survivability in WSNs. Furthermore, we discuss the essential and important information about each scheme including the main features. Moreover, a survey of a myriad of studies on developing several variants of the original DEC system is conducted in this chapter.

2.1 Data Survivability and Network Survivability

Sensed data generated in WSNs needs a reliable mechanism in order to be maintained, especially when there is no sink node as in Delay Tolerant Networks(DTNs). In this regard, the notion of data survivability comes by designing a parameter that provides the resiliency for sensed data against failures [23], [24] .The differences between network survivability and data survivability are in terms of using redundancy for why and where. While redundancy is being used in network survivability for keeping the continuity of the network during node failures [25], it is in data survivability for achieving the data surviving against loss during such node failures. In network survivability, redundancy is in hardware and software but for data survivability, it is in storage devices and data.

2.2 Decentralized Erasure Codes (DEC)

Decentralized erasure codes were first proposed by Dimakis *et al* in [4]. To explain the basic idea behind DECs, as shown in Figure 2.1, consider a wireless sensor network. Assume the network has k nodes acting as source and n nodes acting as storage nodes, where $k < n$, every node j acting as data source produces a single data block b_j and sends b_j to m storage nodes. The selection of the m storage nodes are carried out in a uniform random manner. When a storage node i receives b_j, it creates a random coefficient g_{ij} for each b_j received, which is taken from a finite field $\mathbb{F}_2$ for each block of data that every i receives and combines the received blocks as $e_i = (g_{i1}b_1) \oplus (g_{i2}b_2) \oplus (g_{i3}b_3) \oplus \cdots \oplus (g_{ik}b_k)$

where $\oplus$ represents the *xor* operation..

In each storage node i, the block of encoded data are saved as e_i. The vector, G_i used for encoding is also saved. Where $G_i = \{g_{i1}, g_{i2}, g_{i3} \cdots g_{ik}\}$. The decoding system retrieves back the data from the original blocks by collecting $(1+\epsilon)k$ where $\epsilon > 0$ blocks of encoded data. These encoded blocks are used to get the solution for B in the system of linear equations represented by $E_{1\times n} = B_{1\times k}G_{k\times n}$. The vectors E and B denote the the vector of native data and the matrix of encoding coefficients respectively. DEC methods have been shown to be more robust storage and recovery methods as compare to traditional schemes of redundancy such as replication. Several variants of

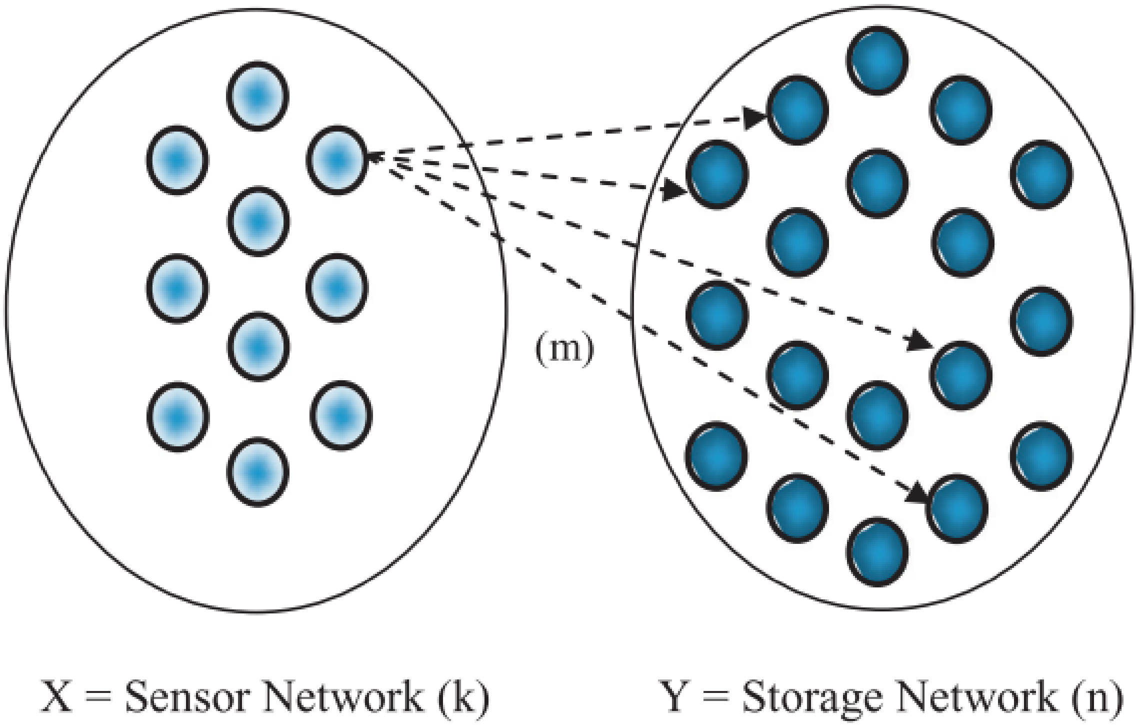


Figure 2.1: Source and storage sensor networks [1]

Decentralized Erasure Code (DEC) have been proposed to ensure data survivability in Wireless Sensor Networks. In [26], the authors constructed DEC based system for local optimum recoverable codes and analyzed it's performance through simulations.

In the following section, we present the distributed data storage methods proposed in [24], [23]. The descriptions given for each method here is a summarized operation mechanism. The network model and system definitions used for the description of these methods is given in the next section.

2.3 Decentralized Erasure Codes for Data Survivability (DEC-DS)

This method produces redundancy in accordance with a supplied data survivability value denoted as s and is designed to increase data survivability against failures or erasures.

This proposed method was design to operate in a similar manner to the original decentralized erasure code with the only difference being, in the Redundancy Factor, m generated by the nodes acting as sources. The main operational goal of this method is ensuring the recovery of data when there are at least $(1 - \frac{1}{1+s})n$ packet existing in the network whereas in the original decentralized erasure code, the main goal is to ensure that any $(1+\epsilon)k$ encoded packets suffices to recover the original packets, where n denotes the number of storage nodes. Mathematically, DEC fashions the generator matrix G such that any $(1 + \epsilon)k$ rows are linearly independent whereas DEC-DS is able to construct an invertible generator matrix, so long as there exist $(1-\frac{1}{1+s})n$ rows.

2.3.1 Network and Code Model

A set of k source nodes A=a_1,a_2,...,a_k and a set of n storage nodes B=b_1,b_2,...,b_n are formed a network such as the one in Figure 2.1. Each sensor node generates a data packet x_i, also called native packet. All native packets generated by all source nodes are represented as a set of X=x_1,x_2,...,x_k. Sensor nodes selects randomly and uniformly m storage nodes Y_i= y_1,y_2,...,y_m where $Y \subseteq B$, and sends its x_i to them, the number of those sent out data packet is refered as Redundancey Factor (RF). When a storage node y_i receives x_i, it creats an entry of what called encoding vector G_j=g_{i1},g_{i2},...,g_{ik} g_{ij}=1 for each x_i received, which is taken from a finite field $\mathbb{F}_2$. If x_i is not rceived by a storage node y_i, g_{ij} =0. Then, storage node y_i combines the received blocks linearly as $e_i = (g_{i1}x_1) \oplus (g_{i2}x_2) \oplus (g_{i3}x_3) \oplus \cdots \oplus (g_{ik}b_k)$, where $\oplus$ represents the *xor* operation. The result is e_i refered as an encoded block which is stored with its corresponding encoding vector in a storage space for each storage node. In a data gathering stage, A data collector gathers data from a subset of storage nodes which forward their encoded blocks with its corresponding encoding vector. Then, data collector builds a matrix G^{-1} generated locally from the received encodig vectors. Also, using the recived encoded blocks, the E encode data matrix is generated locally. Finally, the native packets X are recoverd using the generated equations system X= G^{-1} E .

2.3.2 Data Survivability Design

Survivability is defined as the maximum possible failure fraction of sensor nodes without compromising the retrivability of the native data packets. This fraction is expected to be less than or equal to $f \times n$, where $f = \frac{s}{s+1}$. DEC-DS is designed to achieve survivability s by determining the number of n storage nodes in the network as n= k (s+1). It is also designed to achieve survivability s by building the encoding vectors G to be reversible with the high probability level even though there is a deletion of $f \times n$ rows. It was proved the following : a $k \times n$ random matrix over a finite field $\mathbb{F}_2$ is referred as G, and $s \geq 0$. n is calculated as n= k (s+1) and m = (1+s) (log(k)

$+c_1) + c_2$. G is built by selecting m entries in each of the n columns and setting them to 1. a $k \times k^{'}$ matrix is referred as $\overline{\text{G}}$ by selecting n-$k^{'}$ rows from G in a uniformly random fashion , where $k^{'}$ as $k \leq k^{'} \leq n$-k. For some constants values for c_1 and c_2 , the $\overline{\text{G}}$ is invertible where 7 and 8 are chosen for c_1 and c_2 ,respectively based on the experimental values explained in [23], [24]. So, DEC-DS works as

1. n is calculated as

$$n = k(s+1) \tag{2.1}$$

2. Each sensor node generates a data packet x_i

3. Each sensor node selects uniformly and randomly

$$m = (1+s)(\log(k)+7)+8 \tag{2.2}$$

   distinct storage nodes and sends a copy of x_i to the selected nodes.

4. Each storage node manipulates x_i as explained above.

2.4 Performance Metrics: Data Survivability and Energy Efficient

2.4.1 Data Survivability

The term data survivability is used to describe the redundancies introduced in an algorithm that prevents the loss of data or ensures the recovery of data in case of node, network and/or transmission failures. This metric is crucial in evaluating an algorithm's performance in comparison to other distributed data storage algorithms. To evaluate the survivability of the code, the the erasures are modeled by deleting a node from a network as a failure node. It is a random process where each time unit t_i, a value between 0 to 1 of the erasure rate f is created, and then $f \times n\ f$ uniformly and randomly selected storage nodes are erased. After that, the data survivability evaluation for the portion of the remaining code is implemented.

2.4.2 Energy Efficiency

Energy efficiency in the context of WSNs can be referred to as the amount of joules expended by a sensor node in sensing, processing, transmitting and receiving data packets when executing a algorithm(s) or protocols. The processing and executing of algorithms are usually the function of the number and type of operations executed by the node. Whereas the transfer of data packets depends on the distance between neighborhood nodes, the size of packets and the data transmission rate. The amount of energy consumed by the network is also a function of the total number of hops over which messages are sent. Since most sensor nodes are powered by batteries this energy must be conserved. Also, since encoding is the integral part of processing in DEC algorithms, the energy used in encoding has to be considered. Therefore, taking the number of hops into consideration, the energy components of the total energy are, the energy: for sensing E_s, for transmission E_t, for reception E_r and for encoding E_e, where $E_s, E_t, E_r \gg E_e$. Although there is energy expended in routing, it is normally neglected because its small. The central data collecting node is assumed to have infinite energy.

2.5 Energy Efficient Decentralized Erasure Codes Based Distributed Storage

Relay nodes are normally only used for routing and forwarding in several of the proposed DEC schemes. When data is being disseminated and after a source packet is produced, selecting a set of candidate storage nodes, and forwarding the packet by source nodes, relay nodes are used to send source packets to storage nodes without altering them. Since communication cost in terms of energy is much higher than processing cost, the authors in [24] use coding techniques during packet relaying to make manipulations to achieve a more energy efficient process of dissemination. This is done by making each relay node to partake in the encoding process in the period of dissemination.

Hence they show that although the Decentralized Erasure Code (DEC-DS) is shown to require less energy consumption as compared to the original Decentralized Erasure code (DEC), it does not make use of coding opportunities present when packets are being relayed. This advantage was used to fashion two new schemes which ensure energy efficiency. The key logic in designing these new schemes is to make use of coding opportunities while packets are being forwarded. These schemes are named the Decentralized Erasure Code Encode-and-Forward (DEC-EaF) and the Decentralized Erasure Code Encode-and-Disseminate (DEC-EaD).

2.5.1 Decentralized Erasure Codes: Encode-and-Forward (DEC-EaF)

For the first method named Decentralized Erasure Codes: Encode-and-Forward (DEC-EaF), in every stage of the coding scheme, a destination storage node is selected in a stochastic manner by the source node, and the source packet is sent accordingly in a multi-hop way. Then, each relay node that relays transmitted packet, the relay node adds the packet obtained with the locally encoded saved packets prior to sending the new packet to the next hop. In cases where there are no local packets, the relay node stores a replica of the relayed packet. It assumed that the source node has multiple paths for each target node. Selection of target node in this scheme is done in a random manner whereas the choice of routing path is not done stochastically but depends on a routing layer in choosing the shortest route in reaching the destination storage node.

DEC-EaF can be used in applications where networks carryout or selects routes using a routing layer.

2.5.2 Decentralized Erasure Codes: Encode-and-Disseminate (DEC-EaD)

In the second scheme, named Decentralized Erasure Codes: Encode-and-Disseminate (DEC-EaD), dissemination of native packets by source nodes is carried out by employing a node-centric random walk operation. Since the choosing of destination nodes is randomly carried out in the original DEC-DS, it seen logical to employ random walk to remove the need of creating and maintaining routing tables.

DEC-EaD is shown to be completely decentralized [24] and hence is based on random walk. DEC-EaD is also shown to be much better than DEC-EaF and DEC-DS in terms energy consumption.

2.6 Review of other DEC-Based Distributed Data Storage Schemes

Decentralized Erasure Codes (DEC) schemes sets themselves apart from other coding schemes by several merits. Key among these merits is, in DEC schemes the need for centralized coordination between data nodes is removed. Therefore, randomly distributed nodes acting independent of each other can generate a effective sparsely distributed code structure and ensure a high degree of reliability as a storage scheme. Due to the strength of DEC schemes, a myriad of studies has gone into developing several variants of the original DEC system in [4]. In this section, we review some of these schemes.

In [27], a systematic Reed Solomon based erasure codes is employed to achieve reliable distributed data storage in sensor networks. The authors demonstrated that when any l packets obtained out of m at a receiving node is enough to regenerate the actual packets sent from data collecting sensor nodes in a back-to-back data transmission scheme. In this work, a real time implementation using Mica2Dot Berkeley motes was carried out. These nodes operate on a 310 MHz multichannel radio wireless network. The implementation was done on a TinyOS platform. This method of evaluation however did not account for all packet losses and was not evaluated over a standardized wireless protocol.

Ali et. al., [28] employed Reed Solomon codes whiles considering multi path diversity in wireless sensor networks for reliability in the exchange i.e., sending and receiving of information in a sensor field among nodes of interest. The information gathered at each sensor node was transmitted as packets to a centralized central collection point using proxy nodes labeled ”prongs”. The central collector then reconstructs the message and retrieve the information, when it receives for example, l out of m transmitted packets. In this scheme, the method used for evaluation by the authors was numerical. Hence the wireless channel used for evaluation was theoretical. The protocol therefore needs to be evaluated in a real time simulation or experimental environment with a widely used wireless protocol.

In [29], Marchi et. al. proposed DTSN: Distributed Transport for Sensor Networks. This scheme employs unicast methodology in attaining reliability in transmission in a WSN. The authors suggest a technique composed of two parts. The first technique uses a cache between intermediary sensor nodes located between the data source and destination to shrink to a minimum, the number of retransmission and therefore conserving node resources. The second technique involves the use of Reed Solomon Erasure codes for the transfer of data for applications of non-critical nature. The system was shown to present a certain level of reliable storage as compared to others. DTSN was evaluated on OMNET++ discrete event component based simulator. But the simulator used was a first generation version which did not have a transport layer congestion control protocol. This method was however evaluated in a TinyOS environment using IST Ubisec platform. These node are however not equipped with a standardized wireless protocol.

Srouji et. al. [30] proposed RTSN: A reliable erasure-coding based data transfer system for WSNs using multi-hop transmission. The authors suggested a method that uses erasure codes at hops that is identical to network coding so carry out hop encoding and decoding and partial coding that realizes data reconstruction using enough packets. The main achievement of this work is a significant increase of network lifetime and a small coding overhead. The NS2 network simulator was used for the evaluation of this scheme. The standard IEEE 802.11 protocol model was used for the evaluation. This method was however not evaluated on a real time experimental environment.

Kumar et.al. [31] suggested the FBcast method. This method is able to attain reliability of data tranfer in WSNs by employing a rate-less coding inspired by fountain codes to ensure reliability. This method mainly concerns itself with the upgrade of the core code or firmware installed in WSNs. Since this method employs rate-less coding, its main strength is the variation of message length of encoded packets so as to maximize reliability. FBCast was shown to offer limited data confidentiality and highly reliable. The TinyOS Simulator (TOSSIM) was used for the evaluation of this protocol. Inusing TOSSIM, the authors could not implement data encoding/decoding and had to use an extrapolation of fountain coding based on experience.

Kumar and Selvakumar [26] developed a DDSS system by constructing a decentralized erasure code setup from the optimum localized recoverable codes. This method was shown through extensive simulations to minimize the load on the network and increase system efficiency by regenerating data at the lost nodes in the sensor field. This method was evaluated using MATLAB. This platform is not a real time simulator like NS2 or OMNET++ and does not account for several network conditions for simulations. Hence there is a need for an evaluation on a standard MAC protocol.

CHAPTER 3

DATA SURVIVABILITY WITH ENERGY EFFICIENCY (DSWEE)

In this chapter, we present our enhanced scheme DSwEE, titled ”Data Survivability with Energy Efficiency (DSwEE)”, to implement DEC-DS with saving energy. We show the difference between DEC-EaD and our scheme DSwEE. We show how DSwEE works and illustrate that with a detailed example step by step. In the same way, the operation of DEC-EaD is showed in order to illustrate the differences between DEC-EaD and DSwEE. Comparing DSwEE through a network simulator with DEC-DS and DEC-EaD in terms of the energy consumption to build the code for different sizes of networks and the code survivability will be presented in chapter 4.

3.1 Prelimnaries

DSwEE uses a routing strategy that supports the shortest path to a destination in the network. RLNC provides coding opportunities that help in saving the required energy to building code. This feature is achieved by making relay nodes participants in coding process as it is done with DEC-EaD and DEC-EaF schemes. In the same way, it is suggested in our scheme DSwEE to reduce the required energy for establishing a code. In DSwEE, data is disseminated in a completely decentralized fashion as DEC-EaD, but the difference is in the technique that is utilized to do a data dissemination operation. DEC-EaD uses the rotor-router model, which is a quasirandom analog to the random walk method while DSwEE utilizes a shortest path routing table for the relay forwarding. Each node in DSwEE has the shortest path routing table to all destinations in the network. DSwEE uses the same fundamental model and routing technique that the original DEC-DS uses, but the difference is in the dissemination phase of establishing code where DSwEE tries to save the energy during that process. The main differences are shown in the next section.

3.2 Data Survivability with Eneragy Efficiency (DSwEE)

First, a source node chooses a target storage node randomly in a uniform fashion and then forwards the data source using the shortest path technique. Unlike DEC-DS, all hop nodes in the path manipulate the received data packet by combining it linearly with the previous encoded data packet that stored before this newly received data packet arrived. A Redundancy Factor, m is embedded in the packet to track the distribution of the coding establishment, by decreasing m by one. This decreasing for m is performed only when the newly visited node receives the packet. The packet forwarding process is stopped when m reaches zero. When a forwarded packet arrives a randomly selected storage node (destination), and m is non-zero, a destination node will update m . Then, it selects a new destination storage node uniformly at random, and the same steps continue until m reaches zero. Algorithm 1 and 2 show our mechanism used at respectively, source nodes and storage nodes

Algorithm 1 DSwEE (Source Node)

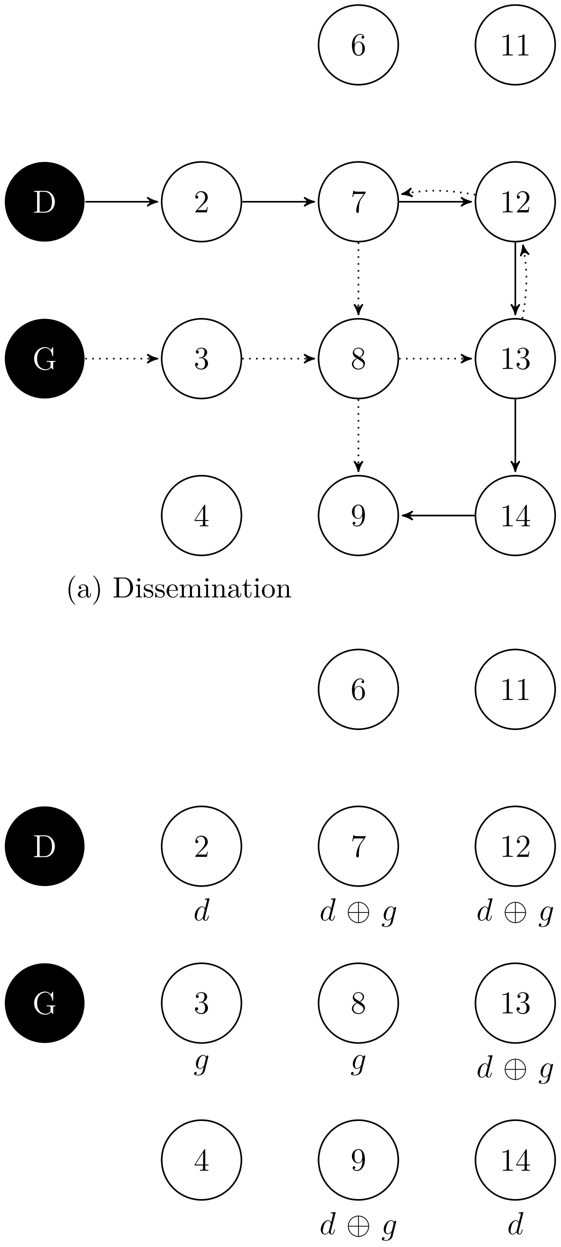
```
Generate a packet $x_i$.
Give a value for $m$, according to the equation 2.2.
Select a random storage node at a uniformly random fashion.
Forward $x_i$ to the next hop in the shortest route.
```

An example shows how DSwEE works in Figure 3.1. We assume that the source nodes are D and G while the storage nodes are the numeric ones. First, node D selects node 12 as a destination storage node. Then, the source data is forwarded using the shortest path through 2 and 7. The value of m is decremented only by all newly visited nodes. After the packet arrived destination 12 and the value of m is non-zero, node 12 selects a new destination storage node 9. Finally, the packet arrives 9 through 13 and 14 and the value of m is updated. Since the m value reaches zero, the packet dissemination stops. Node G applies the same logic by selecting a

Algorithm 2 DSwEE (Storage Node)

```
 1: if $x_i$ already received by (Revisited Node) then
 2:     Forward $x_i$ to a next hop
 3: else if $x_i$ is received by next hop then
 4:     if $g_i$=0 then
 5:         Generate a new coefficient $g_{ji}$ ;
 6:         $e_j$=$e_j \oplus (g_{ji} \times x_i)$ ;
 7:     end if
 8:     Update $m = m$ - 1 ;
 9:     if $m > 0$  then
10:         Forwards $x_i$ to a next hop
11:     end if
12: else
13:     (Destination receives $x_i$)
14:     Apply the same operations in Algorithm  2 from Operation 4 to Operation 9
15:     Select a new random storage node at a uniformly random fashion.
16:     Forward $x_i$ to the next hop in the shortest route.
17: end if
```

new destination storage node 13 and disseminating a source data through 3 and 8. After the value of m is updated and then its value is non-zero, the destination 13 selects 7 as a new destination through 12. Since the value of m is non-zero the node 7 selects 4 as a new destination through hops 8 and 9. As the node 8 is revisited by the same data source, The value of m remains unchanged. Because the value of m reaches zero, the dissemination process stops at hop 9 and the source data is not forwarded to a destination 4. The Figure 3.1b showes the resulting code.

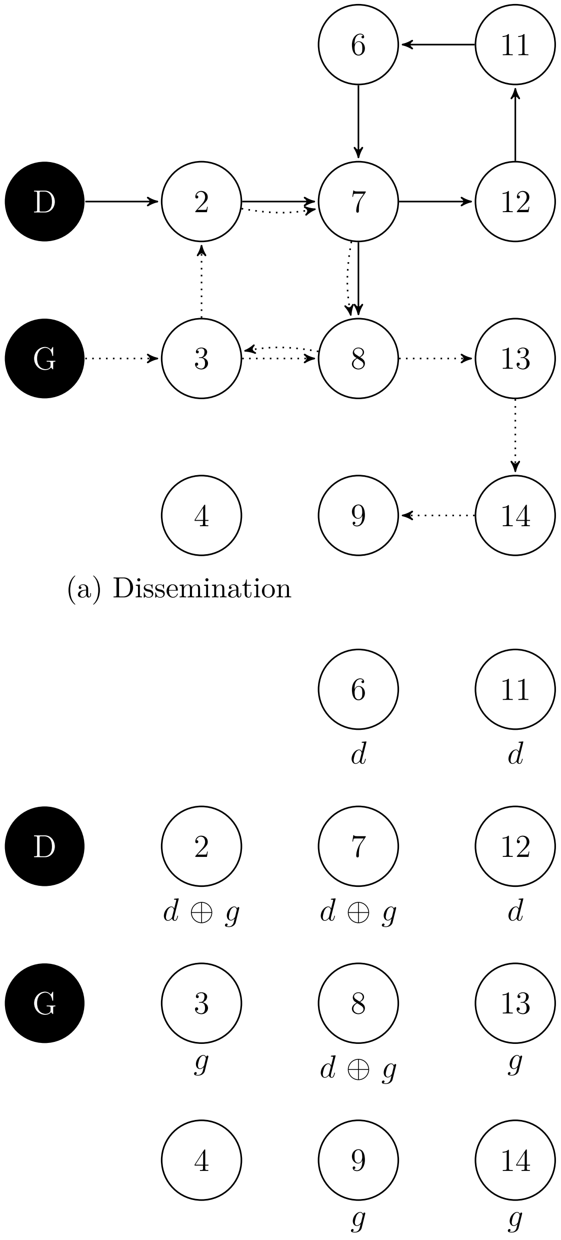


(a) Dissemination

(b) Encoding

Figure 3.1: An Example of DSwEE: m=6

In the same way, we show the operation of DEC-EaD in Figure 3.2. Source node D starts random walk for m=7 as following {2,7,12,11,6,7,8} as shown in Figure 3.2a. The relay node at each hop selects only one of its neighbours in a uniformly random fashion. When the relay node is revisited by the same data packet, the packet will be forwarded to a neighbour which is different from the last one such as node 7 in this random walk. When the random walk reaches node 7 for the first time, the packet is forwarded to node 12 and then to node 8 for the second time. The value of m in the packet is decremented only by all newly visited node. When node 7 is revisited, the value of m is not changed. Finally, this random walk is terminated when a value of m reaches zero at node 8. Node G applies the same logic to execute its random walk as {3,8,3,2,7,8,13,14,9}. The figure 3.2b depicts the resulting code.



(a) Dissemination

(b) Encoding

Figure 3.2: An Example of DEC-EaD: m=7

CHAPTER 4

PERFORMANCE EVALUTION OF DEC-DS,DEC-EAD AND DSWEE

This chapter presents a performance evaluation of DEC-DS, DEC-EaD and DSwEE under WSNs MAC layer 802.15.4 using the OMNeT++ simulation tool with the INET 3.6 framework. In this chapter, we show the various simulation parameters and describe the models used to test the schemes in terms of two performance metrics: code survivability and energy consumption. Finally, results are analysed and discussed.

4.1 Simulation Configuration

We conduct simulations using the OMNeT++ simulation tool and the INET 3.6 framework. We assume a grid topology where each node has four direct neighbours. The transmission range of every node is set to 100m and data forwarding works in a multi-hop fashion. In the Physical Layer, the 2.4 GHz band and a bandwidth of 250 kbps are used in the simulations. We use the 802.15.4 MAC based on collision avoidance via CSMA/CA for MAC layer that included in the INET framework. DEC-DS and DSwEE schemes use a link state routing protocol, where a shortest path routing table for each destination is automatically generated beforehand at the routing layer. For the DEC-EaD scheme, a neighbor discovery mechanism is supposed to apply in which every node maintains all its neighbors. However, in our simulation, every node keeps an index for all its single-hop neighbors. For traffic generator, User Datagram Protocol (UDP) application is used. In addition, some important fixed parameters and default values of variable parameters are listed in Table 4.1.

Parameter	**Value**
Path Loss Type	Free Space Path Loss
Propagation Type	Constant Speed Propagation
Background Noise Type	Isotropic Scalar Background Noise
Carrier Frequency	2.4 GHz
Transmission range	100m
Transmitter power	0.33 mW
Receiver sensitivity	-85dBm
Wireless Protocol	IEEE802.15(ZigBee) based on CSMA/CA
Mac Bitrate	250Kbps
Traffic Generator	UDP

Table 4.1: Simulation parameters

We assume no overlap between the sets of source and storage nodes in order to ease our simulation and analysis. However, in a real implementation all nodes serve as storage nodes. In our experiments, we use $\mathbb{F}_2$ which is appropriate with WSNs applications due to its simplicity. We create a network of k sensor nodes and n storage nodes according to Equation 2.1. The value of data survivability is required to calculate k and n, in our simulation, we use s=3. We calculate the Redundancy Factor, m using Equation 2.2.

4.1.1 Testing Code Survivability

The original DEC-scheme in[4] assumed there would be packet acknowledgments to avoid packets loss, and that assumption is supported due to the entirely randomized nature of the solution. To test the code survivability for DEC-schemes, it is necessary to disseminate the data and build a code according to the value of a redundancy factor m. So, errors caused by interference are not considered in [24], [23]. However, the erasure/failure model is used as explained in Chapter 2. Therefore, in our simulation, the 802.15.4 MAC which is based on collision avoidance via CSMA/CA for the MAC layer is used. Consequently, avoiding errors due to interference is considered. However, we use the erasure/failure model that is used in [24], [23] to test the performance of the code survivability and to compare between all schemes. To test the code survivability, we set f, the erasure rate, to values between 0 and 1. For each value of f, $f \times n$ storage nodes are selected to fail in a uniformly random fashion. Then, data gathering is carried out. Next, decodability is tested by selecting storage nodes randomly to build $\overline{\mathrm{G}}$ to check if the k native data packets can still be recovered. Decoding is successful when the rank of ($\overline{\mathrm{G}}$)=k. Otherwise, decoding fails. We are interested to determine a maximum possible failure rate of f between 0 and 1, which is denoted by f_m. When f > f_m, the available encoded packets are not enough to recover all native data packets. f_m can be defined as in Equation 4.1, where n' is the number of erased storage nodes and n is the total number storage nodes.

$$f_m = \frac{n'}{n} \tag{4.1}$$

f_m is expected to equal $\dfrac{s}{s+1}$ to achieve the best performance of the code survivability and that means the maximum possible rate of sensor nodes that fail with the retrivability of all native data packets. The probability of successful decoding (P_s) is calculated by generating and testing a large number of different test cases. The general steps of the simulation are illustrated in Algorithm 3. The simulation was carried out for k=20,30,40 and 50 and s=3. The results are generated using 100 runs where source nodes in each run execute initial random choices of storage nodes. P_s is defined as:

$$P_s = \frac{\text{Number of successful decodings}}{\text{Total number of trails}} \tag{4.2}$$

Algorithm 3 General Steps of the simulation

```
Require: $k$, $s$, $\mathbb{F}_2$
$n$= $k(s+1)$;
$m$= $(1+s)(\log(k)+7)+8$;
for $i$=1 to 100  do
   Disseminate($n$,$m$)
   for failure level ($f$) from 0 to 1 do
      Collect packets ($f$);
      Determine $f_m$;
   end for
end for
```

Finally, we get, on average, the critical ratio of the maximum possible failure f_m and the remaining nodes as shown in Table 4.2.

It shows that the critical rate of the maximum possible failure is 0.75 for different

k	n	DEC-DS		DEC-EaD		DSwEE	
		Remaining Nodes	f_m	Remaining Nodes	f_m	Remaining Nodes	f_m
20	80	20	0.75	21	0.74	21	0.74
30	120	30	0.75	31	0.74	31	0.74
40	160	40	0.75	42	0.74	44	0.73
50	200	50	0.75	56	0.72	58	0.71

Table 4.2: Probability of successful decoding (P_s) for k=20-50, for DEC-DS,DEC-EaD and DSwEE

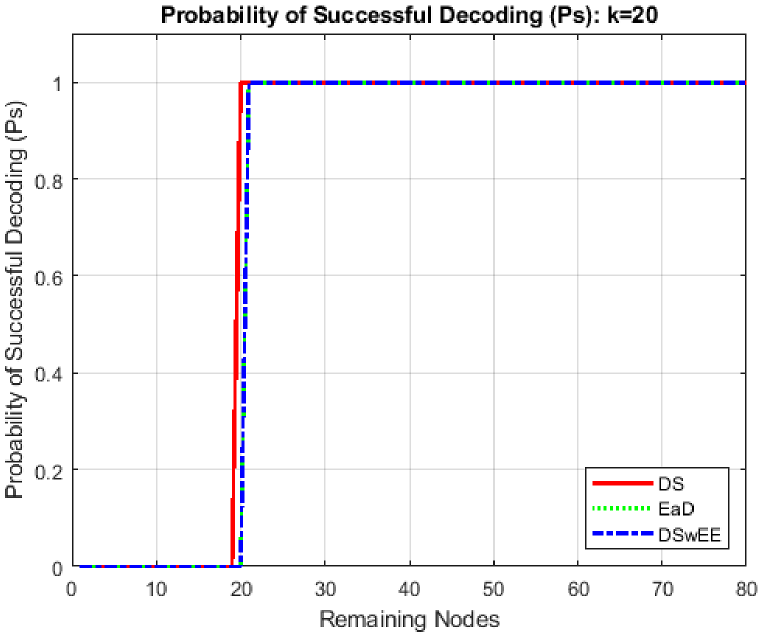


Figure 4.1: Probability of successful decoding (P_s) for k=20, for DEC-DS,DEC-EaD and DSwEE

values of k, for DEC-DS and that means the number of remaining nodes is k. When $f \leq f_m$=0.75, the data collector can gather k encoded packets from any k storage nodes and builds a full rank matrix $\overline{\mathrm{G}}$ with higher probability than DEC-EaD and DSwEE. The reason behind this is that the dissemination phase of constructing DEC-DS is improved in DEC-EaD and DSwEE in terms of energy consumption, but it lowers the performance of the code survivability. When relay nodes participate in the coding process as in DEC-EaD and DSwEE, it results in a higher chance multiple storage nodes creating the same encoded packets (same linear equations). However, DEC-EaD and DSwEE present a reasonable performance of the code survivability especially when we take achieving energy efficiency into account.

The performance comparison of the resulting code in terms of P_s for DEC-DS, DEC-EaD and DSwEE is showed in Figure 4.1, Figure 4.2 , Figure 4.3 and Figure 4.4. Figure 4.1 shows the value of P_s for k=20 and s=3. For DEC-DS, the probability as it is expected, where P_s=1 $\forall$ $f \leq \frac{s}{s+1}$= 0.75, but when $f >$ 0.75, the encoded packets are not enough to retrieve all native data packets. The graph also shows P_s=1 $\forall$ f $\leq$ 0.74 for DEC-EaD and DSwEE. However, we observe there is no a considerable difference in f_m between DEC-DS, DEC-EaD and DSwEE. In addition, we apply the same experiments after changing the survivability value to s=2, and we get the same observations.

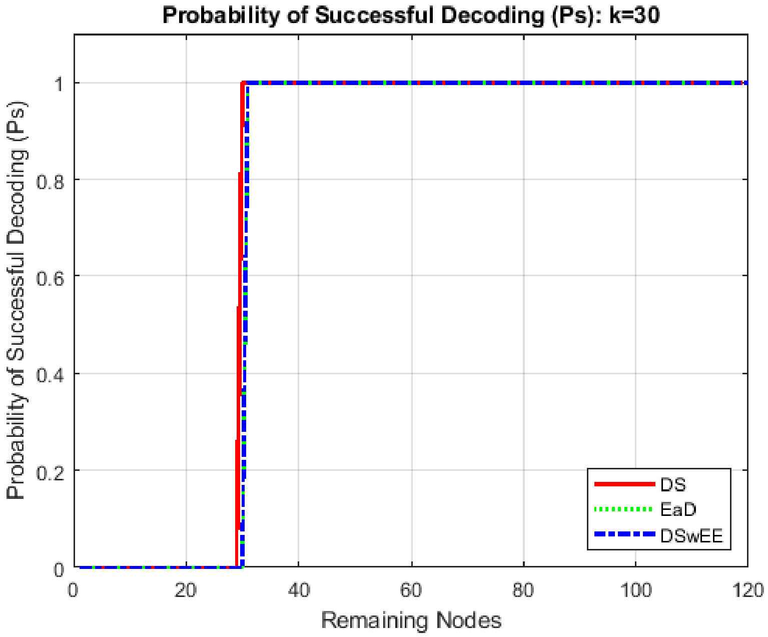


Figure 4.2: Probability of successful decoding (P_s) for k=30, for DEC-DS,DEC-EaD and DSwEE

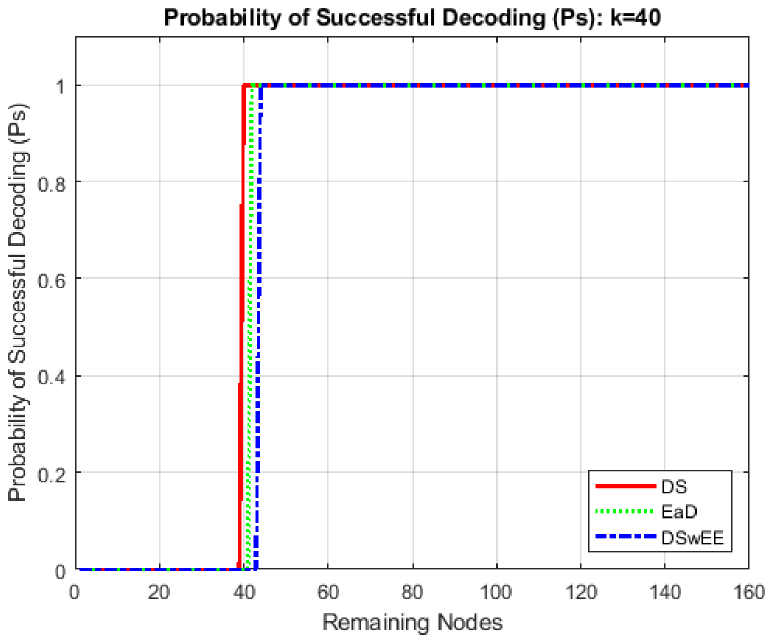


Figure 4.3: Probability of successful decoding (P_s) for k=40, for DEC-DS,DEC-EaD and DSwEE

We can also notice the same observation for k=30 in Figure 4.2, for k=40 in Figure 4.3 and for k=50 in Figure 4.4.

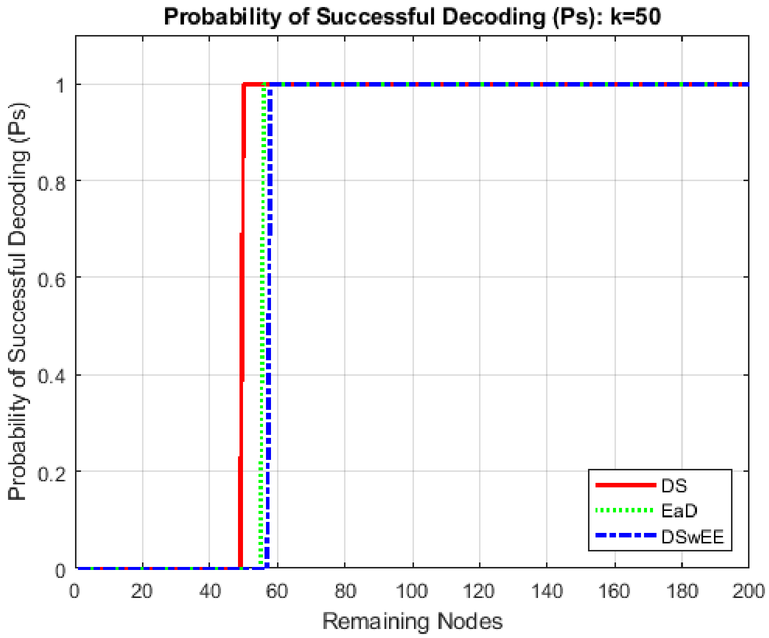


Figure 4.4: Probability of successful decoding (P_s) for k=50, for DEC-DS,DEC-EaD and DSwEE

4.1.2 Energy Efficiency

In the simulation, to record the energy consumption of the radios, every node has an energy consumer module in its radio by default. A constant power consumption value to each radio mode and transmitter/receiver state are assigned in this module for wireless sensor nodes. For example, in receive mode, when radios are idle, a small amount of power is consumed such as during listening for transmissions. The more energy is consumed by radios during the states of receiving a transmission, or transmitting.

Moreover, every node has an energy storage module by default in order to keep track of the energy consumption of the nodes. An infinite amount of energy is stored in this module, but getting fully charged or depleted is not the case, it is used only to measure power consumption.

The difference between radio modes and states is in which radios operate in various modes, such as off, sleep, receiver, transmitter. Setting the mode by the sensor node radio model is done without depending on external effects. On the other hand, states of the radios are relied on what in the given mode, they are performing such as receiving a transmission, transmitting or listening. That is depended on external factors for example transmissions happening the medium. The consumed power by the radio mode, the transmitter, and the receiver states, and the values are shown in Table 4.3, roughly relied on the data sheet for the CC2500 RF transceiver as built in the simulation. The consumed energy required for building code by all schemes is recorded. However, the energy consumed by all nodes to discover their neighbors in DEC-EaD or to build a shortest path routing table for each node in DSwEE is neglected. On the other hand, during the data collecting stage, the failure of nodes may happen to cause the fragmentation for the network. So, by focusing on the data survivability rather than the network survivability, it was assumed that the data collector can reach all nodes in the network.

Parameter	**Value (mW)**
off Power Consumption	0
sleep Power Consumption	0.001
switching Power Consumption	25
receiver Idle Power Consumption	0.005
receiver Busy Power Consumption	0.1
receiver Receiving Power Consumption	50
transmitter Idle Power Consumption	5
transmitter Transmitting Power Consumption	75

Table 4.3: Power consumption of the CC2500 RF transceiver

4.2 Result and Analysis

Figure 4.5 depicts the total energy consumed by nodes in the network for each scheme for different values of *k* to construct the code (in joules). It was shown that DswEE and DEC-EaD can construct the required code with very less energy compared to DEC-DS. DEC-EaD was shown to be better than DEC-EaF in terms of energy consumption in [23], [24]. Unfortunately, there is a simulation limitation to implement DEC-EaF in our study, where a multipath routing table for all destinations in the network is not supported. DSwEE acheives better performance than DEC-EaD in terms of energy consumption as shown in Figure 4.5. The difference between DEC-EaD and DSwEE is that in DEC-EaD, when a packet is received by a relay node, it is forwarded based

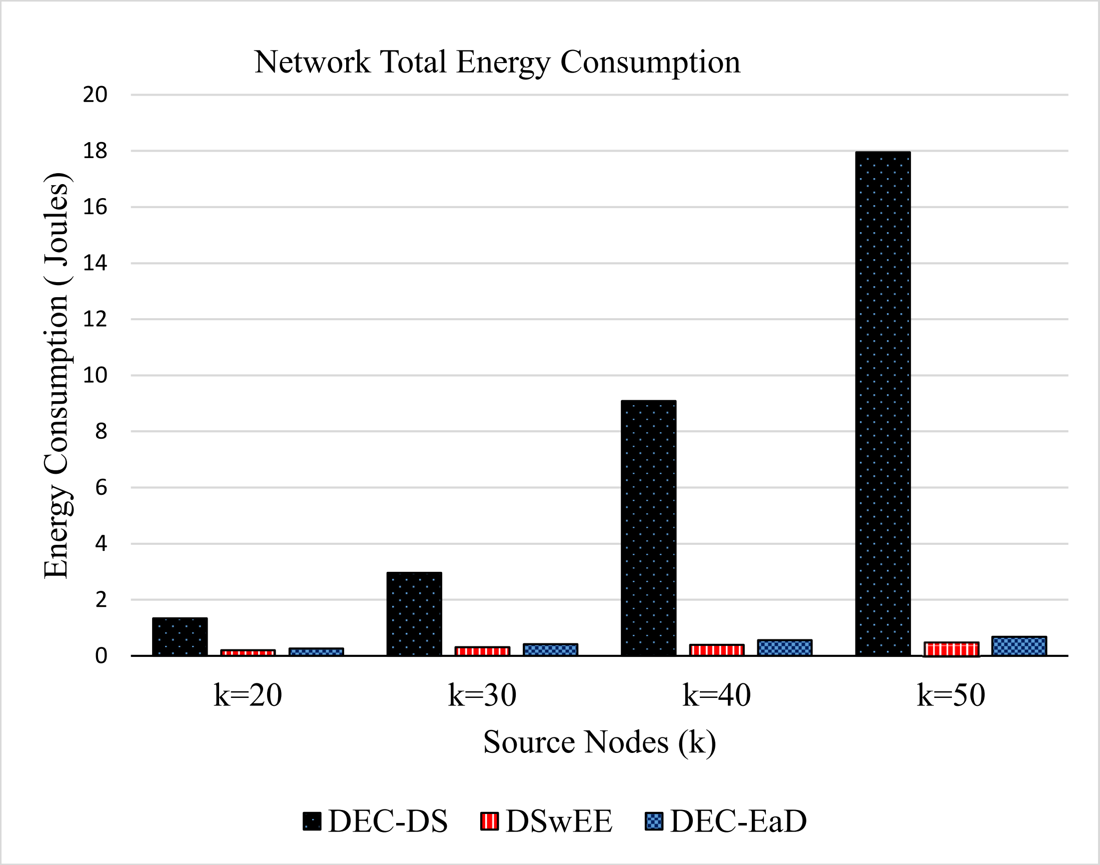


Figure 4.5: Enegry consumption for DEC-DS, DEC-EaD and DSwEE, for s=3.

on the rotor-router model. Every node in the network maintains an index for each received packet. When the received packet is forwarded the first time, the index records the neighbor to whom the packet is forwarded . Later, when the same packet is received by this relay node (called revisited node), it is sent to the next neighbor that is different from the last one and so on. This nature of rotor-router random walk in forwarding a packet only to one of the neighbours of the relay node in localized fashion leads to a higher probability of a node being revisited by the same packet than that of DSwEE. As a result, this increases the energy consumption of the nodes in DEC-EaD since the packet has already been used for encoding. It can be easily shown that the probability of a node being revisited by the same packet is close to 1

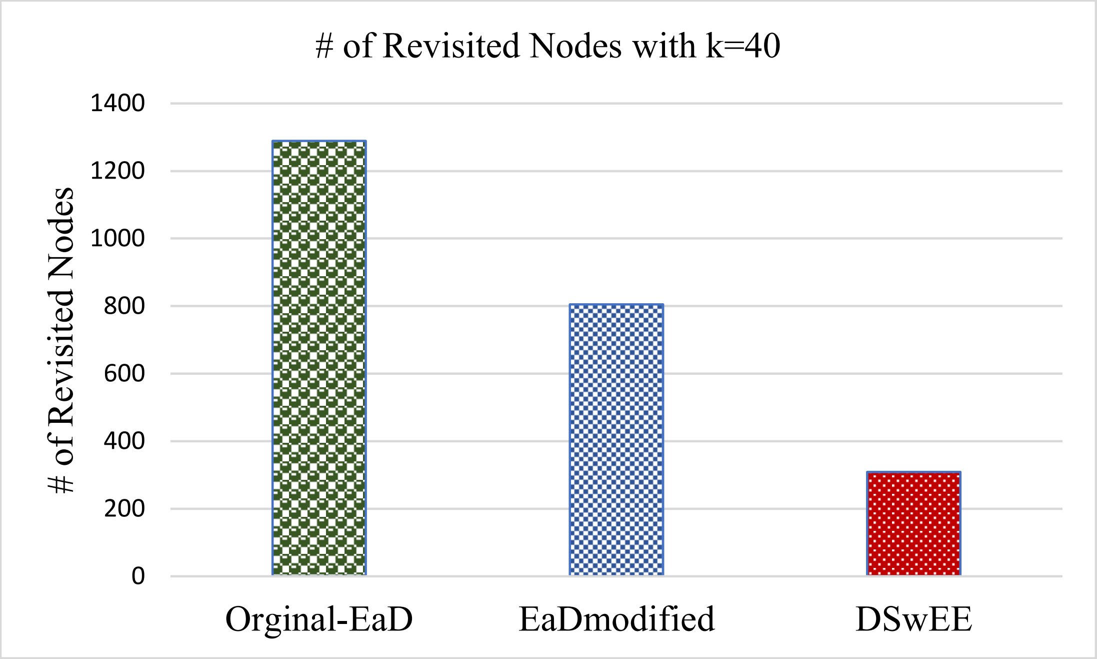


Figure 4.6: The total number of revisited nodes with k=40 for DEC-EaD and DSwEE

after all neighbours have been visited. On the other hand, in DSwEE, the selection of distinct destinations of storage nodes is carried out uniformly at random from a large pool of candidate nodes which could be any node in the network. This leads to a more uniform dissemination of packets over the network. As shown in Figure 4.6, for k=40, we measure the number of nodes that are revisited by the same packet (for all native packets that are generated by source nodes). To enhance DEC-EaD in terms of energy consumption, we tested a modified version of DEC-EaD where after a node receives a packet, its index points to the next neighbor excluding the one from which the packet was received. Figure 4.6 shows that EaD-modified has a lower number of revisited nodes.

We finally, present the Relative Efficiency Factor (REF) as an indicator to be a comparative performance metric that combines two metrics: code survivability and energy consumption in order to compare all schemes at once. To define RFE, we need to normalize the value of energy consumption relatively (Ξ) as follows:

$$\Xi = 1 - \frac{\text{Energy consumption for each scheme}}{\text{Sum of energy consumption for all schemes}} \tag{4.3}$$

So, REF is defined as

$$REF = \Xi \times f_m \tag{4.4}$$

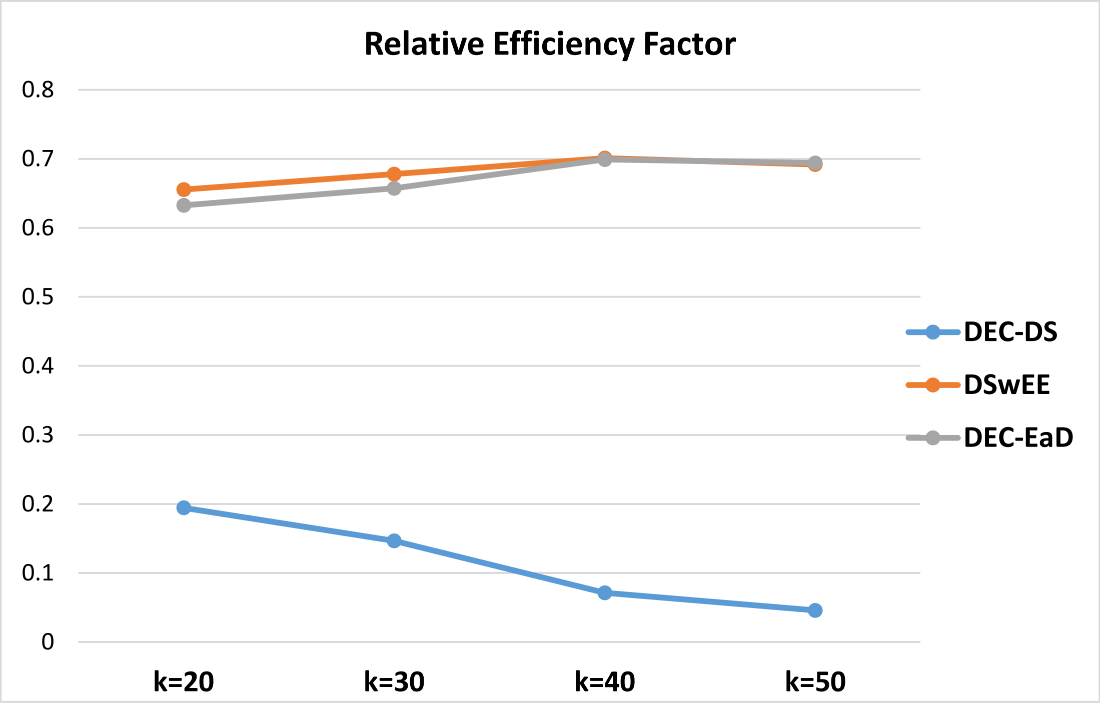


Figure 4.7: REF for DEC-DS, DEC-EaD and DSwEE.

Figure 4.7 shows the REF of all schemes, when the curve goes to 0, it is worst. As it is clear from the figure, DEC-DS achieves the best performance of code survivability for all different values of k, but with more energy consumption. When the network size is increased, the energy consumption increases. DEC-EaD and DSwEE achieves good performance of code survivability for k=20 and k=30 and with less energy consumption, especially DSwEE. However, for k=40 and k=50, DEC-EaD achieves better performance of code survivability than DSwEE and that makes the efficiency of DEC-EaD and DSwEE seems the same. Finally, DEC-DS and DSwEE have one overhead compared against DEC-EaD that nodes need more storage because they have to save the routing tables.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, we reviewed different schemes in DEC-based Distributed Data Storage Systems for data survivability in WSNs. We analyzed their original ideas, models and related literature. A survey of a myriad of studies on developing several variants of the original DEC system is conducted. It was showed that there is a need for an evaluation on a standard WSNs MAC protocol. The original DEC-DS was proposed to achieve the data survivability for WSNs. Although two schemes DEC-EaD and DEC-EaF were suggested to improve DEC-DS efficiency in terms of energy consumption, it still needs enhancements to be used widely. The technique used to disseminate the data according to the redundancy factor m while taking advantage of RLNC could help in improving the DEC-DS in terms of energy consumption. In DEC-EaD, the rotor-router model random walk benefited from RLNC that was used to disseminate a data packet with less energy consumption than DEC-EaF. A complicated technique used to disseminate a data packet was suggested in DEC-EaF by using a multipath routing table and then choosing the shortest path according to the condition of large unvisited hops nodes, and also RLNC was used with a routing setup. We have introduced DSwEE, which is simpler than DEC-EaF in terms routing and RLNC setup, and it shows better performance in energy efficiency than DEC-EaD. In DSwEE, the data is disseminated in a decentralized fashion where each sensor node generates the data packet and forwards it to a randomly and uniformly selected storage node through the shortest route. Each hop nodes in that route participate in establishing the target code. The distribution of the coding process is tracked by the redundancy factor m, which is embedded in the packet and is decreased by one in each newly visited node. If m is non-zero, the selected destination updates m and disseminates a data packet with the same logic is done at first dissemination until m becomes zero, so the forwarding process is finished. Experiments were conducted to evaluate all schemes DEC-DS, DEC-EaD and DSwEE in terms of two metrics, the code survivability and the energy consumption required to build code. There is a limitation to implement DEC-EaF in our simulation where a multipath routing table for all destinations in the network is not supported. All these experiments were carried out where WSN communication protocols were used such as IEEE 802.15.4 MAC layer. Experiments were carried out with various network sizes, k=20,30,40 and 50. DEC-DS achieved a better performance in terms of the code survivability than the others, but with high energy consumption. Relay nodes participate in the code process as in DEC-EaD and DSwEE to enhance the energy consumption. However, it does not seem to be a considerable differences in the performance of the code survivabiity between DEC-

DS, DEC-EaD and DSwEE especially when we take the energy efficiency achieved by DEC-EaD and DSwEE into account. The performance of DSwEE is superior to other DEC-schemes in terms of saving energy during implementing the required code. We recommend using DEC-EaD for WSNs applications where is no need for routing. In addition, we need to investigate how to extend the framework for some situations, for example, the various classes of data need specific requirements of the data survivability. The distributed storage relied on the Quality of Services (QoS) can be implemented to store data packets with various levels of priority. Moreover, in DEC-EaD, the selection of a destination is restricted to only to one of the neighbours of the relay node in rotor-router model while in DSwEE, it is carried out uniformly and randomly from a large pool of candidate nodes which could be any node in the network. As a result, DSwEE distributes the data over the network uniformly while in DEC-EaD the data is disseminated in more localized fashion. Hence, in DEC-EaD, some data may not be recovered when a high rate of failure occurs in a certain region of the network. As a future work, we plan to investigate this kind of failure.

REFERENCES

[1] H.-Y. Lin and W.-G. Tzeng, “A secure decentralized erasure code for distributed networked storage,” *IEEE transactions on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1586–1594, 2010.

[2] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, Apr. 2007.

[3] H. Weatherspoon and J. D. Kubiatowicz, “Erasure coding vs. replication: A quantitative comparison,” in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 328–337.

[4] A. Dimakis, K. Ramchandran, and V. Prabhakaran, “Decentralized erasure codes for distributed networked storage,” cs/0606049, Tech. Rep., 2006.

[5] G. Kaur and R. M. Garg, “Energy efficient topologies for wireless sensor networks,” *International Journal of Distributed and Parallel Systems*, vol. 3, no. 5, p. 179, 2012. [Online]. Available: http://airccse.org/journal/ijdps/papers/3512ijdps16.pdf

[6] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, “Ieee 802.15.4: a developing standard for low-power low-cost wireless personal area networks,” *IEEE network*, vol. 15, no. 5, pp. 12–19, 2001.

[7] A. Giridhar and P. Kumar, “Toward a theory of in-network computation in wireless sensor networks,” *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, Apr. 2006.

[8] S. Kar and J. Moura, “Sensor Networks With Random Links: Topology Design for Distributed Consensus,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.

[9] O. Chughtai, N. Badruddin, and A. Awang, “Distributed on-demand multi-optional routing protocol in multi-hop wireless networks,” in *TENCON 2014 - 2014 IEEE Region 10 Conference*, Oct. 2014, pp. 1–6.

[10] *Protocols and Architectures for Wireless Sensor Networks*, Dec. 2015. [Online]. Available: https://books.google.com.sa/books/about/Protocols_and_Architectures_for_Wireless.html?hl=ar&id=170R-1aZsQYC

[11] A. Abuarqoub, F. Alfayez, M. Hammoudeh, T. Alsboui, and A. Nisbet, “Simulation issues in wireless sensor networks: A survey,” in *SENSORCOMM’12: Proceedings of the 2012 Sixth International Conference on Sensor Technologies and Applications, Rome, Italy. BOOK CHAPTERS*, 2012.

[12] Q. I. Ali, A. Abdulmaowjod, and H. M. Mohammed, “Simulation & performance study of wireless sensor network (wsn) using matlab,” in *2010 1st International Conference on Energy, Power and Control (EPC-IQ).* IEEE, 2010, pp. 307–314.

[13] Y. Tselishchev, A. Boulis, and L. Libman, “Experiences and lessons from implementing a wireless sensor network mac protocol in the castalia simulator,” in *2010 IEEE Wireless Communication and Networking Conference.* IEEE, 2010, pp. 1–6.

[14] A. Sobeih, J. C. Hou, L.-C. Kung, N. Li, H. Zhang, W.-P. Chen, H.-Y. Tyan, and H. Lim, “J-sim: a simulation and emulation environment for wireless sensor networks,” *IEEE Wireless Communications*, vol. 13, no. 4, pp. 104–119, 2006.

[15] A. Varga, “Omnet++,” in *Modeling and tools for network simulation.* Springer, 2010, pp. 35–59.

[16] A. Basit and S. A. Khan, “Design and simulation of a mobile ad-hoc network in hla environment,” in *Proc., 9th WSEAS Int. Conf. on Automatic Control, Modeling & Simulation, WSEAS*, 2007, pp. 151–156.

[17] A. Varga, “Discrete event simulation system,” in *Proc. of the European Simulation Multiconference (ESM’2001)*, 2001.

[18] T. Steinbach, H. D. Kenfack, F. Korf, and T. C. Schmidt, “An extension of the omnet++ inet framework for simulating real-time ethernet with high accuracy,” in *Proceedings of the 4th International ICST Conference on Simulation Tools and*

Techniques. ICST (Institute for Computer Sciences, Social-Informatics and ..., 2011, pp. 375–382.

[19] D. Klein and M. Jarschel, "An openflow extension for the omnet++ inet framework," in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques.* ICST (Institute for Computer Sciences, Social-Informatics and ..., 2013, pp. 322–329.

[20] A. Virdis, G. Stea, and G. Nardini, "Simulte-a modular system-level simulator for lte/lte-a networks based on omnet++," in *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH).* IEEE, 2014, pp. 59–70.

[21] OMNeT++ developers and INET Framework contributors , "Inet framework," Jan 29, 2019, [Online; accessed 26-February-2019]. [Online]. Available: https://inet.omnetpp.org/

[22] G. R. Goodson, J. J. Wylie, G. R. Ganger, and M. K. Reiter, "Efficient byzantine-tolerant erasure-coded storage," in *Dependable Systems and Networks, 2004 International Conference on.* IEEE, 2004, pp. 135–144.

[23] L. Al-Awami and H. Hassanein, "Data survivability for wsns via decentralized erasure codes," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International.* IEEE, 2012, pp. 94–99.

[24] L. Al-Awami and H. S. Hassanein, "Distributed data storage systems for data survivability in wireless sensor networks using decentralized erasure codes," *Computer Networks*, vol. 97, pp. 113–127, 2016.

[25] M. Ashraf, S. Idrus, F. Iqbal, R. Butt, and M. Faheem, "Disaster-resilient optical network survivability: a comprehensive survey," in *Photonics*, vol. 5, no. 4. Multidisciplinary Digital Publishing Institute, 2018, p. 35.

[26] C. P. Kumar and R. Selvakumar, "Efficient data reconstruction in sensor networks using optimal locally recoverable codes," in *Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on.* IEEE, 2017, pp. 217–221.

[27] S. Kim, R. Fonseca, and D. Culler, "Reliable transfer on wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on.* IEEE, 2004, pp. 449–459.

[28] S. Ali, A. Fakoorian, and H. Taheri, "Optimum reed-solomon erasure coding in fault tolerant sensor networks," in *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on.* IEEE, 2007, pp. 6–10.

[29] B. Marchi, A. Grilo, and M. Nunes, "Dtsn: Distributed transport for sensor networks," in *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on.* IEEE, 2007, pp. 165–172.

[30] M. S. Srouji, Z. Wang, and J. Henkel, "Rdts: A reliable erasure-coding based data transfer scheme for wireless sensor networks," in *Parallel and Distributed*

Systems (ICPADS), 2011 IEEE 17th International Conference on. IEEE, 2011, pp. 481–488.

[31] R. Kumar, A. Paul, U. Ramachandran, and D. Kotz, "On improving wireless broadcast reliability of sensor networks using erasure codes," in *International conference on mobile ad-hoc and sensor networks.* Springer, 2006, pp. 155–170.

VITAE

- Name: Mohammed Abdullah Asiri
- Nationality: Saudi Arabian
- Date of Birth: 07 Dec 1982
- Email: *maalasiri1@gmail.com*
- Permenant Address: Najran-Saudi Arabia, Najran University