Development of an energy and geographic aware opportunistic network coding scheme

Dissertation submitted in fulfilment of the requirements for the degree *Master of Engineering* in Computer Engineering at the Potchefstroom campus of the North-West University

M.J. Engelbrecht

20704313

Supervisor: M.J. Grobler

November 2012

Declaration

I, Mario Engelbrecht hereby declare that the dissertation entitled "Development of an energy and geographic aware opportunistic network coding scheme" is my own original work and has not already been submitted to any other university or institution for examination.

M.J. Engelbrecht

Student number: 20704313

Signed on the 20th day of November 2012 at Potchefstroom.

Acknowledgements

I dedicate this thesis to all those who made it possible:

Firstly, I would like to thank my Saviour, Jesus Christ, for the opportunity, motivation, and determination required to have conducted this research.

I would like to thank my supervisor, Leenta, for her guidance and assistance in the process of conducting the research.

I thank my family and friends for their never-ending love and support.

Thank you very much Telkom Centre of Excellence for the financial support.

Abstract

The evolution of communication networks has led us to an era where you cannot only perform surgery halfway across the world, but do so while being in the comfort of your own home. By eliminating the need for wires, wireless networks revolutionised communication networks by enabling nodes to communicate while being in a mobile state. The concept opened many doors to new applications and possibilities.

Network Coding is a technique that optimises the throughput of a network by coding packets. Geo-Routing is a routing method that operates by using the geographical distances between nodes as the routing metric. Opportunistic Routing is a routing method that exploits the broadcast characteristics of wireless networks.

In this thesis, we developed a routing scheme that incorporates Network Coding, Geo-Routing and energy aware conditions. It accomplishes this task by using one of the key phases constituting Opportunistic Routing.

The developed routing scheme was implemented in OMNeT++. Various simulation experiments were conducted in OMNeT++ pertaining to the implemented scheme. The results indicate significant increase in performance metrics such as throughput and survivability.

Keywords: *Energy Awareness, Geo-Routing, Mobile Ad Hoc Network, Network Coding, Opportunistic Routing.*

Contents

List of Figures **x**

List of Tables **xiv**

List of Acronyms **xv**

1 Introduction **1**

1.1 Introduction 2

1.2 Background 2

1.3 Motivation 3

1.4 Proposed research 3

1.5 Hypothesis 4

1.6 Justification for study 4

1.7 Research goals 4

1.8 Issues to be addressed 5

1.9 Research methodology 6

1.9.1 Validation and verification 7

1.10 Dissertation overview 11

1.11 Conclusion 11

2 Literature Study **12**

2.1 Introduction . . . 13

2.2 Wireless Networks . . . 13

2.2.1 Wireless Infra-structured Networks . . . 13

2.2.2 Wireless Ad Hoc Networks . . . 14

2.2.3 Network characteristics . . . 16

2.2.4 Network performance . . . 18

2.3 Network Models . . . 19

2.3.1 OSI Model . . . 19

2.3.2 TCP/IP protocol suite . . . 25

2.4 IEEE 802.11 standard . . . 25

2.4.1 Data link layer . . . 27

2.4.2 Physical layer . . . 33

2.5 IEEE 802.15.4 standard . . . 33

2.6 Geo-Routing . . . 34

2.7 Energy aware routing . . . 35

2.8 Network Coding . . . 36

2.8.1 Benefits of Network Coding (NC) . . . 38

2.8.2 XOR operation . . . 41

2.8.3 Reverse carpooling . . . 42

2.8.4 Deterministic Network Coding . . . 43

2.8.5 Opportunistic Network Coding . . . 44

2.8.6 COPE . . . 45

2.9 Opportunistic Routing . . . 50

2.9.1 ETX-Based Protocols . . . 52

2.9.2 Opportunistic Any-Path Forwarding 53
2.9.3 MAC-Independent Opportunistic Routing & Encoding Protocol . 54
2.9.4 Hop-Count-Based Protocols 55
2.9.5 Geo-Distance-Based Protocols 56
2.10 CORE 56
2.11 Conclusion 57

3 Routing Scheme Design **58**
3.1 Introduction 59
3.2 Basic functioning of scheme 59
3.2.1 Forwarder set selection 59
3.2.2 Priority based forwarding 62
3.3 Conclusion 64

4 Network Model Implementation **65**
4.1 Introduction 66
4.2 Network model 66
4.2.1 Physical layer 66
4.2.2 MAC Layer 66
4.2.3 Network layer 67
4.2.4 Upper layers 70
4.3 OMNet++ implementation 70
4.3.1 Implementation modules 70
4.3.2 Implementation layers 72
4.3.3 Model assumptions 79
4.4 Conclusion 79

5 Validation and Verification **80**

5.1 Introduction . . . 81

5.2 Methodology . . . 81

5.2.1 Verification . . . 81

5.2.2 Validation . . . 82

5.3 Simulated scenarios . . . 83

5.3.1 Hidden node scenario . . . 83

5.3.2 Coding factor . . . 84

5.3.3 Geo-distance factor . . . 91

5.3.4 Energy factor . . . 93

5.4 Model comparison . . . 99

5.5 Conclusion . . . 99

6 Simulation and Analysis **100**

6.1 Introduction . . . 101

6.2 Visual Aspect of the Simulation . . . 101

6.3 Simulation Setting . . . 101

6.4 Simulation 1 - Coding Factor . . . 102

6.5 Simulation 2 - Distance Factor Incorporated . . . 105

6.6 Simulation 3 - Energy Factor Incorporated . . . 107

6.7 Conclusion . . . 111

7 Conclusion **112**

7.1 Introduction . . . 113

7.2 Research hypothesis . . . 113

7.3 Literature study . . . 113

7.4 Development and implementation of routing scheme 114
7.5 Validation and verification 114
7.6 Simulation results 114
7.7 Correctness of hypothesis 115
7.8 Future Work 115

Bibliography **116**

Appendices

A Conference Contributions from this Thesis **121**

A.1 An Improved Method for the Identification of Opportunities for Deterministic Network Coding 121
A.2 Unselfish Network Coding Games 128
A.3 Development of an Energy and Geographic Aware Opportunistic Network Coding Scheme 131

List of Figures

1.1 Deductive research logic 6

1.2 Inductive research logic 6

1.3 Research methodology 7

1.4 Simplified validation and verification model [1] 8

1.5 The research methodology model for this research 10

2.1 Node isolation 17

2.2 Wireless probability 18

2.3 Seven layers of OSI model 20

2.4 OSI model inter- and intra-communication 21

2.5 Physical layer 21

2.6 Data link layer frame 22

2.7 Delivery terms 23

2.8 Segmentation and reassembly 23

2.9 Synchronisation 24

2.10 Presentation layer 25

2.11 The OSI and TCP/IP models 26

2.12 IEEE 802.11 layers 26

2.13 Timing strategies in CSMA/CA 28

2.14 Flow diagram for the CSMA/CA 29
2.15 Hidden node scenario 31
2.16 Exposed node scenario 31
2.17 CSMA/CA with handshaking 32
2.18 Handshaking 33
2.19 Geo-routing 35
2.20 Two-hop traditional routing 36
2.21 Two-hop NC routing 37
2.22 Min-cut max-flow theorem 38
2.23 Network with Network Coding 39
2.24 Packet combining 39
2.25 Minimising delay 40
2.26 Network Coding scenario 41
2.27 XOR coding operation 41
2.28 XOR decoding operation 42
2.29 Reverse carpooling 42
2.30 Network coding topologies a) Bow-tie b) Butterfly 44
2.31 COPE coding 46
2.32 COPE decoding 47
2.33 Virtual queues 48
2.34 Coding and MAC gain 49
2.35 Forwarder set 52
2.36 Forwarder set packet drop 52
2.37 Random Linear Network Coding 55
2.38 Opportunistic Routing in Dynamic Ad-Hoc Networks 55

2.39 Resilient Opportunistic Mesh Routing for Wireless Mesh Networks . . . 56

3.1 Forwarder set problem scenario . . . 61
3.2 Node waves . . . 61
3.3 Forwarder set . . . 62
3.4 Priority timer . . . 63

4.1 Forwarder set neighbour . . . 69
4.2 NewNetwork . . . 71
4.3 Node module . . . 72
4.4 Wireless NIC . . . 72
4.5 From MAC flow diagram . . . 74
4.6 Basic conditions . . . 75
4.7 Calculate forwarder set . . . 76
4.8 Prioritisation calculation . . . 77

5.1 Model confidence . . . 82
5.2 Hidden node topology . . . 83
5.3 Hidden node simulation . . . 84
5.4 Two hop coding topology . . . 85
5.5 Two hop coding simulation . . . 86
5.6 Decoding . . . 86
5.7 Forwarder set coding prioritisation topology . . . 87
5.8 Traffic routes . . . 88
5.9 Forwarder sets . . . 89
5.10 Forwarder set coding prioritisation simulation . . . 90
5.11 Geo-distance factor topology . . . 91

5.12 Geo-distance forwarder set 92
5.13 Geo-distance factor simulation 93
5.14 Energy factor topology 94
5.15 Energy factor forwarder set 95
5.16 Energy factor simulation(1) 96
5.17 Energy factor simulation(2) 97
5.18 Energy factor simulation(3) 98
5.19 Throughput comparison 99

6.1 OMNeT++ network topology 101
6.2 Throughput comparison of different coding factor time periods 103
6.3 Coding comparison of different coding factor time periods 103
6.4 Throughput comparison of different distance factor time periods 106
6.5 Coding comparison of different distance factor time periods 107
6.6 Survivability comparison of different energy factor time periods 109
6.7 Throughput comparison of different energy factor time periods 111

List of Tables

2.1 Physical layer 802.11 standards 34

6.1 Throughput comparison of different coding factor time periods 104

6.2 Throughput comparison of different distance factor time periods 108

6.3 Survivability comparison of different energy factor time periods 110

List of Acronyms

WMN Wireless Mesh Network

DNC Deterministic Network Coding

MAC Media Access Control

MANET Mobile Ad Hoc Network

NC Network Coding

ONC Opportunistic Network Coding

RLNC Random Linear Network Coding

VANET Vehicular Ad Hoc Network

WSN Wireless Sensor Network

WANET Wireless Ad Hoc Network

OR Opportunistic Routing

OSI Open Systems Interconnection

PCF Point Coordination Function

DCF Distributed Coordination Function

LAN Local Area Network

WLAN Wireless LAN

LLC Logical Link Control

CSMA/CD carrier sense multiple access with collision detection

CSMA/CA carrier sense multiple access with collision avoidance

IFS Interframe Space

BEB Binary Exponential Back-off

NAV Network Allocation Vector

RTS Request to Send

CTS Clear to Send

ACK Acknowledgement

DIFS Distributed Interframe Space

SIFS Short Interframe Space

FIFO First In First Out

GR Geo-routing

ExOR Extremely Opportunistic Routing

ETX Expected Transmission Count

OAPF Opportunistic Any-path Forwarding

EAX Expected Any-path Count

OPRAH Opportunistic Routing in Dynamic Ad-Hoc Networks

ROMER Resilient Opportunistic Mesh Routing for Wireless Mesh Networks

MORE MAC-Independent Opportunistic Routing & Encoding Protocol

NIC Network Interface Card

PDA Personal Digital Assistant

Chapter 1

Introduction

The evolution of communication networks has led us to an era where you cannot only perform surgery halfway across the world, but do so while being in the comfort of your own home. By eliminating the need for wires, wireless networks revolutionised communication networks by enabling nodes to communicate while being in a mobile state. The concept opened many doors to new applications and possibilities. This chapter serves as an introduction to the proposed research.

1.1 Introduction

In this chapter, background leading to the research question is provided, followed by motivation and the proposed research. Furthermore, the research will be justified. Specific research goals will be stated as well as the issues addressed. Lastly, the methodology pertaining to the research approach will be discussed.

1.2 Background

Wireless Ad Hoc Networks (WANETs) are wireless networks constituted from wireless nodes. Nodes in these networks communicate in an ad hoc manner without the support of a centralised managing unit [2]. When routing wirelessly, certain routing techniques become viable. Such techniques include Opportunistic Routing (OR), Network Coding (NC) and Geo-routing (GR).

OR is a routing method that exploits the broadcast characteristics of wireless networks [3]. The routing method routes packets without identifying specific next hops. In other words, unlike fixed path routing, the packets travel through the network without being assigned a calculated route to the destination node.

NC is a routing technique that combines packet streams in order to distribute more information from transmissions [4]. When more information is distributed with the same number of transmissions, throughput increases. Throughput increase is a desirable outcome, therefore, making NC a desirable routing technique.

GR is a routing method that operates by using the geographical distances between nodes as the routing metric [5]. GR can only benefit networks consisting of nodes aware of their geographical coordinates.

1.3 Motivation

Different types of networks encompass different network characteristics, therefore necessitating specialised routing protocols to focus on specific network characteristics. In the case of energy constricted networks, a routing protocol is required that focusses on routing in an energy effective manner. In the case of mobile networks, a routing protocol is required to accommodate for the everchanging network topologies. This phenomenon leads to specialised routing protocols that focus on specified network characteristics; And in focussing on particular characteristics, it must do so while maintaining basic network performance with regard to basic network objectives such as throughput.

In making the nodes more **aware** by equipping them with extra information pertaining to nodes in the network, enhanced routing can be performed. If nodes are aware of the packets stored by their neighbours, they are enabled to perform localised NC. If nodes are aware of the geographic coordinates of nodes in the network, they can perform GR. Similarly, if nodes are aware of their battery statuses, they can perform energy effective routing.

When nodes are equipped with the aforementioned information, they are able to include various conditions into the routing process. With these conditions included, nodes will be able to focus on both performance metrics such as throughput as well as network characteristics such as energy limitations.

1.4 Proposed research

Different situations demand different routing methods or metrics. If we incorporate various routing factors, such as those mentioned above, into a single routing protocol, and prioritise those factors, we will have created an intelligent routing protocol that performs routing according to the metrics that a situation demands.

OR is the underlying technique that can be used to combine various routing conditions. The conditions used for prioritising the forwarder set will include NC, GR and energy awareness. These conditions are then prioritised according to the objectives to which the network has to adhere.

The developed scheme will be implemented in OMNeT++ where it will be validated and verified. Various experiments will be conducted on the implemented model to achieve results pertaining to the hypothesis.

1.5 Hypothesis

Using one of the key elements OR is comprised of, namely the *prioritising time delay calculation* phase, we can include NC, GR and energy aware routing into a single specialised routing scheme. This scheme will combine the benefits of the respective incorporated routing conditions.

1.6 Justification for study

When the aforementioned routing conditions are incorporated in a single specialised routing protocol, it will cause a performance increase in network throughput and survivability. There are a plethora of cases, especially in military applications, where high throughput along with survivability is of great concern.

1.7 Research goals

The goals of this research are:

- Develop a routing scheme that incorporates OR, NC, GR and energy aware con-

ditions.

- Determine the advantages of such a scheme.

1.8 Issues to be addressed

The following issues are to be addressed:

- A detailed literature study to show current existing schemes and techniques, how they work and to indicate the challenges and issues involved in the process of solving the problem.
- A conceptual model of the routing scheme needs to be created which incorporates the various conditions or metrics. This model must be a mathematical or logical representation of the concept.
- The conceptual model will then be implemented in network simulation software such as OMNeT++, OPNET or Matlab. Verification will occur throughout the computerisation process.
- The model will be verified and validated through the use of various validation and verification methods.
- Simulations will be conducted on the implemented model to obtain informative results. The simulation results will be analysed and discussed. The results will also provide an indication regarding the correctness of the hypothesis.
- Finally, the development procedures, implementations and simulation results will be concluded in a final chapter.

1.9 Research methodology

A scientific method for full disclosure is proposed in [6] by Gauch. The proposed model is known as "PEL". The author states that a combination of presuppositions (P), evidence (E) and logic (L) results in scientific conclusions. According to Gauch, presuppositions refer to beliefs we as humans encompass in this world. Evidence refers to the things we perceive. And lastly, logic uses presuppositions to draw conclusions from the acquired evidence.

The logic used to derive conclusions are either deductive or inductive of nature.

In the case of deductive logic, as depicted in figure 1.1, a hypothesis is stated from theory followed by observations to determine the correctness of the hypothesis. A deductive argument is considered as valid providing the constituting foundations guarantee the truth of the conclusion.

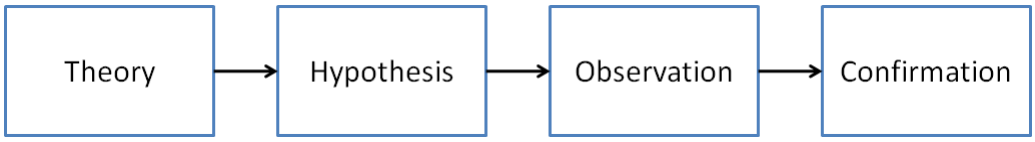

Figure 1.1: Deductive research logic

In the case of inductive logic, as depicted in figure 1.2, we start with observations, leading to patterns and generalisations, from which the hypothesis follows, and conclusions with regard to theory. An inductive argument is considered as strong when the constituting premises support the truth of the conclusions to an acceptable degree.

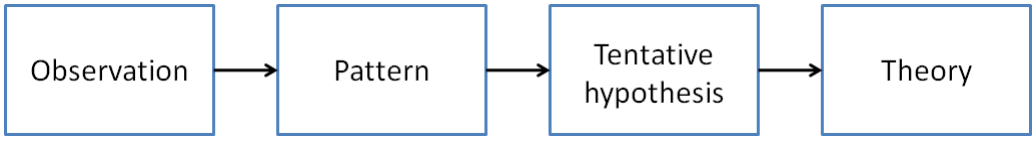

Figure 1.2: Inductive research logic

In this research we will apply **deductive** logic. Our research methodology, therefore, incorporates this type of logic. When deductive logic is applied to a research problem,

it results in a research approach depicted in figure 1.3.

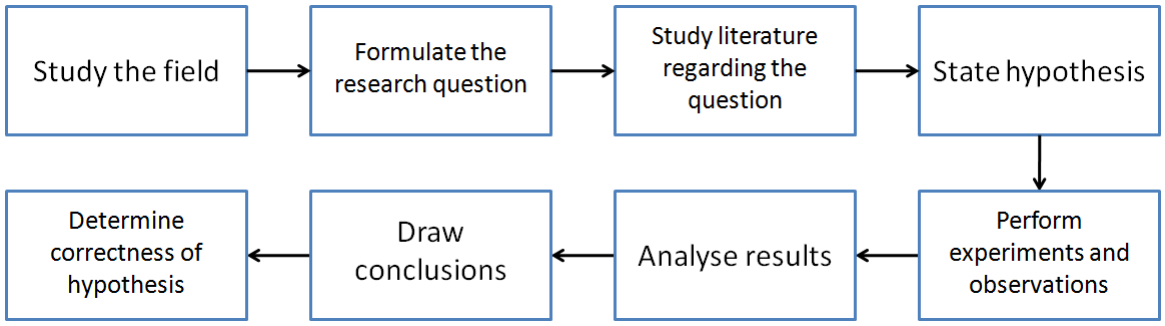

Figure 1.3: Research methodology

The process initiates with studying the field. In this case, a literature survey was conducted on OR, NC and GR. The literature survey then leads to a research question, in this case it led to the following question: How can we combine OR, NC and GR into a single specialised routing protocol and what would the consequences be?

A detailed literature study needs to be performed to show what is done, how it works and to indicate the challenges and issues involved in the process of answering the research question.

A hypothesis is then made of what is presumed the answer will be. The hypothesis leads to the testing of the idea through experiments and simulations to produce results.

The results are then analysed, conclusions are drawn, and the truthfulness of the hypothesis is revealed.

The deductive method, specifically applied to our research, is depicted in figure 1.5.

1.9.1 Validation and verification

Please refer to figure 1.4. The simplified validation and verification model from [1] will be applied in the research.

The problem entity refers to the idea or situation that we wish to model. The concep-

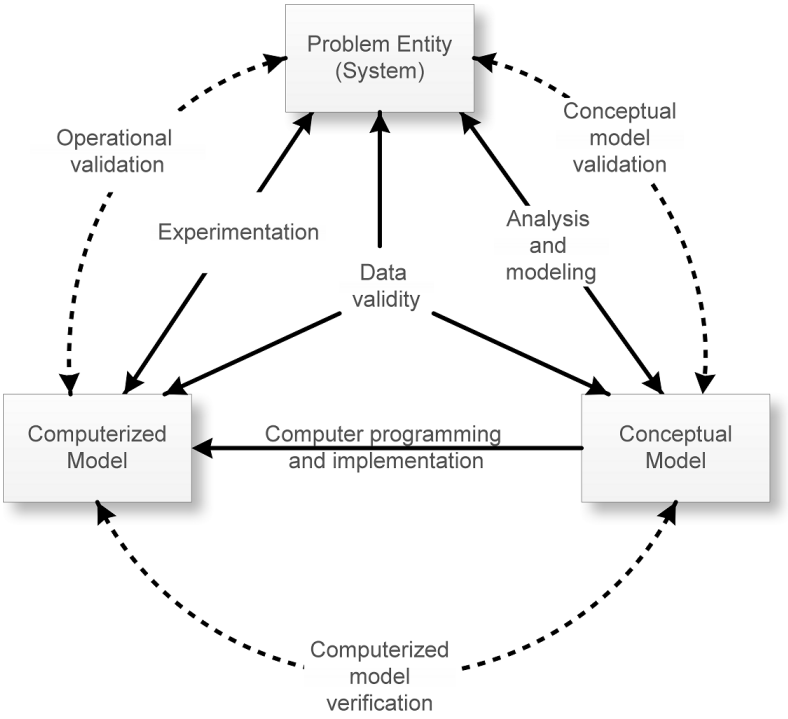

Figure 1.4: Simplified validation and verification model [1]

tual model is the mathematical or logical representation of the real problem. The model is created through analyses and modelling of the problem. The conceptual model is then validated by ensuring that the underlying theories and ideas behind the model are correct and that it is a valid representation of the real problem. If the model fails at the validation phase, the analysis and modelling phase is repeated to recreate the conceptual model. These two phases are repeated until the conceptual model is successfully validated.

The next step is to computerise the conceptual model. This is done by means of computer programming. The computerised model is verified by looking at the correctness of the programming carried out. If the computerised model fails at the verification phase, the computerised programming and implementation phase is repeated. These two phases are repeated until the computerised model is verified.

The computerised model is validated by ensuring that the output of the model is an accurate enough representation of the real problem as it was intended to be. The im-

plementation of the computer model is then used to conduct experiments. The experiment results are compared to the real problem and conclusions are drawn. Data validity refers to the correctness of the data used for building the model, testing and evaluation.

When the validation and verification model is incorporated into the research methodology model from figure 1.3, it results in the model depicted in figure 1.5.

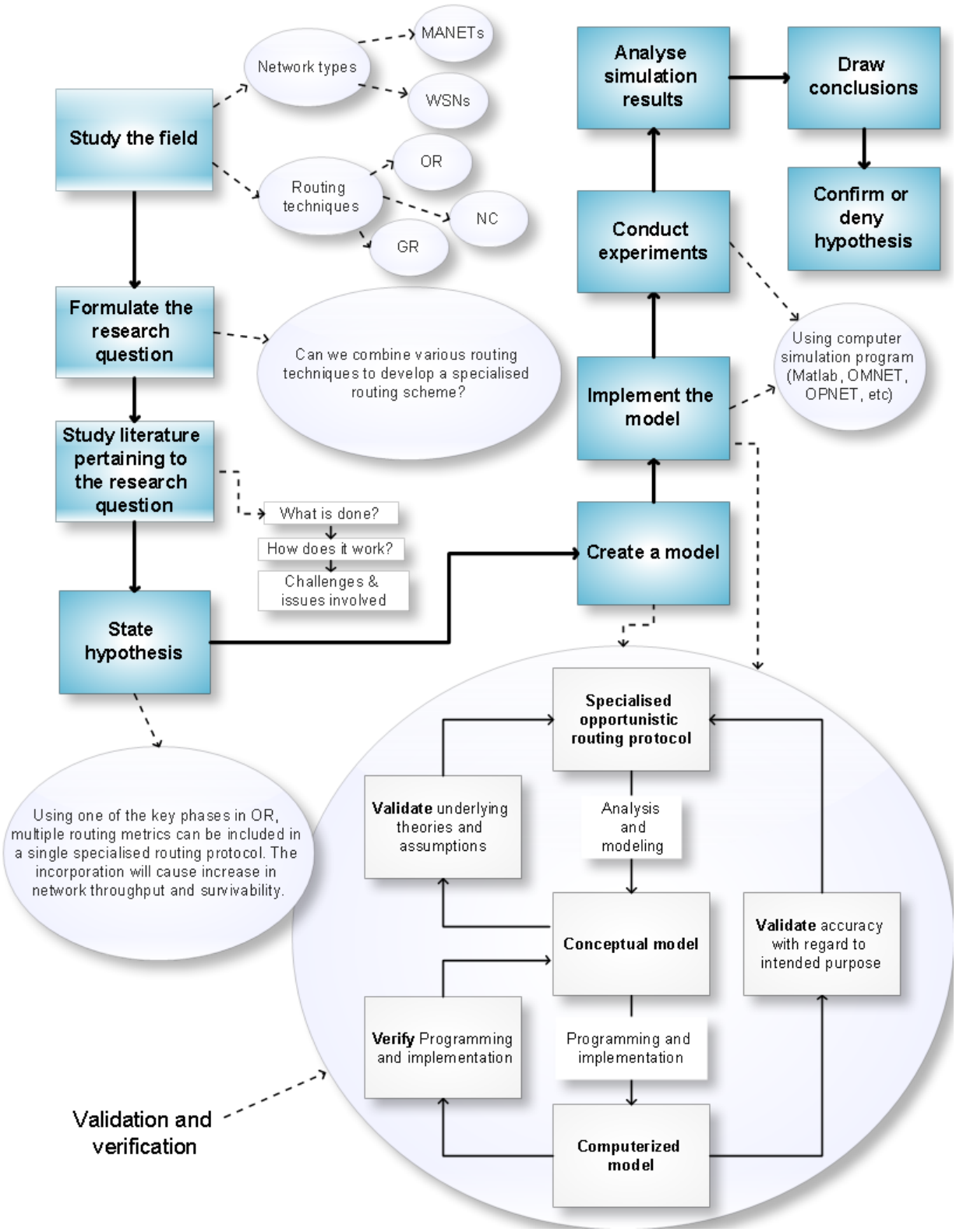

Figure 1.5: The research methodology model for this research

Validation and verification methods and how they are applied specifically in this research are discussed in chapter 5, *Validation and Verification*.

1.10 Dissertation overview

The remainder of the dissertation is constructed as follows: Chapter 2 details an in-depth literature study of topics relevant to the research question as well as aids in the development process in the chapter to follow, chapter 3. The implementation of the developed scheme and details surrounding the simulation software environment is discussed in chapter 4. The implementation is verified and validated in chapter 5. Furthermore, experiments are conducted in chapter 6. Lastly, concluding remarks of the conducted research and suggestions for future work are provided in a final chapter, chapter 7.

1.11 Conclusion

In this chapter, we explored some background regarding WANETs and routing methods. We discovered a new area for research and provided motivation and justification. We then presented a detailed layout of the planned research conducting methods to be applied.

Chapter 2

Literature Study

In this chapter, we will conduct an in-depth literature study of issues pertaining to the proposed research.

2.1 Introduction

We will begin this chapter by reviewing wireless networks. Furthermore, network characteristics and performance metrics pertaining to these wireless networks will be discussed. Generic network models will be broken down into their core functioning components. The remainder of the chapter will detail NC, GR, OR and energy aware routing.

2.2 Wireless Networks

Wireless networks can be divided into two main categories namely wireless infra-structured networks and WANETs. They are discussed below.

2.2.1 Wireless Infra-structured Networks

Wireless mesh networks

Wireless Mesh Networks (WMNs) are ad-hoc networks that are self-established and self-maintained by the nodes of which it is comprised. WMNs consist of mesh clients and mesh routers. Mesh routers form the backbone of the WMN and enables communications between different wireless network access technologies. Mesh clients only have single access technology interfaces and do not have bridge or gateway functions. This allows for simpler hardware and software inside a mesh client than a mesh router. Nodes with wireless Network Interface Cards (NICs) can gain access to a WMN by connecting wirelessly to the mesh routers. Nodes without wireless NICs can also gain access to a WMN by connecting to a mesh router through other access technologies, for example, Ethernet [7].

2.2.2 Wireless Ad Hoc Networks

WANETs are decentralized networks formed by wireless nodes. WANETs are self-established and do not depend on any infrastructure. Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs) are examples of WANETs [8]. Some Wireless Sensor Networks (WSNs) can also be considered as WANETs when the sensor nodes operate in an ad hoc manner [9]. Nodes in these networks communicate in an ad-hoc manner without the use of a centralised controlling or managing unit [2].

Advantages and disadvantages of WANETs are classified a follows [10]:

Advantages

- Flexible with regard to network topology changes
- Cheaper implementation cost compared to wired networks
- Effective for use in hard-to-reach places
- Easily adaptable in infrastructure-less environments

Disadvantages

- Lower speed compared to wired networks
- Extremely vulnerable to various attacks seeing that communication occurs over an open medium
- In some networks such as MANETs, nodes have limited energy resources

We will now discuss specific WANETs such as MANETs, VANETs and WSNs.

MANET

MANETs are ad-hoc networks that are self-established and lack infrastructure [8]. All of the nodes in MANETs are mobile devices such as laptops, Personal Digital Assistants (PDAs) or cellphones. Topologies in the network may change randomly at any given time because of the unpredictable mobility of the nodes.

The fact that the nodes are mobile also indicate that they have limited energy and computational resources as well as limited transmission ranges [11]. Limited energy resources necessitate custom routing protocols dedicated to preserving energy. Due to the limited transmission ranges within MANETs it is common for packets transferred in MANETs to hop between a number of nodes in order to reach their destinations; Therefore, usually a number of nodes have to cooperate to achieve successful communication [8].

VANET

VANETs are a subset of MANETs. VANETs are ad-hoc networks formed between vehicles. The networks are formed randomly without any need for a centralized structure. Vehicles are enabled to become nodes in a wireless network by equipping them with wireless communication devices. The mobility of nodes in VANETs result in nodes entering and leaving networks randomly whilst changing positions. VANET topologies are, therefore, continuously changing.

VANETs are currently established for various purposes such as to increase safety and comfort, and in some cases even to provide entertainment [12]. Evidently inter-vehicular communications will lead to many more uses and utilities.

WSN

The primary use for WSNs is to retrieve real-time information from the environment such as temperature, geographical characteristics and particulate matter (e.g. CO_2). A WSN consists of sensor nodes and base stations (sinks). The sensor nodes are responsible for acquiring data from the environment, processing the data and transmitting information to the base stations [13]. The base stations integrate and combine the data collected from the sensor nodes to produce useful information. Sensor nodes are usually battery powered, and it is therefore essential that WSNs operate and communicate in an energy-efficient manner in order to keep the nodes alive for as long as possible [14, 15]. Environment, medical healthcare and military monitoring are only few of the fields included in WSN applications.

2.2.3 Network characteristics

Each type of network encompasses its own set of unique network characteristics. Network characteristics include properties such as node mobility and energy-limitations. Some routing protocols are tailored to serve characteristics encompassed by a network.

Energy-constrained nodes

MANETs and WSNs are wireless communication networks that consist of nodes with limited energy resources [11, 14]. Energy-constraint nodes need energy to transmit, receive and process data. When such a node is drained from all its energy resources, it cannot perform any communication functions and is, therefore, of no use to the network, or in the case of user operated nodes, the operating client. It is imperative that nodes in such networks remain alive for as long as possible, especially in WSNs where each node is responsible for monitoring a specific area or item [16].

When routing protocols do not take into account the energy levels of the nodes problem

scenarios may arise. One of such problem scenarios that may occur is node isolation. This concept is depicted in figure 2.1. In this scenario, nodes *A* and *C* are depleted of energy resources. This situation isolates node *B* from the network.

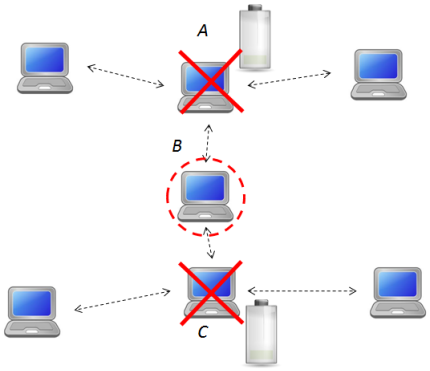

Figure 2.1: Node isolation

It is quite beneficial for routing protocols implemented in such networks to be aware of the energy levels of the nodes as it enables them to route energy effectively. Energy aware routing in networks refers to the monitoring of the energy levels of nodes, or the calculation of the amount of energy required per transmission, and using the acquired information as deciding factors when performing routing.

Mobility

This dynamic characteristic exists in ad hoc networks such as VANETs and MANETs. It refers to the never-ending change in node positions, delivering in turn dynamic network topologies to networks. Mobile nodes have limited energy resources.

Broadcast characteristics

When a wireless node, such as node *A* in figure 2.2, transmits a signal, any wireless node close enough to the transmitting node has a certain probability *P* of overhearing the transmission [3]. This is a common broadcast characteristic of any wireless

network. Routing techniques such as OR exploits wireless broadcast characteristics to increase network performance.

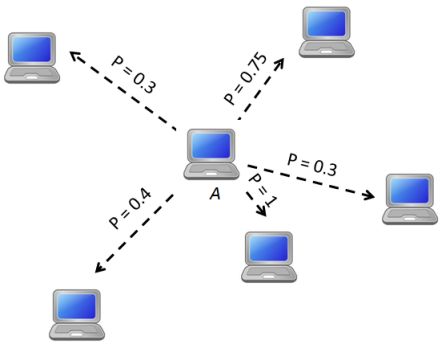

Figure 2.2: Wireless probability

2.2.4 Network performance

Network performance refers to measuring how well a network performs with regard to the objectives it requires to adhere to. The importance of specified objectives vary with the characteristics of the network.

Throughput

The throughput of a network refers to the number of packets or the amount of data successfully delivered to their intended destinations in a time interval [17]. It is a representation of how fast data can successfully travel from source to destination nodes through a specific network.

Delay

Delay is the measured time period from when the first bit of the message is transmitted at the source, to when the entire message is received at the destination [17]. Network

delay is comprised of propagation delay, transmission delay, queuing delay and processing delay.

Survivability

Survivability is a performance metric that applies to energy-constrained networks. Survivability refers to the lifespans of nodes with limited energy resources. Energy of a node is drained through transmissions, movement and computation [18]. When a node is depleted of energy, it cannot generate or forward data and is of no use to a network. Death of nodes lead to problem scenarios in networks such as node isolation, broken paths and evidently poor network performance.

2.3 Network Models

In this section, generic network models are discussed. It is imperative to have a sufficient model that details the network system, especially considering the development of a routing scheme.

2.3.1 OSI Model

The Open Systems Interconnection (OSI) model is a universal representation of a network system. The model portrays the communication process in *seven* logical layers as depicted in figure 2.3 [17].

The purpose of the OSI model is to provide a framework for communication between different systems that can be universally applied in the presence of diverse hardware and software. The model, therefore, is valid regardless of the hardware and software of the environment where it is applied.

Please refer to figure 2.4. In this scenario, host *A* sends information to host *D*. The data

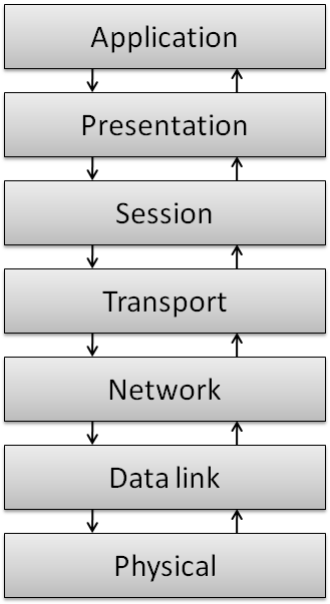

Figure 2.3: Seven layers of OSI model

is generated at the **application layer** at host *A*. It then travels downwards through all the layers to the **physical layer**. At intermediate nodes, in this scenario hosts *B* and *C*, the data usually only travels up three layers, to the **network layer**, and then back down to the **physical layer**. At the destination node, host *D*, the data will travel all the way up to the top layer, the **application layer**. Information exchange between layers are referred to as *intra*-communication while information exchange between OSI models of different nodes are referred to as *inter*-communication.

The seven layers will now be briefly discussed starting with the lowest layer, the **physical layer**.

Layers of the OSI model

Physical layer

The **physical layer** is responsible for the physical, electrical and mechanical aspects of the communication process [17]. It manages the processes required to send data over a

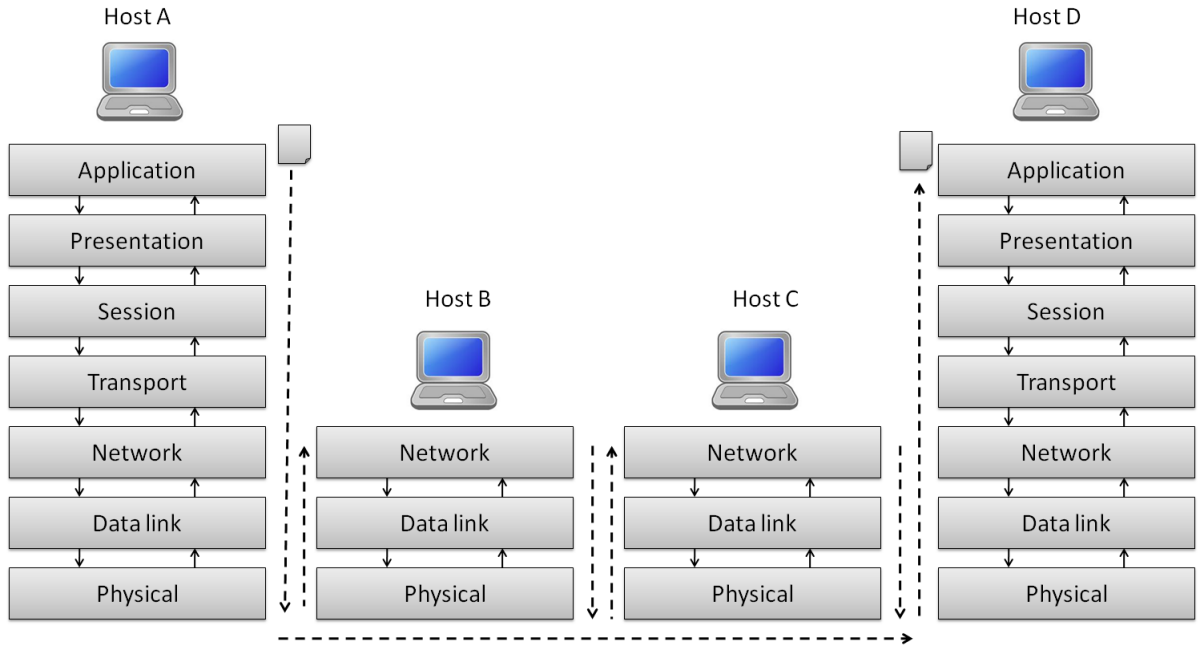

Figure 2.4: OSI model inter- and intra-communication

transmission medium. Please refer to figure 2.5. The physical layer receives bit streams of information from the data link layer. It then converts the bit streams into electrical or optical signals in order to send them over the transmission medium.

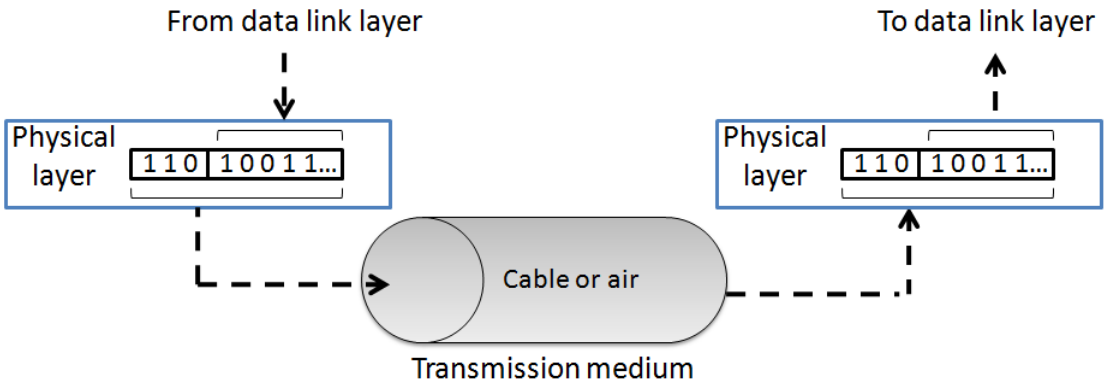

Figure 2.5: Physical layer

Data link layer

While the physical layer focusses on transmitting and receiving data with absolute regard to the transmission media, its focus is diverted from error control. The **data link layer** receives raw data from the physical layer and through processing thereof delivers presentable error-free data to the upper layers [17]. In the same sense, all data received from the upper layers are prepared for the physical layer by being divided into manageable data units called frames as depicted in figure 2.6.

The layer is also responsible for **flow control** and **access control**. Flow control refers to managing the flow of data. In doing so, it aids the receiver in scenarios where the incoming data rate is faster than that which the receiver can manage. Access control refers to the protocols regulating the access of devices to the communication medium.

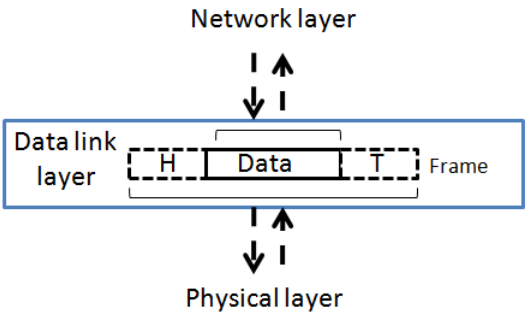

Figure 2.6: Data link layer frame

Network layer

The data link layer is responsible for hop-to-hop delivery while the **network layer** manages the source-to-destination delivery [17]. These delivery terms are depicted in figure 2.7. Hop-to-hop delivery refers to communication between neighbour nodes while source-to-destination delivery refers to the entire route the data travels from where it was generated to where it requires to be delivered.

It is, therefore, evident that the network layer is responsible for performing routing. It

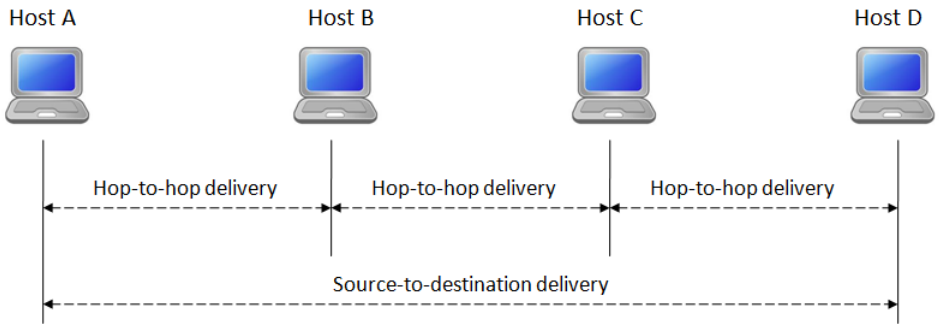

Figure 2.7: Delivery terms

determines the next-hop nodes along which the packets need to travel and in doing so controls the flow of data.

Transport layer

Although the network layer is responsible for source-to-destination delivery of individual packets, it remains uneducated regarding the relationship between those packets. The **transport layer** is responsible for delivering the entire message as a whole [17]. As depicted in figure 2.8, the transport layer divides the messages travelling downwards, towards the network layer, into transmittable segments, and reassembles segments received form the network layer in order to deliver complete messages to the session layer.

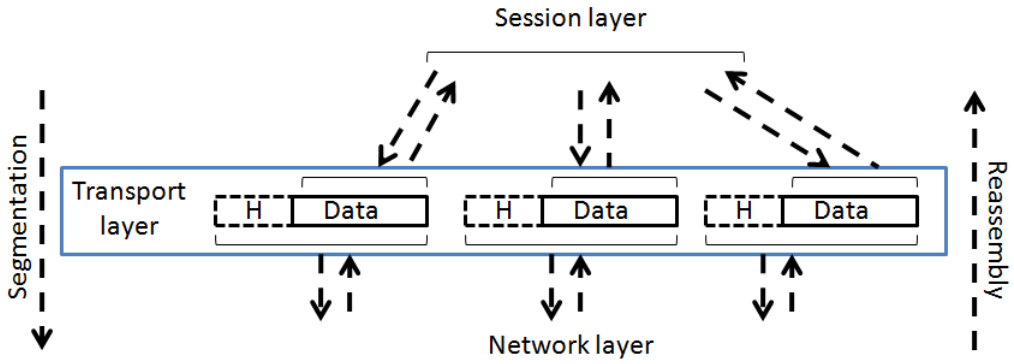

Figure 2.8: Segmentation and reassembly

Session layer

The **session layer** manages the **synchronisation** of data streams as depicted in figure 2.9. It inserts checkpoints into the data stream allowing it, when a crash occurs, only to have to resend data subsequent to the previous checkpoint [17]. This minimises the amount of data needed to be resent after communication failure.

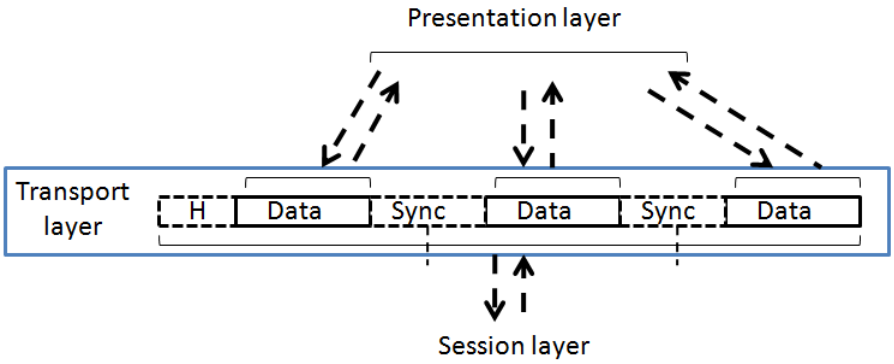

Figure 2.9: Synchronisation

Presentation layer

Information that requires to be exchanged between nodes are usually in a data form corresponding to the respective systems where it is generated [17]. Different computers encompass different data forms since they have different encoding systems. The information usually exists in the form of character strings, numbers, etc. The **presentation layer** is responsible for translating this specific information into bit streams, a more common form of data, as depicted in figure 2.10.

The presentation layer is also responsible for **encryption**. The layer encrypts sensitive, outgoing information and decrypts incoming encrypted information. And lastly, the layer performs **data compression** by reducing the number of bits required to represent data. Data compression is imperative especially when working with images, video and audio.

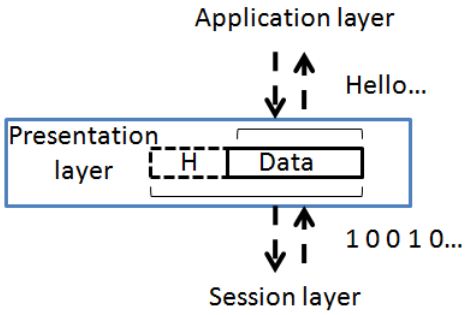

Figure 2.10: Presentation layer

Application layer

The **application layer** provides network access features to the end user, whether it is human or software [17]. It provides services such as file transfer and management, mail services and database services. In other words, it serves as an interface between the operating client and the network.

2.3.2 TCP/IP protocol suite

The TCP/IP protocol suite was established prior to the OSI model and does, therefore, not match the OSI model precisely. The model consists of *four* layers namely the link layer, internet layer, transport layer and application layer [19]. The similarity between the TCP/IP and OSI models are depicted in figure 2.11 [20] [17].

2.4 IEEE 802.11 standard

The IEEE 802 committee was established in 1980 with the main focus of developing standards for Local Area Network (LAN)s [21]. In 1990, they launched a new networking group,known as the 802.11, fully devoted to Wireless LAN (WLAN)s and the development of wireless routing standards. The 802.11 standard dictates the operation

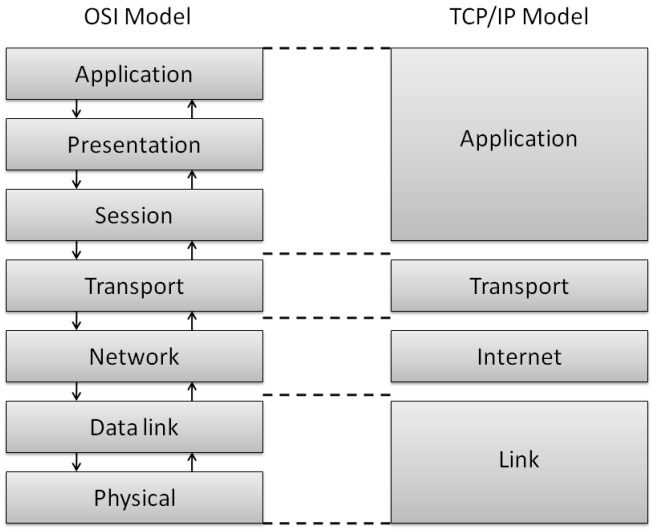

Figure 2.11: The OSI and TCP/IP models

of the lowest two layers of the OSI model namely the **physical** and **data link** layers. Furthermore, it divides the data link layer into two sublayers namely the **Logical Link Control (LLC)** and **Media Access Control (MAC)** sublayers as depicted in figure 2.12.

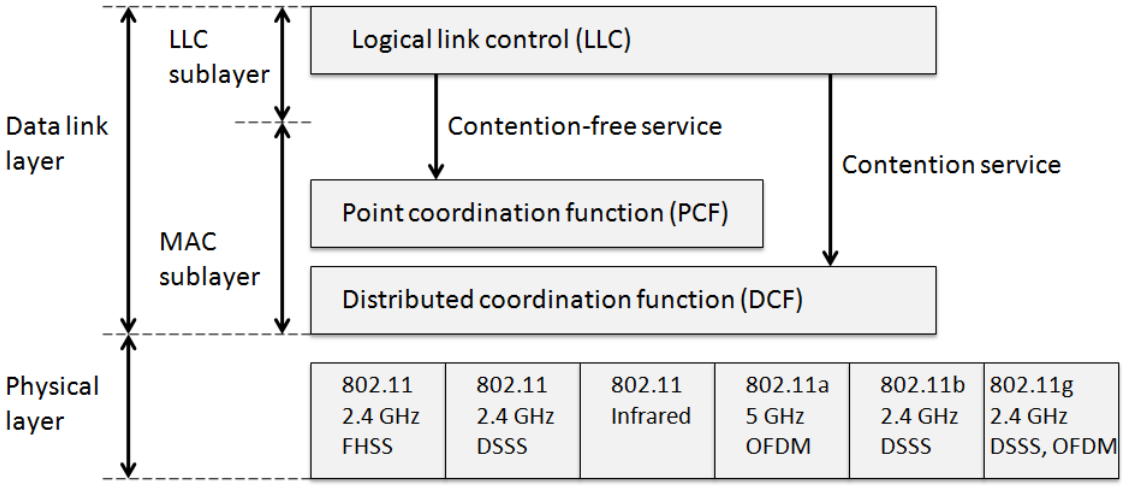

Figure 2.12: IEEE 802.11 layers

2.4.1 Data link layer

As described in the previous section pertaining to network models, the data link layer has certain responsibilities, some of which include **flow control**, **error control** and **access control**. The 802.11 standard divides the data link layer into sublayers that respectively undertake those responsibilities [17,21].

Logical link control

The LLC sublayer is responsible for **flow control** and **error control**. Its objective is to handle errors occurring between the physical layer and the upper layers as well as manage the flow of data [17,21].

Media access control

When two or more devices share the same medium, access to the medium needs to be managed [17]. The MAC sublayer handles the procedures regarding device access to the communication medium. The IEEE defines two protocols in this sublayer namely Distributed Coordination Function (DCF) and Point Coordination Function (PCF). PCF is only applicable in infrastructure networks, not in ad hoc networks. On the other hand, DCF is distributed and is used in ad hoc networks. It implements carrier sense multiple access with collision avoidance (CSMA/CA) as access protocol.

CSMA/CA

CSMA/CA regulates access of multiple devices to the communication medium and applies signal collision **avoidance** strategies [17]. In wireless networks, we aim to avoid collisions rather that detecting them. In wired networks, carrier sense multiple access with collision detection (CSMA/CD) can be implemented since it is possible to detect collisions by measuring the energy levels of the medium. In wireless networks, signals

lose significant amounts of energy during transmission rendering it, therefore, difficult to detect collisions.

CSMA/CA avoids collisions by implementing timing strategies as depicted in figure 2.13. The flow diagram of the process is depicted in figure 2.14. When a node has data to send, it **continuously** senses the channel until it becomes idle. Before it transmits, it waits for a time period known as the **Interframe Space (IFS)**.

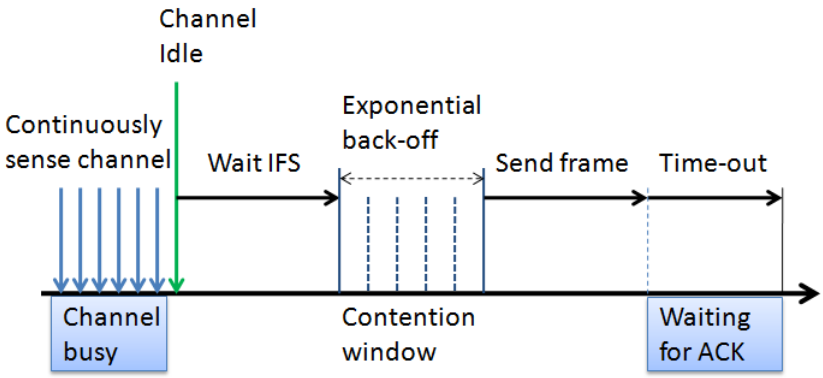

Figure 2.13: Timing strategies in CSMA/CA

IFS

Even when the channel appears **idle**, a distant node may have already transmitted data not yet received by this node due to propagation delay [17]. Hence, the IFS provides a time window allowing signals from distant nodes to reach this node. The IFS can also be used to assign priorities to nodes or frames. Nodes or frames with shorter IFSs will be enabled to transmit earlier. When the IFS timer expires, it again senses the state of the channel. If the channel is busy, it must return to the start of the entire process. If the channel is idle, the process continues to the next phase called the **contention window**.

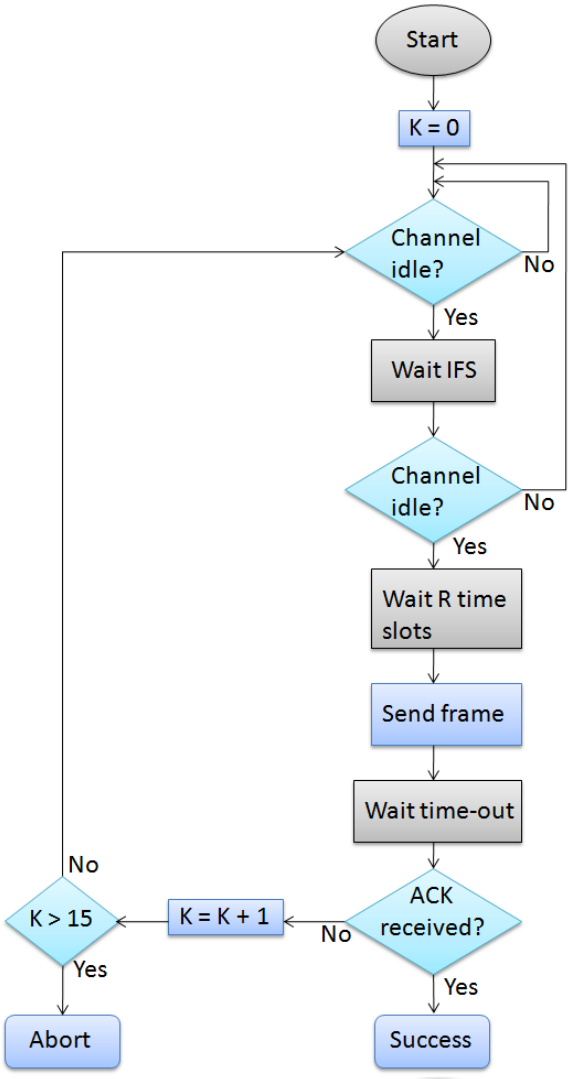

Figure 2.14: Flow diagram for the CSMA/CA

Contention window

The contention window is a time period divided into equivalent time slots as demonstrated in figure 2.13 [17]. It works on the concept of Binary Exponential Back-off (BEB) [22]. After every failed transmission, the contention window will exponentially increase in size. As depicted in the flow diagram, figure 2.14, the process chooses a random number **R** between zero and $2^K - 1$. At first this value is zero since K is initiated as zero. Every time a transmission fails, K is incremented, and the process repeats itself. This procedure is known as exponential back-off.

According to the procedure, after each time slot the node must first sense the channel. If the channel is busy it must freeze, and only continue when the channel resumes idle. Statistically, this process gives higher priority to nodes with longer waiting times.

Acknowledgement

After a frame is sent, the node waits for a specified time period known as the **time-out**, in which a transmission is considered successful if an acknowledgement is received in that time window. A transmission is regarded as unsuccessful if the time-out period expires without the reception of an acknowledgement [17].

CSMA/CA with handshaking features

The primary focus of CSMA/CA is for use in wireless networks. However, the process discussed in figure 2.14 is insufficient in some cases regarding wireless communication such as the hidden node or exposed node scenario [17]. These two scenarios are discussed in the following sections, after which we will take a look at the proposed solution, including concepts such as handshaking and Network Allocation Vector (NAV).

Hidden node scenario

A hidden node situation is depicted in figure 2.15 [17]. This type of situation occurs when a central node *2* is aware of its two neighbour nodes *1* and *3*, but neighbour nodes *1* and *3* are unaware of each other. This phenomenon causes nodes *1* and *3* to, from time to time, unknowingly transmit simultaneously leaving node *2* unable to receive either transmissions due to interference.

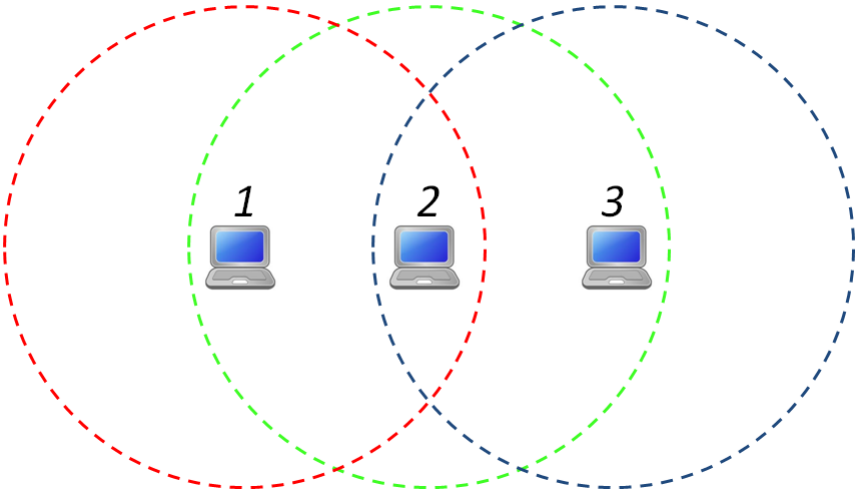

Figure 2.15: Hidden node scenario

Exposed node scenario

An exposed node scenario refers to a situation where a node is unable to communicate due to communication occurring within its communication range [17]. This concept is depicted in figure 2.16. When node *2* is busy transmitting data to node *1*, node *3* is unable to communicate to node *4* as it would cause interference with transmissions from node *2*.

Figure 2.16: Exposed node scenario

Handshaking

Handshaking incorporates processes such as Request to Send (RTS) and Clear to Send (CTS) into the CSMA/CA procedure [17]. Please refer to figures 2.17 and 2.18. If a node has a message to transmit, it first senses the energy level at the carrier frequency as also done in the conventional CSMA/CA procedure. After the persistence strategy

succeeds in finding the channel idle, it waits for a time period called the Distributed Interframe Space (DIFS). It then sends a control framed called the RTS and waits for a **time-out** time period. The node that received the RTS waits for a Short Interframe Space (SIFS) time period and then replies with a CTS. When the sender node waits for a SIFS time period after receiving the CTS and then transmits the data. The destination node replies with an Acknowledgement (ACK) after receiving the data and waiting a time period equal to the SIFS.

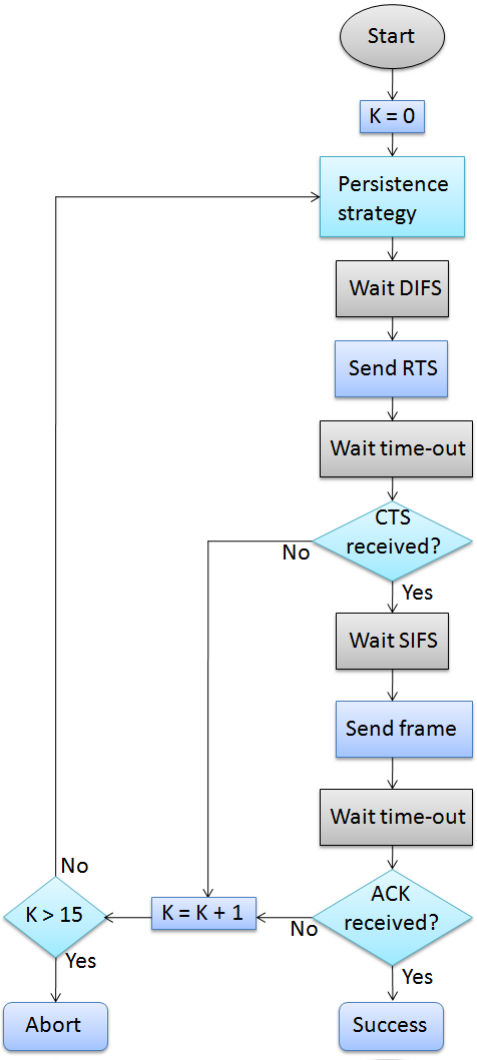

Figure 2.17: CSMA/CA with handshaking

NAV

Included in the RTS control frame is the time duration that the node requires to occupy the medium [17]. All other stations, excluding the source and destination nodes, that receive the RTS control frame create NAV timers that prevent nodes to sense the channel for a NAV time period. This concept is depicted in figure 2.18.

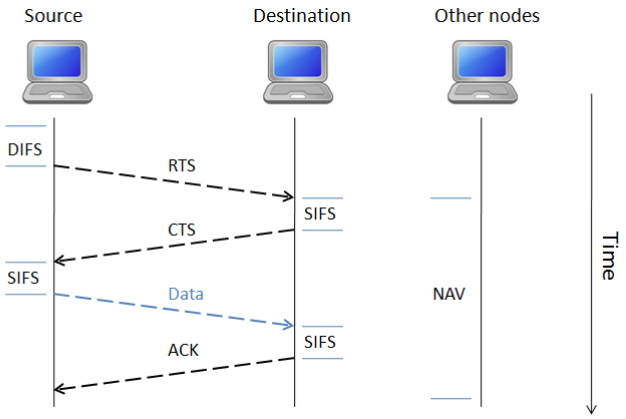

Figure 2.18: Handshaking

2.4.2 Physical layer

The physical layer covers aspects such as the encoding method, frequency band and data rate [17].

The developed 802.11 standards and their specifications are summarised in Table 2.1 [17, 21].

2.5 IEEE 802.15.4 standard

The IEEE 802.15.4 standard is a protocol developed to promote low energy usage in wireless networks, and it operates at low data rates [23, 24]. It is, therefore, ideal for

Table 2.1: Physical layer 802.11 standards

Standard	Date	Technique	Frequency band	Data rate (Mbps)
802.11	1997	FHSS	2.4 GHz	1,2
		DSSS	2.4 GHz	1,2
			Infrared	1,2
802.11a	1999	OFDM	5.725 GHz	6 to 54
802.11b	1999	DSSS	2.4 GHz	5.5 and 11
802.11g	2003	OFDM	2.4 GHz	22 and 54

application in limited resource networks such as WSNs where data rate is not of imperative concern. Although this standard produces significant results with regard to decreased power consumption, it lacks performance in numerous categories such as throughput, packet delivery and average end-to-end delay when compared to the traditional 802.11b WIFI protocol [24]. And since the aforementioned categories are important factors in the proposed research, the 802.11 standard is the preferred protocol in this case. Additional support regarding this choice of standard will be provided in later sections.

2.6 Geo-Routing

GR refers to routing based on the geographical coordinates of nodes [5]. Geo-distance between nodes refers to the geographical distance between nodes. Nodes with a smaller geo-distance to the destination node, in comparison with the sender node, forward the packets. GR can only be implemented in networks where the nodes know their coordinates. Nodes can acquire geographical coordinate information by GPS or localisation techniques.

Please refer to figure 2.19. This is an example of a network in which the nodes are aware of their geographical coordinates. These coordinates are used to calculate dis-

tances between nodes and enable them to perform routing decisions based on these geo-distances.

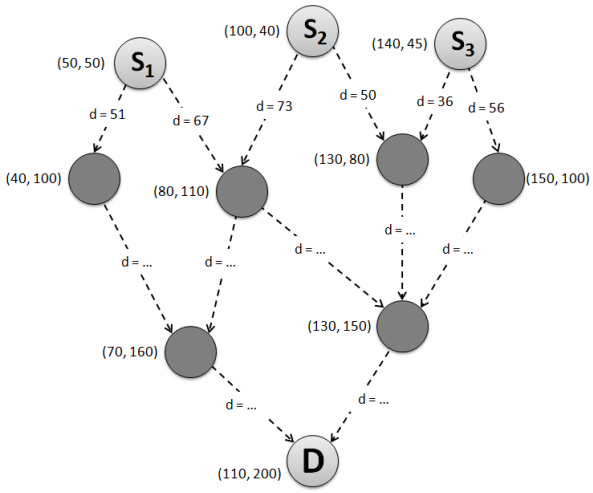

Figure 2.19: Geo-routing

2.7 Energy aware routing

Energy aware routing can be divided into two main categories namely, transmission control and path selection.

In the case of transmission control protocols, the scheme manages the power of the transmissions based on the signal strengths required to successfully communicate [25]. Since our routing protocol utilises OR, we want to keep the transmission radius at a constant magnitude as the more nodes hear the transmission, the more effective and robust the routing protocol becomes. This phenomenon will become clear in later sections regarding OR.

In the case of path selection protocols, the scheme selects the best path with regard to specified conditions. These conditions usually refer to choosing the path that requires the minimum amount of energy to successfully transfer the data to the destination node [26], or choosing the path that contains the nodes with the highest energy re-

sources [27]. The drawback associated with the former method is that nodes on those minimum energy paths get burned, which eventually results in disjointed networks.

In the case of the latter, the nodes with higher energy levels are utilised, allowing nodes with lower energy statuses to live longer, hence increasing the survivability and coverage time of the network.

2.8 Network Coding

NC is a technique that attempts to optimise the throughput of a network by coding packets [4]. This technique enables us to transmit multiple packets in a single transmission; hence transmitting more data for the same number of transmissions or requiring fewer transmissions to send the same amount of data when compared to traditional routing protocols.

Please refer to figure 2.20. In this scenario, nodes *A* and *C* requires to exchange information. Traditional routing would require **four** transmissions to accomplish this objective as depicted in the figure. When NC is implemented, the two packets **A** and **C** are coded at node *B* and are distributed to both nodes *A* and *C* in a final transmission, resulting in a total number of **three** transmissions as depicted in figure 2.21

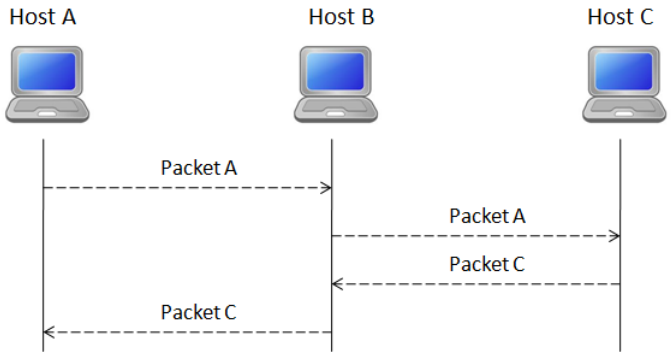

Figure 2.20: Two-hop traditional routing

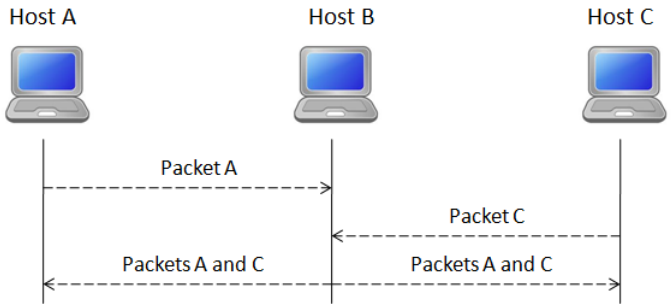

Figure 2.21: Two-hop NC routing

NC increases network throughput to the theoretically maximum achievable throughput as can be calculated using the min-cut max-flow theorem [4,17]. The theorem states that the maximum throughput achievable in a network is equal to the minimum cut between the source and the destination nodes in a network, that if removed, results in no flow capabilities between the source and destination nodes. The theorem is explained in figure 2.22. All the links in the example have unit capacity. In this scenario, source node S sends packets **a** and **b** to destination nodes X and Y. It can be seen from the dotted lines that the minimum cut between source node S and destination nodes X and Y, is **two**. Therefore, the maximum throughput obtainable in this network is **two**. In this scenario, after a full round of transmissions, **three** packets reached the destination nodes. **Two** packets were sent. This translates to a suboptimal throughput of **1.5** (3/2).

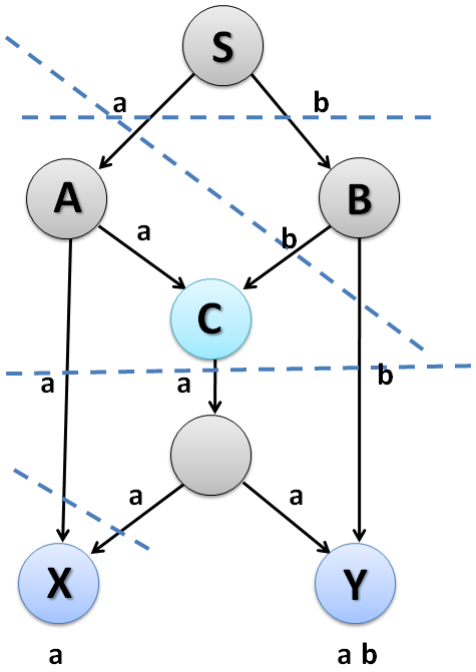

Figure 2.22: Min-cut max-flow theorem

When NC is implemented, the throughput increases to **two** as shown in figure 2.23. In this scenario, packets **a** and **b** are encoded at node *C*. The coded packet is sent in a single transmission to node *D*, and delivered to the destination nodes *X* and *Y* in the next transmission. The destination nodes then use the packets they have overheard from previous transmissions in order to decode the NC packet. More detail pertaining to the coding and decoding processes follow in later sections. In this scenario, **four** packets have reached the destination nodes, and it translates to a throughput of **two** (4/2), which is the maximum obtainable throughput of the network.

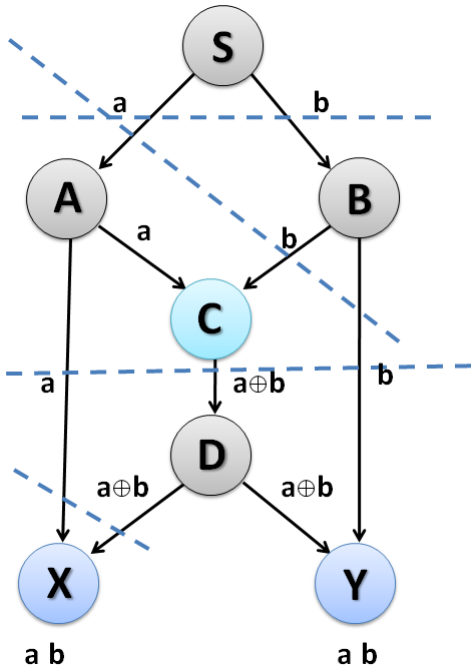

Figure 2.23: Network with Network Coding

From this example it is evident that one of the core benefits of NC is to maximise network throughput. The benefits entailed by NC follow.

2.8.1 Benefits of NC

Maximises throughput [28, 29]

NC achieves the theoretically maximum achievable throughput as specified by the min-cut max-flow theorem as aforementioned. It is able to achieve this maximum throughput since it is able to combine data streams. As depicted in figure 2.24, the image on the left illustrates traditional non-coding transmission, while the image on the right illustrates the same process when NC is applied. NC eliminates the bottleneck effect by sending both data streams over a single link simultaneously.

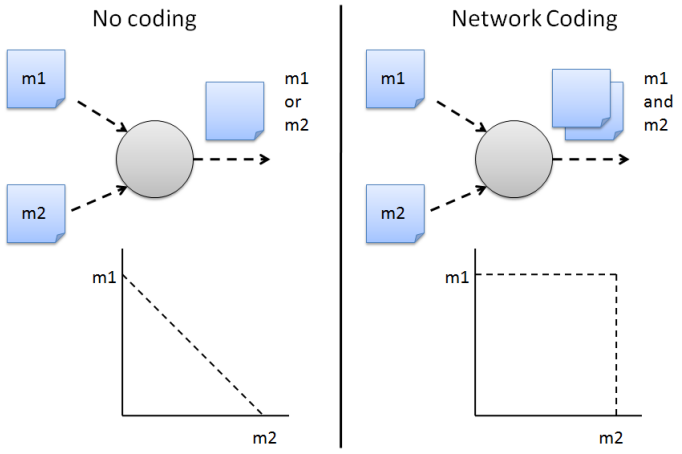

Figure 2.24: Packet combining

Minimises energy usage [28]

The same concept depicted in figure 2.24 applies here. Every transmission costs energy. Being able to send more data simultaneously over a single link results in fewer transmissions equating less energy use.

Minimises delay [28]

As depicted in figure 2.25, node *S* sends two messages *a* and *b* to nodes *1*, *2* and *3*. Traditional routing would accomplish this objective with a delay of **three**. Here, the term "delay" simply refers to the number of hops required to distribute the messages to the destination nodes. The number of hops entailed by the blue and yellow paths individually are **three** as depicted in figure 2.25(a), and therefore the delay is **three**. When NC is applied, messages *a* and *b* can be coded at source node *S* creating the possibility of routing the packets with a minimum delay of **two** depicted in figure 2.25(b)

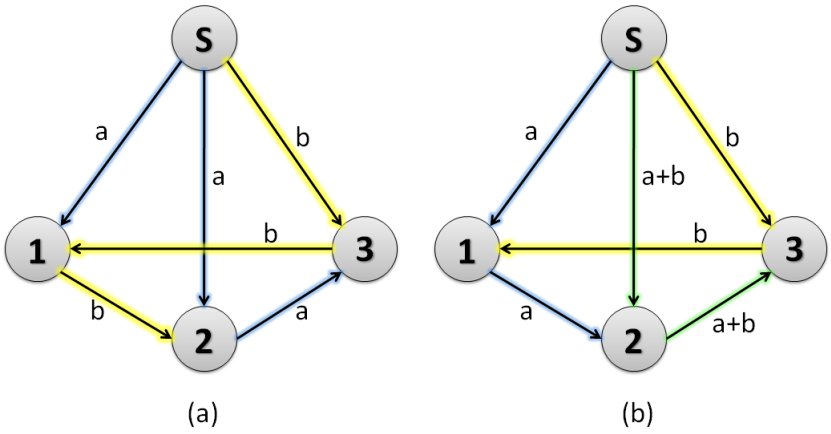

Figure 2.25: Minimising delay

Robustness and adaptability [29]

NC enhances the robustness and adaptability of the networks where it is applied since even if the intended destinations do not receive the transmitted data, active nodes that did, in fact, overhear the transmission can use the newly found information.

2.8.2 XOR operation

NC optimises the throughput of a network by coding packets [30]. We will now take a look at how the packets are coded and decoded.

Please refer to figure 2.26. In this example, similar to the one explored in chapter 1 *introduction*, hosts *A* and *C* exchange information. Packets *P1* and *P2* are XORed at relay host *B*.

The packets are coded by XORing them together as depicted in figure 2.27. The coded packet can then be decoded by XORing it with the sufficient number of packets it is comprised of. In this case, packet *P1* or packet *P2* would suffice. If either of the individual packets is XORed with the coded packet, it would result in the other individual packet; hence decode the packet. This concept is depicted in figure 2.28.

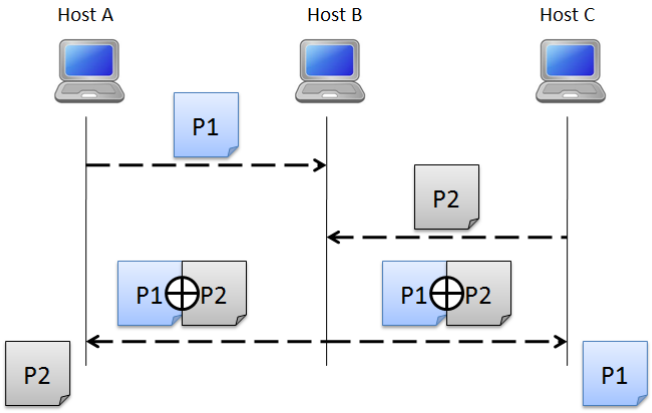

Figure 2.26: Network Coding scenario

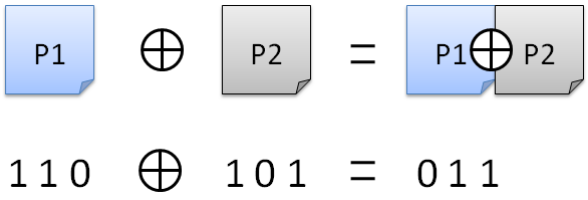

Figure 2.27: XOR coding operation

In mathematical terms, let p_i represent a **native** packet. The term *native packet* refers to a non-coded packet. Let p^c represent a coded packet. When n packets, $p_1, ..., p_n$, are coded together to form coded packet p^c, packet p_j can be retrieved by XORing $n-1$ packets p_i with the coded packet p^c, given $i \neq j$.

2.8.3 Reverse carpooling

The two-hop NC example discussed in chapter 1, *introduction*, indicated how the number of transmissions required for communication can be reduced. This idea can be generalized to an entire route of bidirectional information flow and is known as reverse carpooling. This concept is introduced in [31] as depicted in figure 2.29.

NC can furthermore be divided into two categories namely Deterministic Network Coding (DNC) and Opportunistic Network Coding (ONC) [32]. In DNC, coding op-

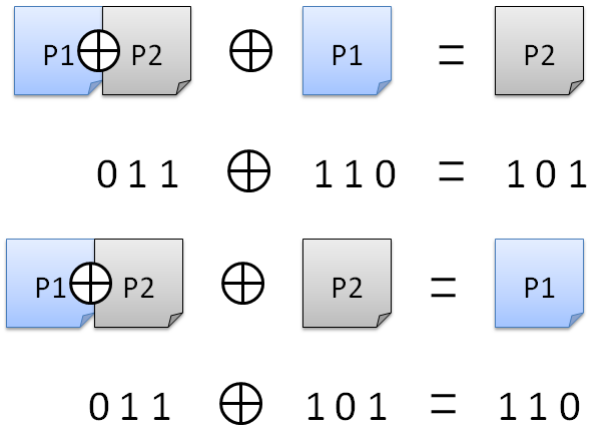

Figure 2.28: XOR decoding operation

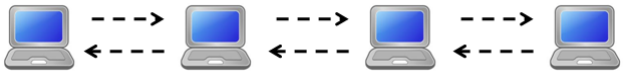

Figure 2.29: Reverse carpooling

portunities are determined beforehand, and nodes are then assigned encoding and decoding roles. DNC is, therefore, heavily dependant on the topology of the network. In ONC, nodes calculate the local coding opportunities dynamically using the available information acquired from neighbour nodes.

The two methods, DNC and ONC, are discussed in the following sections.

2.8.4 Deterministic Network Coding

In the case of DNC, coding opportunities are identified beforehand, and fixed nodes are assigned encoding or decoding roles [32]. This eliminates the need for coding vectors seeing that decoding nodes will always have prior knowledge of how the packets were encoded. DNC is therefore heavily reliant on the topology of the network, and it is therefore a challenge to implement DNC in WMNs, of which the topology may change frequently. In 2011 Grobler and Helberg [32] suggested a method to identify opportunities for the implementation of DNC in WMNs. In their work, they identify

the opportunities, by searching for known "NC suitable sub-topologies" within the larger network topology.

Implementation of DNC

Depending on the positions of the nodes and the connections formed inside a WMN, certain "known" NC suitable topologies can be used to identify where and how NC can be implemented [32]. In doing this, DNC can be implemented opportunistically each time a "known" NC suitable topology is formed as a sub-topology in a WMN. Certain coding opportunities will arise depending on the data transfer between relevant nodes that cause a flow of multiple data packets through nodes with encoding and decoding capability.

Identification of Topologies

In [32], three different methods were introduced to find the sub-topologies within a larger network: Iterative looping, Cross-correlation and Concatenation. Typical sub-topologies that the authors of [32] used include the "bow-tie" and the "butterfly" topologies depicted respectively in figure 2.30(a) and figure 2.30(b).

2.8.5 Opportunistic Network Coding

In the case of ONC, for example, Random Linear Network Coding (RLNC), packets arriving at an encoding node, are randomly combined as linear combinations of the original packets [33]. This type of NC is independent of the topology of the network because a coding vector is added by the encoding node to the header of the encoded packet. The receiver node then waits for a sufficient number of linearly independent packets to reach it, and then decodes the original packets.

In ONC, NC opportunities are calculated dynamically at each forwarder node by using

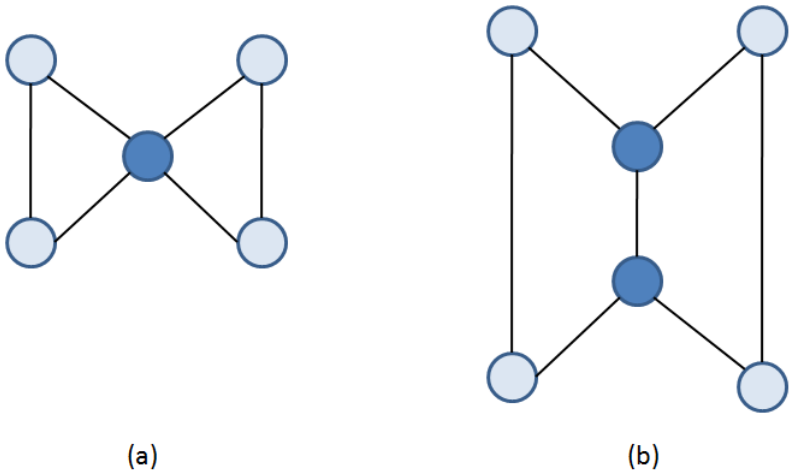

Figure 2.30: Network coding topologies a) Bow-tie b) Butterfly

the acquired neighbour information [30, 34]. This type of coding scheme is very effective in MANETs since the network topology of such networks are dynamic. COPE and CORE are also examples of ONC.

2.8.6 COPE

Basic Operation

COPE: "Practical Wireless Network Coding", presented in [30], is the first practical implementation of ONC. The NC scheme proves to deliver significant throughput gain as opposed to traditional routing schemes, in some cases even up to 400% throughput increase. COPE employs three key elements namely *opportunistic listening*, *opportunistic coding* and *learning neighbour state*. The three components work as follows.

In wireless networks, any node close enough to a node that transmits, may overhear the transmission. This broadcast characteristic is exploited by enabling nodes to store and process those overheard packets even if the receiving node is not the intended next-hop. Nodes can then use the overheard packets to decode coded packets. Each node is configured to temporarily store all overheard packets for a default time of T =

0.5s. This is known as the *opportunistic listening* phase.

Opportunistic coding refers to the intelligence in coding as many packets as possible that all of the specified next-hop destination nodes can decode the coded packet and retrieve the **native** packet intended for them. A **native** packet refers to a single non-coded packet. When a node calculates coding opportunities, it searches for the combination of packets delivering the most **code gains**. In COPE, the **code gain** refers to the ratio of the number of transmissions required by traditional routing, to the number of transmissions required with NC implemented.

In order to discover and perform the best possible coding combination in opportunistic coding, it is essential that the coding node knows which packets each of its neighbour nodes has. This concept refers to *Learning neighbour state*. Reception reports are appended to data packets. The reports contain information regarding the stored packets at the sender node. In this way, nodes can learn what packets are stored at their neighbours and are enabled to perform optimal coding accordingly.

Please refer to figure 2.31 for an example on COPE. By means of reception reports, source node *S* knows which packets are stored at its neighbour nodes. Using this information, node *S* can calculate the pristine combination of XORed packets to achieve maximum code gains. In this scenario, node *S* would achieve maximum code gains by coding packets *P1*, *P2* and *P3*.

The coded packet is broadcast to the neighbour nodes as depicted in figure 2.32. The packets in the coded packet are then cross-referenced with the packets stored at the respective node as illustrated in the figure. The decoded packet at each node is indicated by the circled packet in the coded packet. In this case, it can be seen that all three nodes receive a native packet intended for them.

This scenario corroborates the effectiveness of NC giving that one single transmission distributed three packets to their intended destinations.

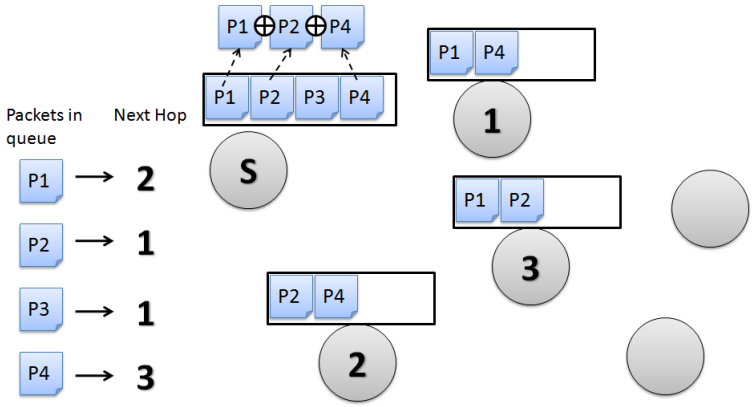

Figure 2.31: COPE coding

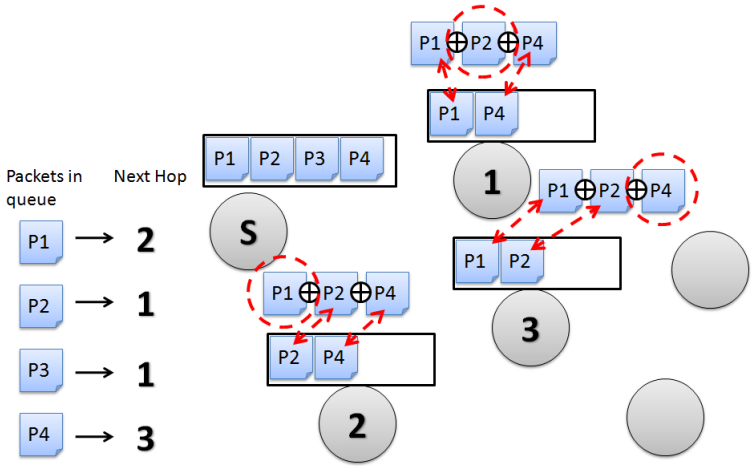

Figure 2.32: COPE decoding

Packet coding

COPE forwards packets without delay [30]. In other words, it never waits for packets to enable it to perform NC, but rather sends them as soon as the channel becomes available regardless of the amount of coding opportunities. The moment that the wireless channel becomes available, COPE will take the packet at the head of the queue, check for coding combinations, XOR the packets together and transmit them.

COPE never codes packets together that originate from the same source. The reason is attributed to the fact that, when n packets are XORed, n - 1 packets are required to

give the receiving node the ability to decode the coded packet. Therefore, when two packets are generated at the same node, it is impossible for any neighbour node to decode the XOR between those packets since the receiving node would not have either of the generated packets. Even if the receiving node had one of the generated packets, it would nullify the need to include that generated packet in the coded packet.

For a similar reason, COPE does not code packets together that is heading for the same next-hop. For obvious reasons, if two packets are heading for the same destination, it infers that neither of them are currently located at that destination rendering it impossible to decode the packet since at least one packet needed for decoding is lacking.

Therefore, when coding packets, only those that are not self-generated and are heading to different next-hops are considered. At each node, COPE maintains a virtual queue for each of its neighbours. It effectively harbours two virtual queues per neighbour, one for large packets and one for small packets. For explanatory purposes, we will assume only one queue per neighbour. This concept is depicted in figure 2.33. This scenario depicts source node *S*'s virtual queues for neighbours *1*, *2* and *3*.

When the wireless channel becomes available, source node *S* will take the packet at the head of its First In First Out (FIFO) queue as the primary packet, and consider packets at the head of the virtual queues to be possibly coded along with the primary packet. Only the combinations that deliver all the native packets to their intended next-hops are considered. The packets are coded according to the combination that delivers the most code gains.

Pseudo-broadcast

The 802.11 MAC defines two communication modes namely unicast and broadcast [30]. Coded packets in COPE need to be distributed to various nodes making the broadcast mode the expected choice. Alas, the broadcast mode lacks reliability. The broadcast mode does not support acknowledgements of received packets and does, therefore, not perform the exponential backoff procedure. The sender node, therefore,

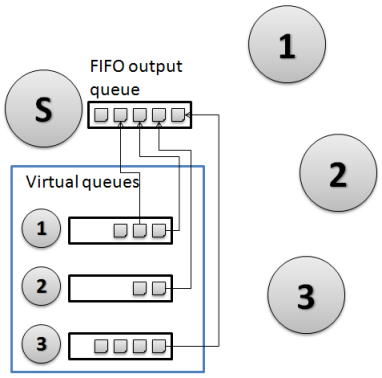

Figure 2.33: Virtual queues

has no guarantee whether the packet reached its destination or not. Hence in the case of a collision, the transmitted packet would never reach its destination and the sender node would not be aware of the lost packet.

COPE introduces a new communication mode, called pseudo-broadcast. The idea behind this method is to piggyback on the unicast mode. COPE assigns the MAC address of one of the intended next-hops as the destination for the coded packet. This action allows the MAC to send the packet in unicast mode, therefore, including the acknowledgement and backoff procedures that warrant reliability.

Coding and MAC gain

The experiments executed in [30] revealed throughput increase above the theoretically calculated code gain for the respective topology. The reason is attributed to the phenomenon that the interaction between the MAC and NC creates a synergy effect.

Please refer to figure 2.34. When hosts A and C exchange information, it travels through relay host B. The MAC is designed to divide and distribute the available bandwidth equally. The MAC, therefore, allocates equal bandwidth to all three hosts respectively without taking into account the fact that relay host B requires to transmit twice the amount of transmissions than required by hosts A and C. When NC is applied as depicted in the figure, it allows relay host B to drain packets twice as fast, effectively

doubling the throughput of this network.

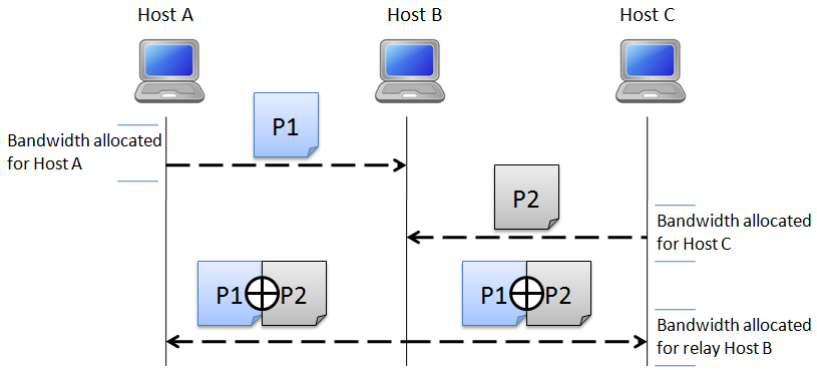

Figure 2.34: Coding and MAC gain

Packet decoding

Each node maintains a pool of packets containing all previously sent and received packets. When a node receives a coded packet, it cross-references the ID's of the XORed packets with the ones in the packet pool. If n packets is XORed, the node will be able to decode the coded packet as soon as n - 1 matches are found.

Limitations within COPE

Whilst COPE is opportunistic with regard to performing NC, it still lacks opportunistic features with regard to the paths the packets travel. Although the packets are coded opportunistically, the paths that they travel remain fixed. The fixed-path routing mechanism creates a limitation in the process of calculating coding opportunities since the calculating node may only code packets that can be decoded at the **fixed** next-hops; hence it may only include the specified next-hop nodes in the coding opportunity calculation.

This phenomenon allows the existence of scenarios where packets are sent along sub-optimal paths. CORE alleviates this shortcoming by incorporating OR into the scheme.

In the next section, we will conduct a review of OR methods. Finally, we will discuss how CORE incorporates COPE and OR.

2.9 Opportunistic Routing

In *fixed path* routing, fixed routes are established between all possible source and destination nodes. Therefore, based on the current position of a packet, it is assigned a specified next-hop node to where it needs to be routed. Packets travel these paths regardless of the presence of additional viable routes.

OR, unlike fixed path routing, does not specify a next-hop node for each packet [3]. It calculates a forwarder set. The forwarder set of a packet contains nodes that are closer to the packet's destination according to a specific routing metric. Upon packet reception, each node in the forwarder set calculates a prioritising **delay timer** based on *further criteria*. The term *further criteria* refers to conditions relevant to the objectives the routing protocol requires to accomplish. The objectives can include an increase in robustness, optimisation of throughput or to conserve energy. When the **delay timer** triggers, the node continues to forward the packet. The nodes with the better conditions will calculate smaller delays and will, therefore, transmit sooner than the other nodes in the forwarder set.

When a node forwards a packet, the remainder of the nodes in the forwarder set that overhears the packet will drop matching packets to suppress duplicate forwarding and unnecessary transmissions. OR is very effective in high packet-loss scenarios since the probability of an entire forwarder set failing as opposed to a single next-hop, is very slim. This phenomenon incorporates a high level of robustness into the network [35].

This opportunistic concept allows us to perform routing in an improvised fashion. Since routes are calculated at every intermediate node, dynamic topology changes are no longer a significant performance risk. Routes can now be chosen efficiently based on the immediate circumstances at nodes. We will now describe the basic functioning

of OR by means of an example, followed by a deeper look into various existing OR protocols.

Please refer to figure 2.35. In this scenario, source node *S* has a packet to send to destination node *D*. In traditional routing, only one node, lets assume in this case node *2* would be considered as the next-hop node in the process of forwarding the packet to its destination. In OR, node *S* firstly calculates a forwarder set according to a specified metric. As mentioned, nodes are elected into the forwarder set only if they are closer to the destination node according to a specified metric. In this instance, lets assume source node *S* calculates nodes *1*, *2* and *3* as its forwarder set. All the nodes in the forwarder set are now allowed to partake in the routing process. This means that nodes *1*, *2* and *3* will individually calculate priority timers based on their current circumstances. According to the calculation, the node with the best circumstances will calculate the smallest delay and, therefore, be the *primary forwarder*. The *primary forwarder* refers to the node that is first to transmit in the forwarder set.

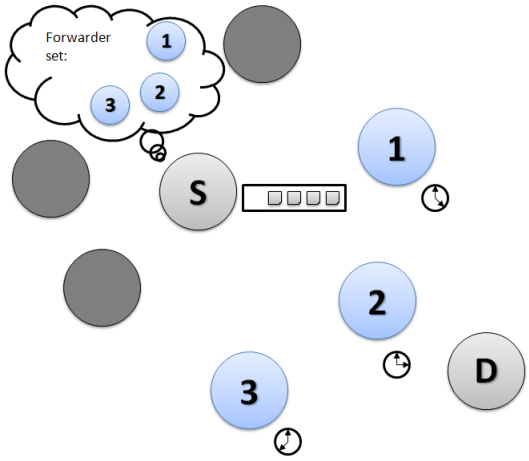

Figure 2.35: Forwarder set

In this scenario, we assume that node *2* is the primary forwarder. And, therefore, when nodes *1* and *3* receive the transmission from node *2*, they drop matching packets to prevent duplicate forwarding as depicted in figure 2.36

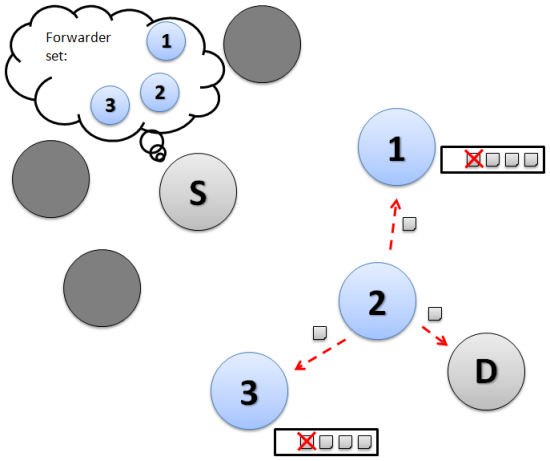

Figure 2.36: Forwarder set packet drop

2.9.1 ETX-Based Protocols

Extremely Opportunistic Routing (ExOR) was introduced in [3] by Biswas and Morris. ExOR consists of three phases: *forwarder set calculation*, *transmission acknowledgement* and *decision based forwarding*.

When a node has a packet to send, ExOR calculates a prioritised list of nodes that will bring the packet closer to the destination according to the number of hops to the destination. This is known as the *forwarder set calculation* phase. Forwarder set nodes are determined and prioritised as follows: ExOR identifies the shortest path to the destination and assigns the first next-hop node in that path top priority. It then removes the node from the equation and again calculates a new shortest path. The first next-hop node from the new shortest path is assigned next priority. This process continues until all the possible candidates are identified and prioritised.

When nodes in the forwarder set receive packets, they respond with acknowledgements. These acknowledgements are appended with the ID of the highest priority node currently known to them. This is the *transmission acknowledgement* phase.

When a node in the forwarder set receives acknowledgements from other nodes in the same forwarder set, it is enabled to determine whether it needs to forward the packet,

or discard it to suppress duplicate forwarding. Lower priority nodes will only send packets not acknowledged by higher priority nodes. This concludes the *decision based forwarding* phase.

2.9.2 Opportunistic Any-Path Forwarding

In ExOR, the Expected Transmission Count (ETX) metric used in calculating and prioritising the forwarder set is a single-path metric. Therefore, it does not take into account the fact that the forwarder set also performs OR, causing it to be an inefficient routing metric when implemented in OR. In [36], Zhong et al. proposed an opportunistic scheme, Opportunistic Any-path Forwarding (OAPF), in which they introduce a metric, Expected Any-path Count (EAX), specialised for OR. The EAX metric allows OAPF to select only forwarder candidates that really contribute to packet delivery. It accomplishes this task by recursively calculating optimal forwarder sets to reach the destination.

2.9.3 MAC-Independent Opportunistic Routing & Encoding Protocol

MAC-Independent Opportunistic Routing & Encoding Protocol (MORE) improves ExOR by incorporating RLNC into the routing process. A node that needs to transmit data firstly divides the packets into batches containing k packets. It then continuously transmits random linear combinations of packets from that batch. Forwarder nodes only store *innovative* packets. *Innovative* packets are packets linearly independent of the previously received packets form the same batch. When the destination node has received k linearly independent packets, it is able to retrieve the k original packets. After the k original packets is retrieved, the destination node sends an acknowledgement to the sender node. When the sender node receives the acknowledgement, it continues to send the next batch.

Linearly independent packets can be seen as follows:

$y_1 = x_1$

$y_2 = x_1 \oplus x_2$

$y_3 = x_2 \oplus x_3$

$y_4 = x_3$,

where x_i represents a **native** packet. The term **native** packet refers to the original packet created by the source node, thus as it was before it was coded. A coded packet would typically be represented by a packet such as y_2 or y_3 in this scenario.

Please refer to figure 2.37. Once destination node D has received sufficient linearly independent packets it is able to extract all the originally sent packets.

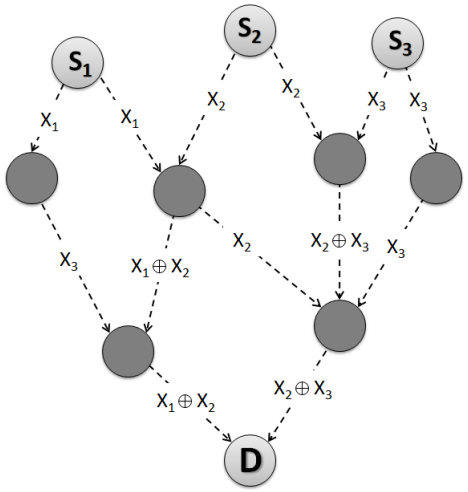

Figure 2.37: Random Linear Network Coding

2.9.4 Hop-Count-Based Protocols

The Opportunistic Routing in Dynamic Ad-Hoc Networks (OPRAH) protocol, presented in [37], is an on demand routing scheme for MANETs. During the request-reply route discovery cycle, OPRAH creates a braid multiple-path route between the source

and destination nodes. Every node in the braid broadcasts every data packet it receives. This technique causes many duplicate end-to-end transmissions. The concept of the OPRAH protocol is depicted in figure 2.38

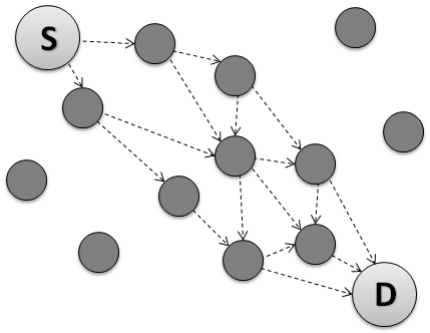

Figure 2.38: Opportunistic Routing in Dynamic Ad-Hoc Networks

In the Resilient Opportunistic Mesh Routing for Wireless Mesh Networks (ROMER) protocol, presented in [38], every packet carries a credit value recording the maximum distance it is allowed to deviate from the shortest path between the source and destination. The nodes on the shortest path forward packets with a probability equal to **one**. Nodes that do not reside on the shortest path have a forwarding probability based on the downstream throughput of the link. The concept of the ROMER protocol is depicted in figure 2.39

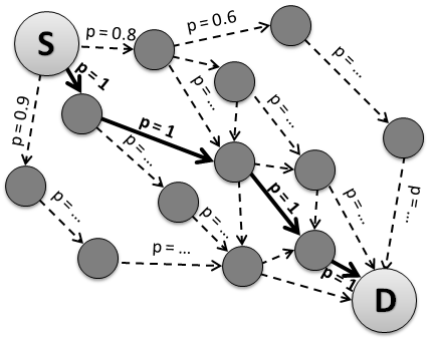

Figure 2.39: Resilient Opportunistic Mesh Routing for Wireless Mesh Networks

2.9.5 Geo-Distance-Based Protocols

A current existing routing protocol which incorporates GR into OR is known as GeRaF, presented in [5]. In this routing scheme, forwarder nodes are prioritised based on their geographical distance to the destination node. Nodes in the forwarder set with smaller distances to the destination node have higher priority.

2.10 CORE

CORE: "A coding-aware opportunistic routing mechanism for wireless mesh networks", presented in [34], is based on COPE and shares the same three key elements namely *opportunistic listening*, *opportunistic coding* and *learning neighbour state*. Instead of routing packets on fixed paths as in COPE, CORE enables nodes to opportunistically forward packets without specifying the precise next-hop node. CORE incorporates OR by means of first, as described in the section on OR, selecting a forwarder set of candidate nodes.

In this case, nodes have to be geographically closer to the destination node to be elected into the forwarder set. The criteria used for prioritising the forwarder set nodes are the number of local coding opportunities calculated at each forwarder node. The nodes with more coding opportunities will calculate smaller delays and, therefore, transmit packets sooner than nodes with less coding opportunities. Upon hearing forwarded packets, nodes in the forwarder set cross-reference the sent packets with packets in their own queue and drop them accordingly to prevent duplicate forwarding.

2.11 Conclusion

In this chapter, we reviewed WANETs. Some network characteristics and performance metrics were discussed. We took an elaborate tour through the layers of generic net-

work models. Discussions on routing methods including GR, NC, OR and energy aware routing were conducted. The incorporation of NC and OR was discussed in CORE, thereby concluding this chapter.

Chapter 3

Routing Scheme Design

In this chapter, we discuss how we propose to merge routing metrics such as NC, GR and energy awareness into a single superior routing protocol. We propose to do this by using one of the key phases that constitute OR. We will take an elaborate look into the development of the scheme.

3.1 Introduction

In this section, we develop a NC scheme. The scheme merges NC based on CORE [34], GR, and energy aware routing. We will firstly discuss the universal key components that constitute OR. The developed scheme will then be broken down into its core functioning elements to deliver simple understanding of how it works.

3.2 Basic functioning of scheme

The developed scheme is based on the basic operation of OR. In chapters 1 and 2 we discussed OR and its basic functioning components. The universal key elements of OR are *forwarder set selection* and *priority based forwarding*. Forwarder set nodes are elected, according to a specified metric, in the *forwarder set selection* phase and are then respectively prioritised in the *priority based forwarding* phase. The developed scheme is based on these universal components.

3.2.1 Forwarder set selection

If a node has a packet to send, it first selects a forwarder set. Nodes must comply with the following prerequisites to be included in the forwarder set: *1)* the node must be a direct neighbour node, *2)* it must be geographically closer to the destination node and *3)* it may not exist in the packet's previous forwarder set.

Direct neighbour node

In wireless networks, every node has a certain probability to overhear transmissions. A direct neighbour node refers to a node within high probability communication radius; hence a node that is in a position to communicate with high probability.

Geographically closer

The distance between nodes can be measured according to various metrics. Geo-distance refers to the physical distance between the nodes. Geographically closer denotes physically closer with regard to position.

Exclude previous forwarder set nodes

When nodes from the previous set are considered into the new forwarder set, it creates room for problem scenarios that degrade the performance of the protocol. A typical problem scenario of this kind is depicted in figure 3.1. Since geographical routing is included in the routing process, we can assume that the primary forwarder node will have the capability to forward the packet to a next wave of nodes. The concept *wave of nodes* refers to nodes with similar geographical distances with regard to the source and destination nodes as depicted in figure 3.2. We want to prevent packets from hopping between nodes in the same wave.

As depicted in figure 3.1, if a node is allowed to include nodes from the previous forwarder set into its newly calculated one, node *2* will be included in the calculated forwarder set of node *3* because it is geographically closer to the destination node.

Various conditions, including number of coding opportunities and or handshaking procedures, may lead to node *3*, from the same wave as node *2*, to forward the packet first. Since node *2* is part of the newly calculated forwarder set of node *3*, it will, alongside node *4*, partake in the next forwarding process. If node *2* succeeds to be the primary forwarder, it will cause packets to hop between similar wave nodes thereby decreasing network performance.

On the other hand, when nodes form previous forwarder sets are excluded from new forwarder sets, node *2* would be discarded from node *3*'s forwarder set, allowing the packet to gracefully advance from wave to wave and finally to reach its destination.

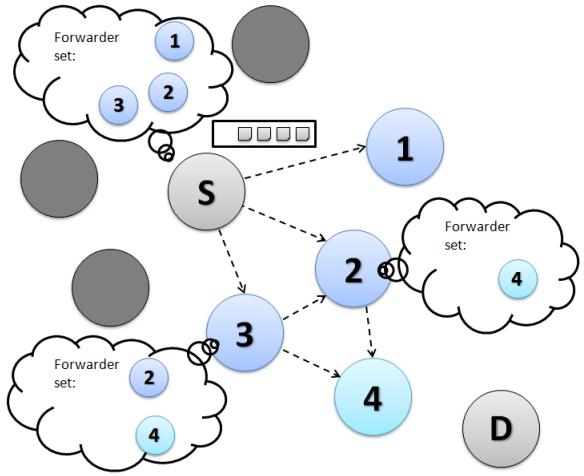

Figure 3.1: Forwarder set problem scenario

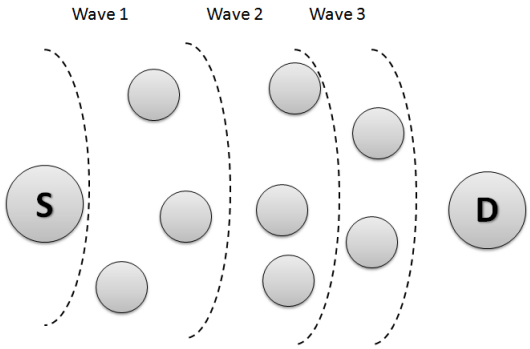

Figure 3.2: Node waves

Forwarder set example

Please refer to figure 3.3. In this scenario, source node *S* has a packet to send to destination node *D*. Nodes *1*, *2* and *3* are included in the forwarder set of node *S* because *1)* their geographical distances, *b*, are smaller than distance *a* between source node *S* and destination node *D*, *2)* they are direct neighbours of source node *S* and *3)* they were not included in source node *S*'s previous forwarder set.

As in OR, the forwarder set is a list of nodes with respective probabilities of being the primary forwarder of the packet. Nodes in the forwarder set are the only nodes considered when calculating delay timers.

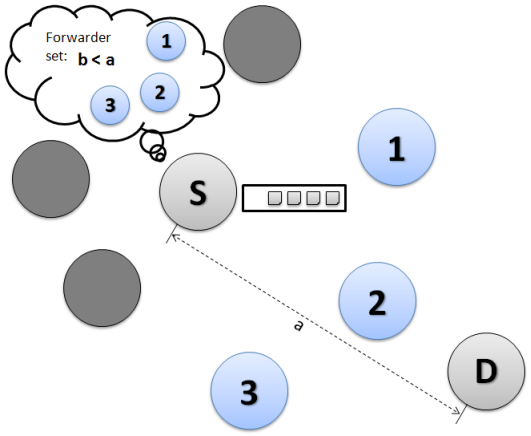

Figure 3.3: Forwarder set

3.2.2 Priority based forwarding

The *priority based forwarding* phase can be broken down into **two** phases namely *priority timer calculation* and *triggered forwarding*.

Triggered forwarding

The nodes in the forwarder set need to be prioritised to allow the primary node to forward the packet first. Each node uses the same algorithm to calculate a priority timer. The formula used to calculate the priority timer is formulated to include NC, GR and energy aware routing conditions. A node transmits only when the priority timer expires.

Priority timer calculation

The *priority timer calculation* phase is responsible for calculating the values of the **priority timers**. Every node, upon packet reception, calculates a time period after which the received packet will be sent. The calculation can include any number of influencing factors. In this scheme, we include NC, GR and energy aware conditions.

The priority timer calculation is illustrated in figure 3.4. The prioritising time interval t must be large enough to *1)* prevent collisions between forwarder set nodes and *2)* allow more packets to reach the node and possibly be included in a coded packet. It should also be small enough to prevent it from causing a delay of such magnitude that it causes packets to pile up and, in effect, decrease throughput.

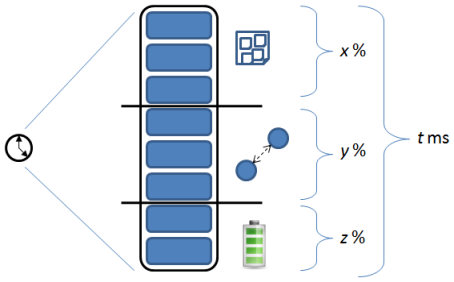

Figure 3.4: Priority timer

In figure 3.4, $x\%$ represents coding opportunities, $y\%$ represents geographical distance and $z\%$ represents the energy aware condition. Similarly, as the size of calculated time t is responsible for prioritising nodes, the size of the percentage of time t assigned to a condition prioritises the importance of that condition with regard to the other conditions. This figure only depicts the concept of condition prioritisation. Investigation regarding optimal condition percentage ratios will be executed in later chapters. Explanation of the three respective conditions follows.

Prioritising conditions

The three conditions that contribute to the calculation of the priority timer are discussed here. These conditions include NC, GR and energy awareness.

NC factor

This factor simply refers to the *code gain*. Here, *code gain* refers to the number of neighbour nodes that can decode the coded packet and retrieve a packet intended for it.

More coding opportunities denote smaller delays while fewer coding opportunities denote larger delays. We want nodes with more coding opportunities to be the primary forwarders since they increase throughput.

GR factor

The differences in the geographical distances between the forwarder set nodes and the destination node are used to calculate the magnitude of this factor. Further distances signify larger delays while smaller distances signify smaller delays. The node in the forwarder set closest to the destination achieves a delay of zero. Every other node with a greater distance is assigned a delay in accordance with its distance.

Energy aware factor

Clearly, nodes with higher energy resources are able to communicate longer if similar communication hardware, software and procedures are used. It would, therefore, simply put and generalised, be beneficial if nodes with lower energy resources leave communication to nodes with higher resources, in the case that they are indeed present. This concept will allow nodes with lower energy resources to spare their resources by avoiding unnecessary communication that do not necessitate their participation or do not serve them. Higher battery statuses, therefore, denote smaller delays while lower battery statuses signify larger delays.

3.3 Conclusion

In this chapter, we discussed the development of a routing scheme. The routing scheme incorporates NC, GR and energy awareness. We discussed how we propose to integrate and prioritise the respective factors. The following chapter discusses the implementation of the scheme developed here.

Chapter 4

Network Model Implementation

In this chapter, we provide a detailed discussion on the implementation of the developed routing scheme in OMNeT++. We will accomplish this task by taking a look at the technical side of the development process.

4.1 Introduction

The implementation of the scheme developed in chapter 3, *Routing Scheme Design*, is discussed in this chapter. We will describe the procedures constituting the scheme by means of flow diagrams and algorithms.

4.2 Network model

We will now discuss the implementation with regard to the OSI layers. For implementation purposes, we will work with four layers namely the *physical layer*, *MAC layer*, *Network layer* and *upper layers*. They are described below.

4.2.1 Physical layer

The physical layer is responsible for transmitting and receiving data over the medium. It handles the physical aspects of the communication process. Its functions are described in chapter 2, *Literature Study*.

4.2.2 MAC Layer

For the implementation of the developed routing protocol, we use the renowned 802.11b standard, also known as Wi-Fi. OMNeT++ MiXiM is equipped with an accurate implementation of the 802.11b standard [39]. Each node is equipped with a 802.11b MAC layer. As discussed in the literature study, chapter 2, the 802.11 standard defines the operation of the physical and data link layers. Some modifications were conducted on the MAC layer to make it more sophisticated for the purposes of the routing scheme.

It was modified to send all overheard packets to the network layer for processing seeing that the MAC layer cannot identify coded packets let alone perform decoding op-

erations. The layer was also modified to receive packet cancellation control messages from the network layer. The network layer sends cancellation messages to the MAC layer to indicate that the packet to be sent must be dropped. This feature is required for situations where the network layer realises it is in the process of sending a packet already transmitted by a neighbour node in the forwarder set. This event can only be realised at the network layer since it is responsible for decoding coded packets.

4.2.3 Network layer

The network layer performs all the coding and routing operations. It is, therefore, responsible for *1)* storing neighbour information, *2)* calculating forwarder sets, *3)* identifying coding opportunities, *4)* XORing packets; hence encoding and decoding, *5)* calculating priority timers and *6)* managing the output queue with regard to the priority timers.

Storing neighbour information

In CORE [34] and COPE [30], neighbour information are distributed in report packets. As mentioned before, report packets are appended to data packets or are sent periodically in control frames. The purpose of them is to inform nodes of the packets currently stored at their neighbour nodes. In this implementation, we assume that knowledge regarding stored packets and geographical coordinates are already available at the nodes.

Calculating forwarder sets

On packet reception, the network layer searches through its list of neighbours to identify viable forwarder set candidates. The neighbours with shorter geo-distances, than the current node, to the packet's destination are considered as viable forwarder set nodes.

Identifying coding opportunities

The network layer is responsible for identifying code gains. It runs through a list of possible combinations of how the packets can be coded. Every combination is cross-referenced with the packets stored at the neighbour nodes to determine where the combination will be decodable. Every neighbour node that can decode the combination is considered as a *code gain* when it exists in the respectively retrieved native packet's forwarder set.

Please see figure 4.1. This illustration is similar to the one used to demonstrate COPE in chapter 2. Source node *S* has packets *1*, *2*, *3* and *4* in its output queue. For each combination, in this case the combination that includes packets *1*, *2* and *4*, it checks what native packet will be retrieved by each neighbour that can indeed decode the combination. The retrievable native packets are circled in the figure.

The next step is to calculate the number of *code gains*. On the left, the figure also depicts the forwarder set nodes pertaining to each packet. The *code gain* calculation will entail the cross-referencing of those forwarder sets with the neighbours where the respective packets are decoded. Every match denotes a *code gain*. In this case, all the retrieved native packets have those neighbours in their forwarder sets, therefore, resulting in a *code gain* of **three**. It must be noted that the combination itself is only considered viable if all the packets in the combination respectively reaches at least one of the nodes in their forwarder sets. The valid combination with the most *code gains* wins, and the packets will be coded according to that combination.

Encoding and decoding

Encoding refers to the layer physically XORing the packets in the respective combination. Decoding refers to the cross-referencing of the packets contained in the coded packets with locally stored packets to determine whether the combination is decodable, and then XORing stored packets with the coded packet to retrieve a native packet

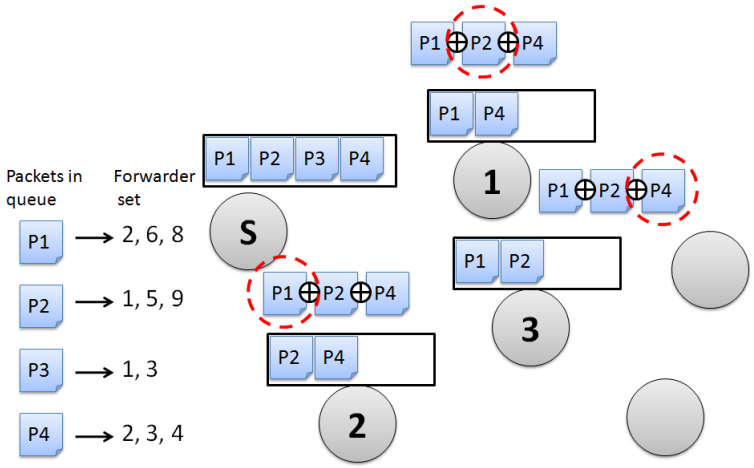

Figure 4.1: Forwarder set neighbour

should the combination be decodable here.

Calculating priority timers

Nodes in the forwarder set need to be prioritised by means of priority timers. The network layer has all the information required to calculate priority timers. The *coding factor* is determined by means of *code gains*. The *geo-distance factor* is calculated by comparing the current node's coordinates with the coordinates of the other nodes in the forwarder set. The *energy factor* is determined by using the battery status of the current node. These three factors are combined to determine the magnitude of the priority timer; hence the priority of the node.

Managing output queue

Finally, the network layer manages the output queue. Traffic generated at this node is not considered for coding. When new packets are added to the queue, new coding opportunities may arise and need to be managed. When priority timers expire, the correct packets must be taken from the queue and coded. All the aforementioned responsibilities are handled at the network layer.

4.2.4 Upper layers

All the layers above the *network layer* are merged into a single layer called the *upper layers*. Its purpose is to generate traffic.

4.3 OMNet++ implementation

The network model was build in OMNET++, specifically in the MiXiM environment. OMNeT++ is the non-commercial version of the well-known network simulator, OMNEST.

4.3.1 Implementation modules

Each node is equipped with the following communication layers starting at the lowest: A *physical layer*, *MAC layer*, *network layer* and *upper layers*. These layers are divided and integrated into modules. The modules include the *network*, *connectionManager*, *world* and *node* modules. They are described below, starting with the *network* module.

Network module

The network module, entitled *NewNetwork*, is depicted in figure 4.2. It includes three modules namely the *world*, *connectionManager* and *node* modules. Please pardon the uncouth layout of the *connectionManager* and *world* modules as OMNeT++ disables placement of those modules.

Connection Manager

The *connectionManager* module is responsible for determining connections between nodes based on factors such as distances, path loss and collisions. It allows the user

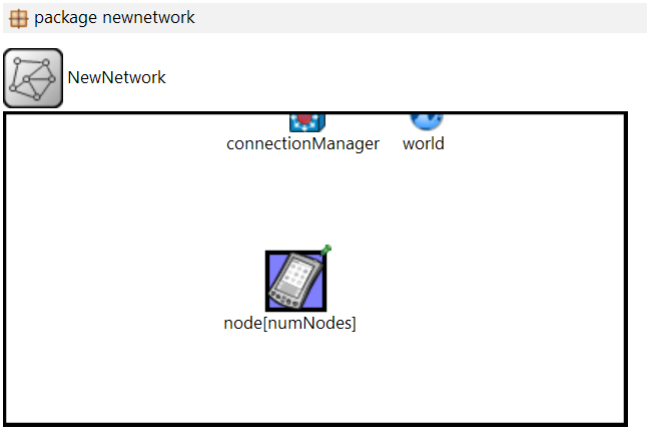

Figure 4.2: NewNetwork

to specify parameters such as the carrier frequency, transmission power, attenuation threshold and path loss coefficient alpha.

World module

The *world* module denotes the area of the simulation. It defines and manages the x, y and z coordinates of the area and the node placements.

Node module

The *node* module is depicted in figure 4.3. The module here is configured as the *Host80211* entity. The *appl* module represents the upper layers. The *netwl* layer represents the network layer module.

The *nic* module is depicted in figure 4.4. It is comprised of the *mac* and *phy* modules. Evidently the *mac* and *phy* modules respectively represent the MAC and physical layers.

The physical layer allows the user to specify parameters such as propagation delay and thermal noise. It also manages the reception of transmissions by taking into account

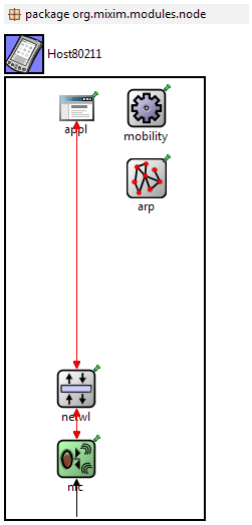

Figure 4.3: Node module

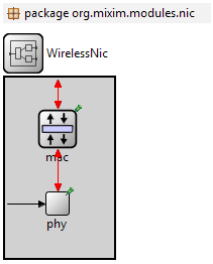

Figure 4.4: Wireless NIC

collisions. The MAC layer manages the communication bit rate.

4.3.2 Implementation layers

The respective layers and the modules they reside in were conferred in the previous section. The responsibilities and operation of the layers themselves are discussed here.

Upper layers

The upper layers denote all the layers above the network layer. Its sole purpose is to generate traffic. Since we only focus on the network layer and lower layers for the implementation and simulation purposes of the routing scheme, it would suffice to have a single layer responsible for generating the traffic.

Parameters of the generated traffic such as data flow, source and destination are configured at this layer. After the traffic is generated and configured, it is sent down to the network layer.

Network layer

The network layer receives traffic form the upper layers and from the MAC layer. Traffic from different layers undergo different procedures and are, therefore, handled differently. The respective processes are discussed below.

Traffic from upper layers

When a packet arrives form the upper layers, the forwarder set of that packet is calculated, after which the packet is added to the output queue. These packets are not considered for coding at this node due to reasons discussed in chapter 3, *Routing scheme design*.

Traffic from MAC layer

When packets arrive from the MAC layer, they continue through a rather more complex procedure. This procedure is depicted in figure 4.5. The first objective is to check whether the packet is a report or data type packet. Report packets carry information pertaining to neighbour nodes regarding stored packets and geographical coordinates.

If the packet is a report packet, the node continues to extract the information regarding its neighbours. If the packet is a data packet, the node must check whether it is a native or a coded packet.

In the case of a coded packet, the node must check in its packet pool whether it has the required packets to decode the coded packet. If it is not able to decode the packet, it will discard it. If it is, in fact, able to decode the coded packet and retrieve a native packet, it does so and continues with the procedure.

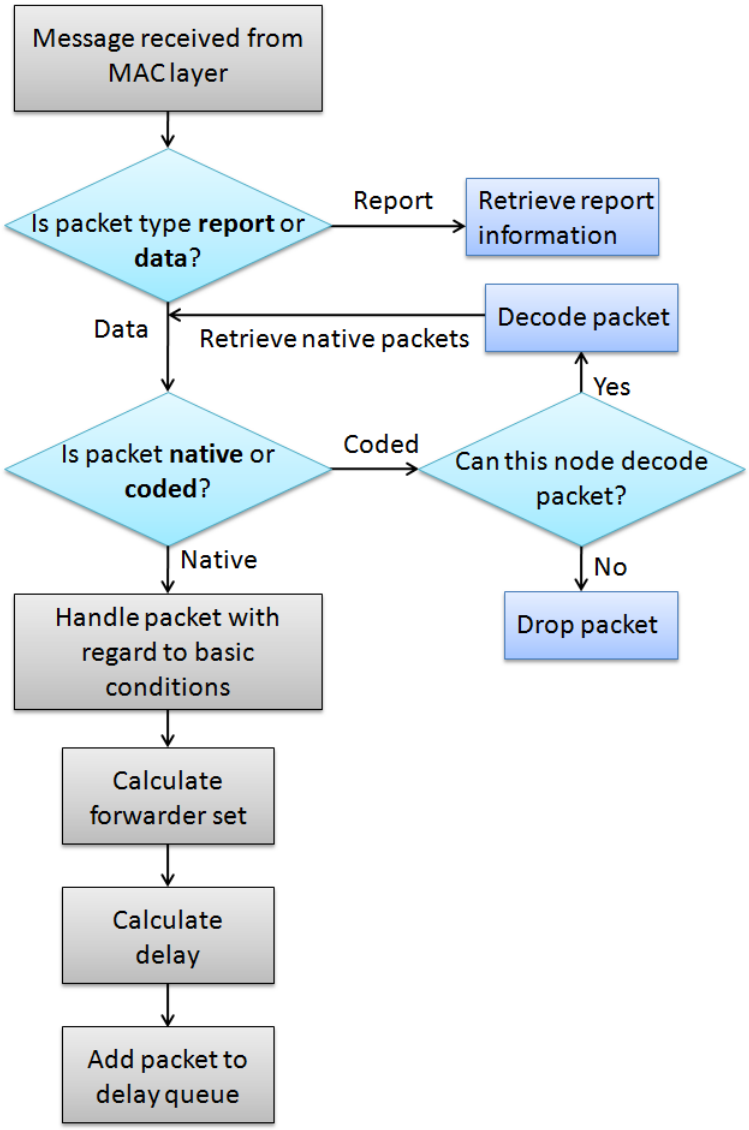

Figure 4.5: From MAC flow diagram

The next step is to handle the packet with regard to basic conditions. This procedure

is depicted in figure 4.6. The procedure simply checks whether this packet has been previously received by this node and whether this node is the final destination of the packet.

If this packet has already reached this node, it is dropped, otherwise its ID is added to the history pool. The history pool keeps track of packets previously received by this node. If the packet reached its destination, it is sent to the upper layers if this host is the destination node, otherwise it is dropped since it is of no further use to this node.

If this packet has not reached its destination, it implies that this node is possibly an intermediate forwarder. The current node now checks whether it is geographically closer to the destination than the sender node. If it is not, it will discard the packet. If it is closer, the packet is added to the NC pool, and the procedure continues. The NC pool stores packets that may be required at a later time to perform NC.

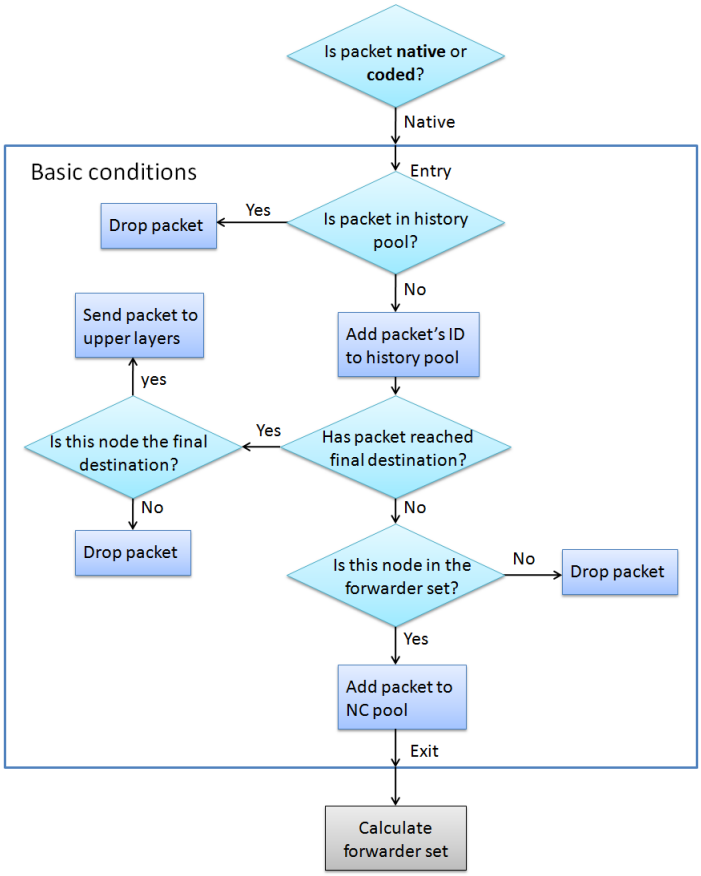

Figure 4.6: Basic conditions

The next step in the procedure is to calculate a new forwarder set. This procedure is depicted in figure 4.7. The node iterates through its list of neighbours. A neighbour is only elected into the forwarder set if it is *1)* geographically closer to the destination node and *2)* not present in the packet's previous forwarder set. If this node is not part of the forwarder set, the packet is dropped. If this node is, in fact, part of the new forwarder set, the procedure continues to prioritise the node by calculating a prioritising time delay for the respective packet.

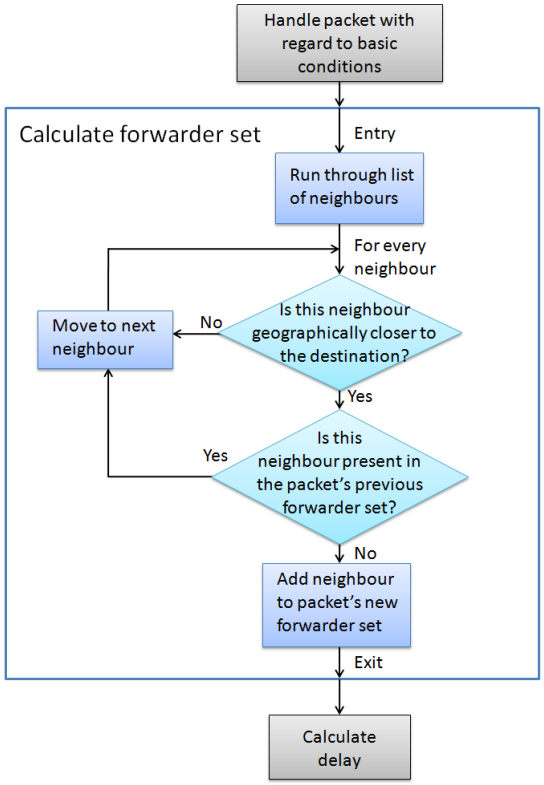

Figure 4.7: Calculate forwarder set

The time delay is the aggregate of three separate time periods namely *coding factor delay*, *distance factor delay* and *energy factor delay*. They respectively represent the delays obtained from the amount of coding opportunities, geographical distances and energy statuses. The prioritisation figure depicted in chapter 3 is included in figure 4.8 to aid in the discussion to follow.

The *coding factor delay* must be inversely proportional to number of calculated *code gains*. Reason being, the more coding opportunities we have, the higher priority this

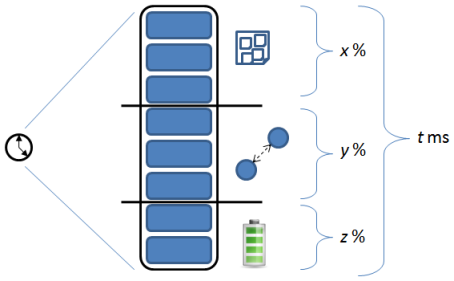

Figure 4.8: Prioritisation calculation

node must achieve, therefore denoting a smaller time delay. The *distance factor delay* must be directly proportional to the distance. As the magnitude of the distance between a forwarder set node and the destination increases, the node has a lower priority; hence denoting a larger delay. The *energy factor delay* is inversely proportional to the battery status of the node seeing that a lower battery status signifies a lower priority denoting a larger delay.

Based on the aforementioned conditions, prioritising delays are calculated as follows.

1) Coding factor delay:

$$Coding\ delay\ x\% = \frac{x}{CG}\,, \tag{4.1}$$

where x is the assigned coding factor prioritising period and CG is the amount of code gains. From this equation, it is evident that more code gains cause smaller delays while fewer code gains cause larger delays.

2) Distance factor delay:

$$Distance\ delay\ y\% = (y) \times (d - d_{min})\,, \tag{4.2}$$

where y is the assigned distance factor prioritising period, d is the distance between the current node and the destination node, and d_{min} is the distance between the destination and the node in the forwarder set closest to the destination. From this equation, it is

evident that smaller destination-distances cause smaller delays while larger distances cause larger delays.

3) Energy factor delay:

$$Energy\ delay\ z\% = (z) \times (1 - \frac{b}{b_{max}}) , \tag{4.3}$$

where z is the assigned energy factor prioritising period, b is the battery status of the current node, and b_{max} is the maximum achievable battery status. From this equation, it is evident that weaker battery statuses cause larger delays while better battery statuses cause smaller delays.

Different magnitudes of time periods x, y and z will be allocated to the respective factors and the effects thereof will be analysed in chapter 6, *Simulation and Analysis*. For now, only note that longer time periods denote more weight, denoting higher priority.

MAC layer

The MAC layer handles the access to the medium. It is responsible for the 802.11b standard procedures such as handshaking, specifically RTS, CTS and ACK control messages. More information pertaining to this layer has already been discussed in previous sections.

Physical layer

The physical layer is responsible for communicating data over the communication medium with regard to the physical aspects. More information pertaining to this layer has already been discussed in previous sections.

4.3.3 Model assumptions

For simplification purposes, the following assumptions regarding the model were made:

- All the nodes have equal chance to be a source or sink node, given that there exists a geographical route between them.
- All the nodes know their geographical coordinates by means of GPS or localisation techniques.
- The data distributed through report packets are already present at the nodes. This includes geographical coordinates and stored packets.

4.4 Conclusion

In this chapter, we discussed the implementation of the scheme developed in chapter 3. We provided explanations regarding the modules and layers utilised for implementation purposes. The next chapter focusses on validating and verifying the model.

Chapter 5

Validation and Verification

In this chapter, we verify and validate the model implemented in the previous chapter. We establish scenarios in which we respectively test and observe various aspects of the implementation. We provide simulation exhibits from OMNeT++ and discuss the results obtained from the experiments.

5.1 Introduction

Verification and validation is a crucial part of the development process. Verification and validation processes were discussed in the *introduction* chapter. To refresh your memory, verification refers to checking whether the model is correctly built or created. Validation refers to ensuring that the correct model is built.

5.2 Methodology

In this section, we aim to give the methodology of how validation and verification will be applied. We will take a look at methods used to perform validation and verification as well as underline those methods that effectively apply to our research. In chapter 1, *Introduction*, we took an overview of validation and verification by means of a simplified model. In the following sections, we will explore specific techniques that we can utilise in the process of verifying and validating our model.

In [1], Sargent introduces a term "model confidence" which refers to the validity of the model. The author states that it can be excessively costly and time consuming to completely validate the model over the entire spectrum of its possible applications. This concept is depicted in figure 5.1. Continuous validation will inevitably reach a point where the cost of validating the model outweighs the value of the model itself. Rather than completely validating the model, we aim to validate the model sufficiently pertaining to its intended purpose and application.

5.2.1 Verification

The model is partially verified in the previous chapter with regard to the discussion on the implementation procedures. Here, we look at specific scenarios and observe how the model performs in different circumstances. The scenarios are chosen in a manner

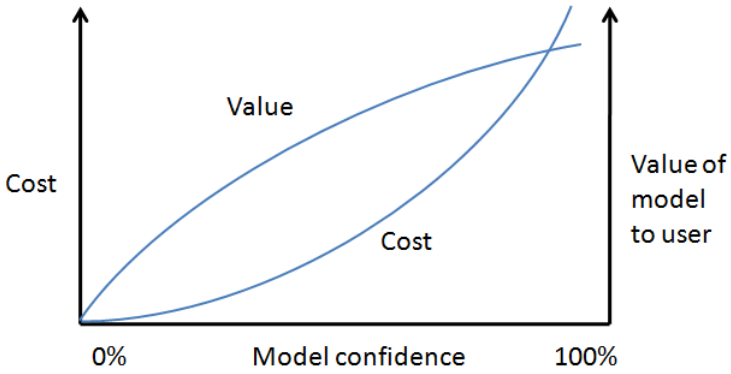

Figure 5.1: Model confidence

that reveal the operation of the sub modules exclusively. This will give us a clear view as to whether the underlying code is implemented correctly.

In every scenario, the actions of the nodes can be predicted beforehand since every node is equipped with the model designed in chapters 3 and 4. We, therefore, state what we expect from every simulation prior to the simulation and afterwards check whether the simulation results corresponds with the expectations. By means of animations and an output console we can track every event that occurs. We will use this information to verify the model [40].

5.2.2 Validation

The simulated scenarios mentioned in the previous section also significantly contributes as validation [1].

A model can be validated by comparing it to other validated models regarding network characteristics or performance [1]. The coding factor of the developed routing protocol is based on the NC scheme CORE, presented in [34]. We can, therefore, compare our results with those acquired in [34]. Similar results would reinforce the validity of our implementation.

5.3 Simulated scenarios

In this section, scenarios are created and simulated in OMNeT++. Simulation diagrams are constructed from the simulation results and they depict every significant and relevant simulation event. The purpose of the carefully designed scenarios is to provide verification and validation regarding the implemented model.

5.3.1 Hidden node scenario

As discussed in chapter 2, *Literature Study*, the hidden node problem is dealt with by implementing handshaking procedures into the CSMA/CA process. Since the 802.11b MAC layer is implemented in the nodes in the simulations, they should perform handshaking procedures accordingly. The handshaking procedures would support collision avoidance and allow the hidden node to communicate successfully with its neighbour nodes.

A hidden node scenario is created in OMNeT++ as depicted in figure 5.2. In this simulation, *node[0]* and *node[2]* are configured to transmit packets to *node[1]*.

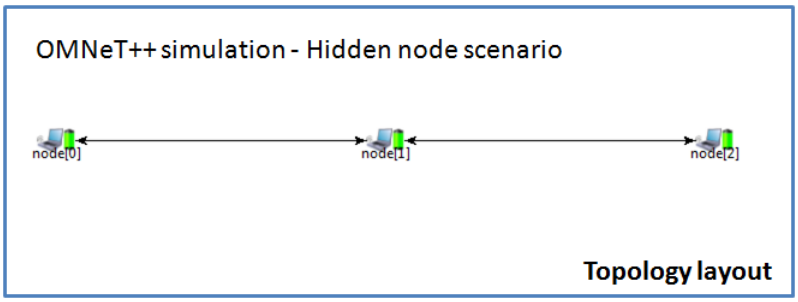

Figure 5.2: Hidden node topology

The simulation results are shown in figure 5.3. The RTS, CTS and ACK messages are used for all communicated traffic; hence handshaking messages are present throughout communicated traffic. The simulation shows that *node[1]* successfully receives packets from its neighbour nodes. For simplification and explanation purposes, these

handshaking messages will be excluded from the remainder of the simulation output diagrams. This simulation output diagram includes them merely to provide a detailed picture as to how the simulations are observed.

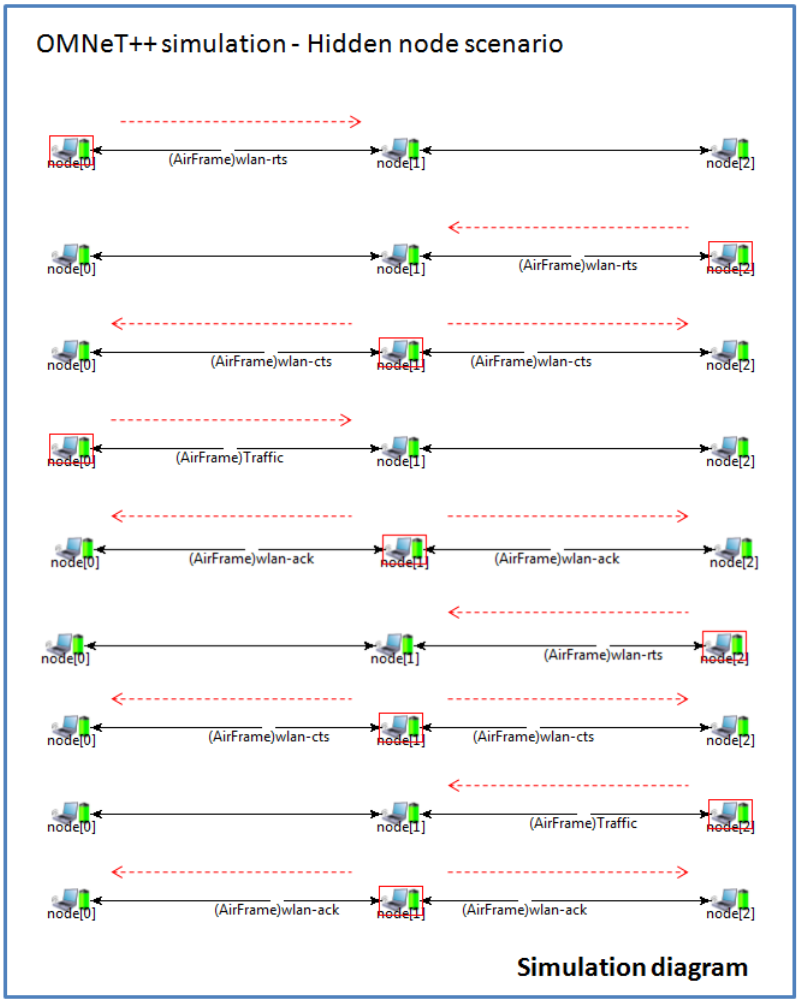

Figure 5.3: Hidden node simulation

5.3.2 Coding factor

The coding factor refers to the NC portion of the developed scheme. In this section, we set up communication scenarios where coding opportunities exist. We simulate the scenarios and evaluate the actions of the nodes by means of operation and visual outputs. OMNeT++ animates all the inter- and intra-interactions of the nodes which

allows users to observe all the communication processes visually.

Scenario 1 - Two hop coding

In the first scenario, we evaluate coding performed over two hops. The scenario created in OMNeT++ is depicted in figure 5.4. In this simulation *node[0]* and *node[2]* exchange packets.

According to the operation of the scheme, *node[1]* must wait a default delay after packet reception. The delay is configured large enough to allow packets from *node[0]* and *node[2]* to reach *node[1]* before it transmits. This will create a situation where both packets from *node[0]* and *node[2]* are present at *node[1]*, in which *node[1]* should be able to perform NC.

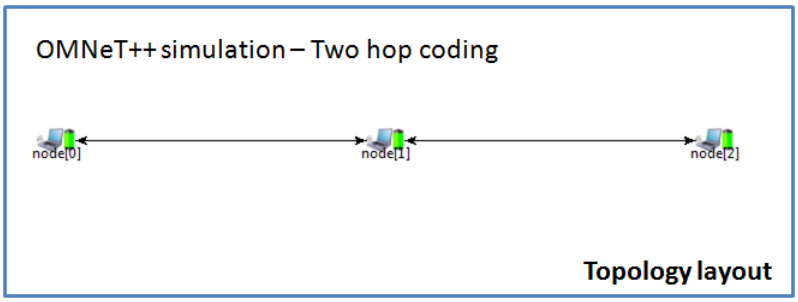

Figure 5.4: Two hop coding topology

The simulation results are depicted in figure 5.5. Traffic from both neighbours *node[0]* and *node[2]* are received and coded at *node[1]*, which responds by transmitting a coded packet to both neighbours. Both *node[0]* and *node[2]* are able to decode the packet, retrieve the native packet intended for it, and send it to the upper layers. This process is depicted in figure 5.6. Similarly to the handshaking processes, these respective decoding processes inside the nodes are only shown for the sake of being thorough. Simulation depictions from here on will exclude such intrinsic details to avoid unnecessary over-extensiveness. It must be noted that these processes still takes place regardless of their presence or absence in the depictions.

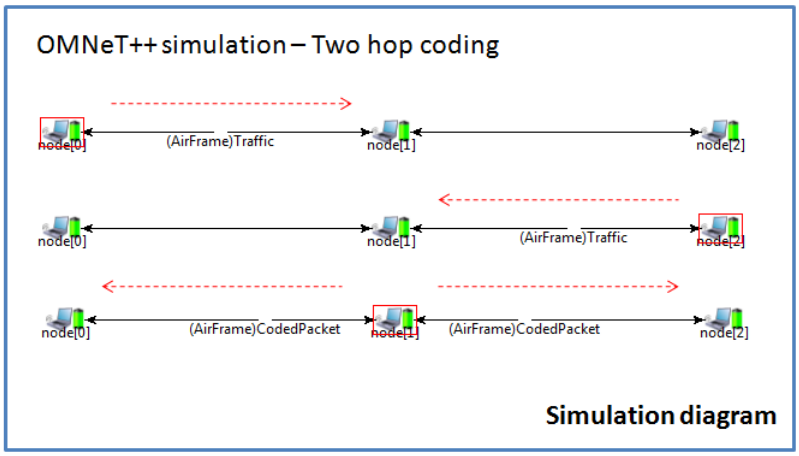

Figure 5.5: Two hop coding simulation

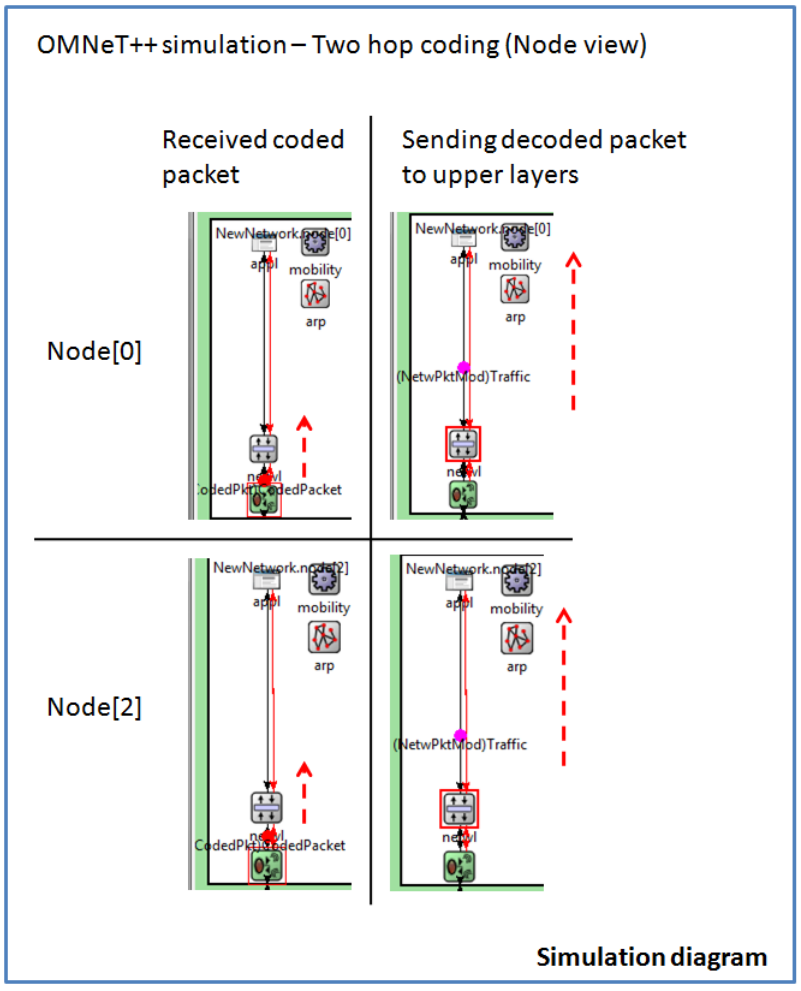

Figure 5.6: Decoding

The purpose of this scenario was to test whether the packets are coded and decoded correctly. The simulation results indicated the correct implementation of those procedures.

Scenario 2 - Forwarder set prioritisation

The purpose of the following simulation is to check whether the forwarder set is correctly prioritised. The scenario is depicted in figure 5.7. In this simulation, **1)** *node[0]* sends traffic to *node[1]* and **2)** *node[4]* sends traffic to *node[0]*. It must be noted that, for the sake of testing purposes, only the coding factor is included in the prioritisation process in this simulation. The exclusion of the geo-distance and energy factors allow us to test the coding factor discretely.

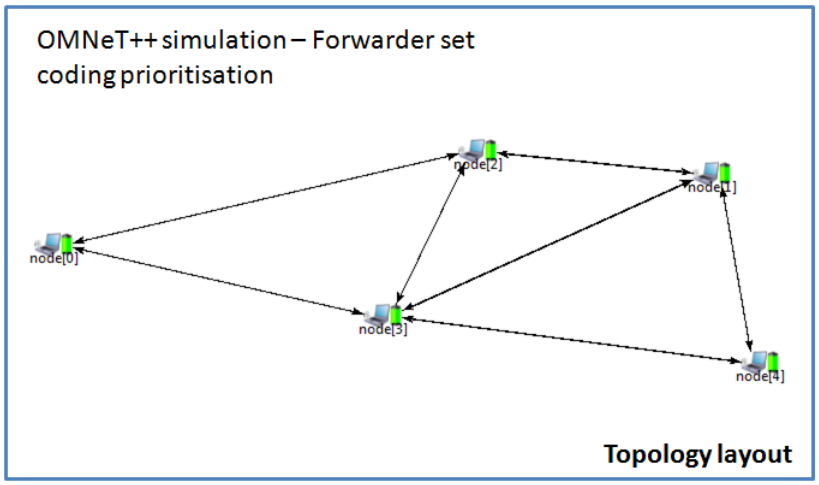

Figure 5.7: Forwarder set coding prioritisation topology

The minimum hop traffic routes for aforementioned communication conditions are depicted in figure 5.8. In the event of traditional routing, traffic generated at *node[0]* will travel along routes *1* and or *2* based on the routing metric applied. If implemented correctly, the developed scheme should give priority to coded routes. Since routes *2* and *3* passes through *node[3]*, a bidirectional flow of information occurs at the circled node causing coding opportunities.

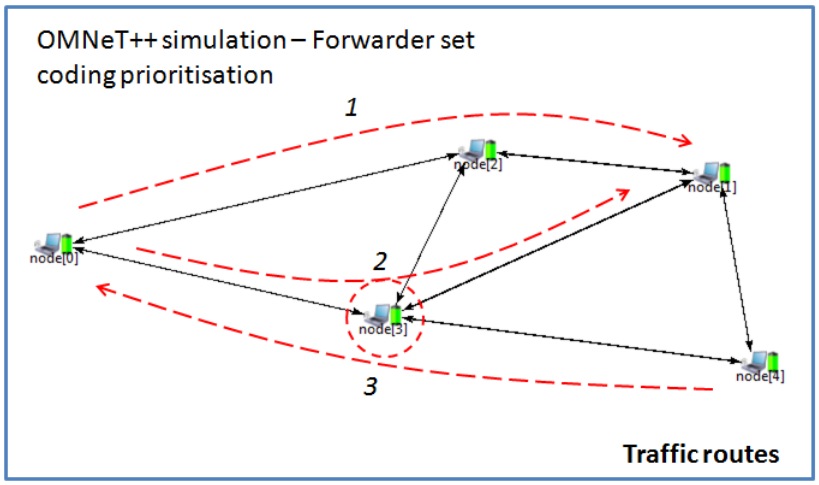

Figure 5.8: Traffic routes

The forwarder sets are depicted in figure 5.9. According to the routing protocol, *node[2]* and *node[3]* comprise the forwarder set of *node[0]* since they are geographically closer to destination *node[1]*. In the same sense, *node[1]* and *node[3]* comprise the forwarder set of *node[4]* as they are geographically closer to destination *node[0]*.

Since *node[3]* has the most code gains in both forwarder sets, it should attain higher priority than the other forwarder set nodes and, therefore, be the first and only node in the forwarder set to transmit data traffic. After *node[3]* transmits and *node[1]* and *node[2]* receive the coded packet, they will drop matching traffic and, therefore, never forward data traffic. It must be noted that although *node[1]* will not forward data traffic in this scenario, it will nevertheless continue transmitting data it generates.

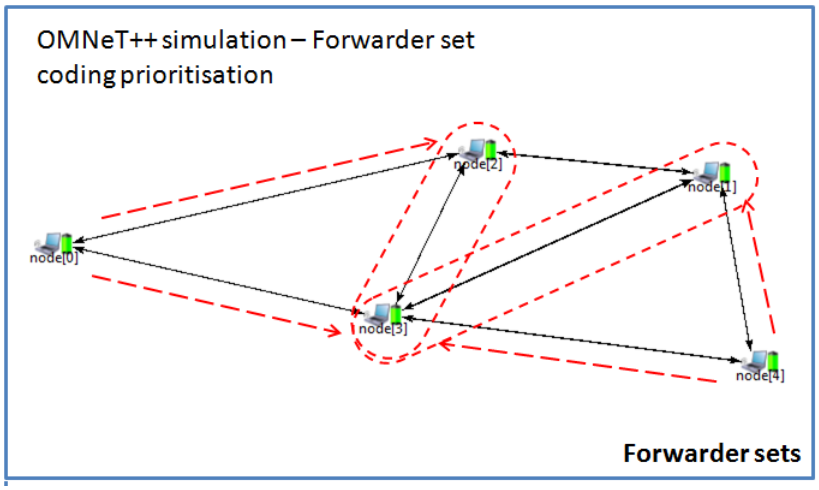

Figure 5.9: Forwarder sets

Since *node[0]* and *node[4]* commence traffic generation simultaneously, we expect all the traffic to arrive at intermediate *node[3]* in a small enough difference in time of arrival in order for a time window to exist where traffic from both nodes are present at *node[3]*. At this time period, they can be considered for coding.

The simulation output is depicted in figure 5.10. The simulation verified the correct operation of the routing protocol. The traffic reached forwarder set nodes where it was prioritised. Since *node[3]* had the most code gains, it calculated the smallest delay compared to that of other nodes in the forwarder sets and, therefore, transmitted first. The other nodes in the forwarder sets dropped their matching packets on reception of the coded packet from *node[3]* and, therefore, did not forward any data traffic.

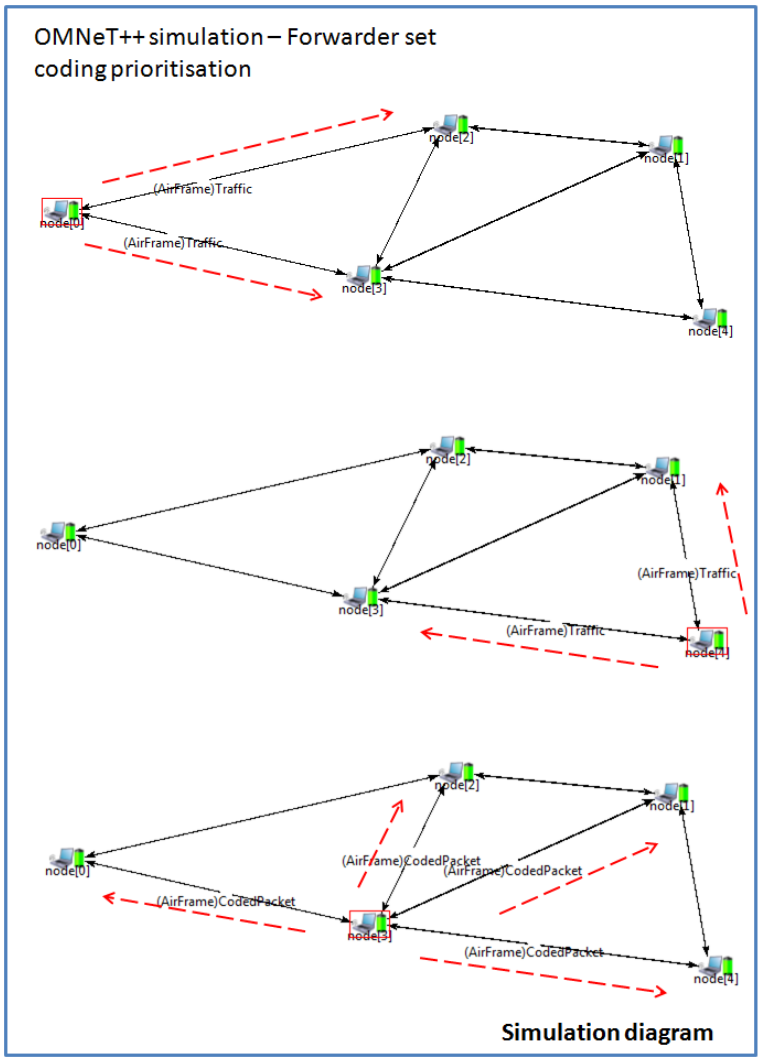

Figure 5.10: Forwarder set coding prioritisation simulation

The purpose of this simulation was to examine whether the time delay calculation phase correctly prioritises nodes in the forwarder set according to coding opportunities. The simulation confirmed the correct operation of the implemented routing scheme.

5.3.3 Geo-distance factor

The geo-distance factor refers to the geographical prioritising condition. It must be noted that, for the sake of testing purposes, only the geo-distance factor is included in the prioritisation process in this simulation. The exclusion of the coding and energy factors allow us to test the geo-distance factor individually. The simulation layout is depicted in figure 5.11.

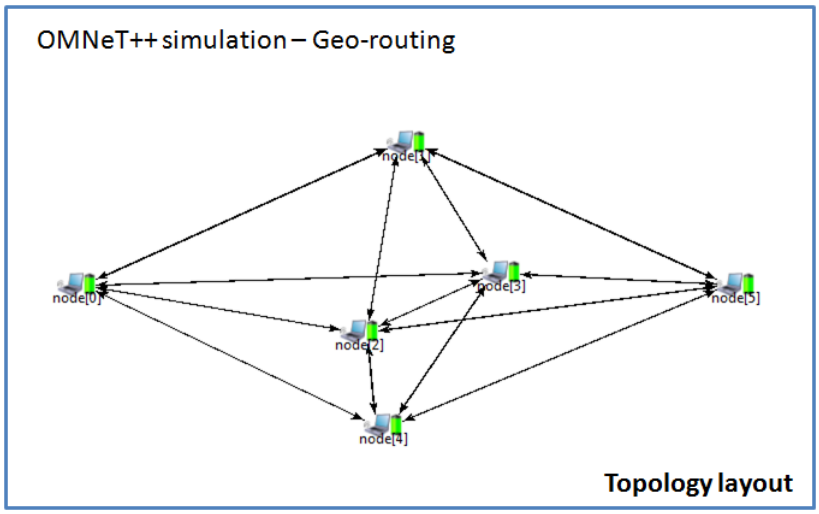

Figure 5.11: Geo-distance factor topology

In this simulation, *node[0]* is configured to send data traffic to *node[5]*. There are **four** nodes in the forwarder set as depicted in figure 5.12. The nodes are elected into the forwarder set because their geographical distances to destination *node[5]* is smaller than that of source *node[0]*.

The nodes in the forwarder set are then prioritised according to their geographical distance to the destination nodes. We, therefore, expect *node[3]*, since it is closer to destination *node[5]* than all the other nodes in the forwarder set, to calculate the smallest prioritising delay and therefore transmit first. When the other nodes in the forwarder set overhears the transmission from *node[3]*, they drop matching packets preventing duplicate forwarding in this scenario.

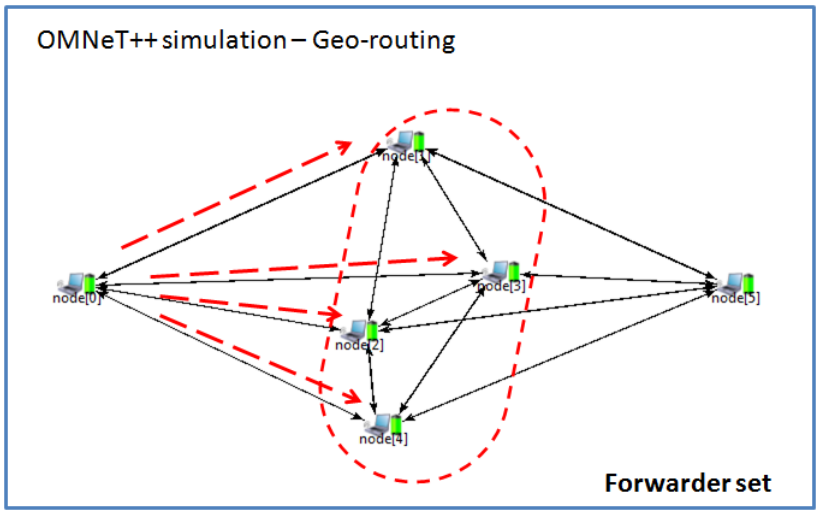

Figure 5.12: Geo-distance forwarder set

The simulation results indicate that the implemented model operates correctly in accord with the developed scheme. The simulation is depicted in figure 5.13. The entire forwarder set receives traffic from *node[0]*. They then prioritise themselves by calculation of priority delays. As expected, in every case *node[3]* calculates the smallest delay and becomes the primary sender.

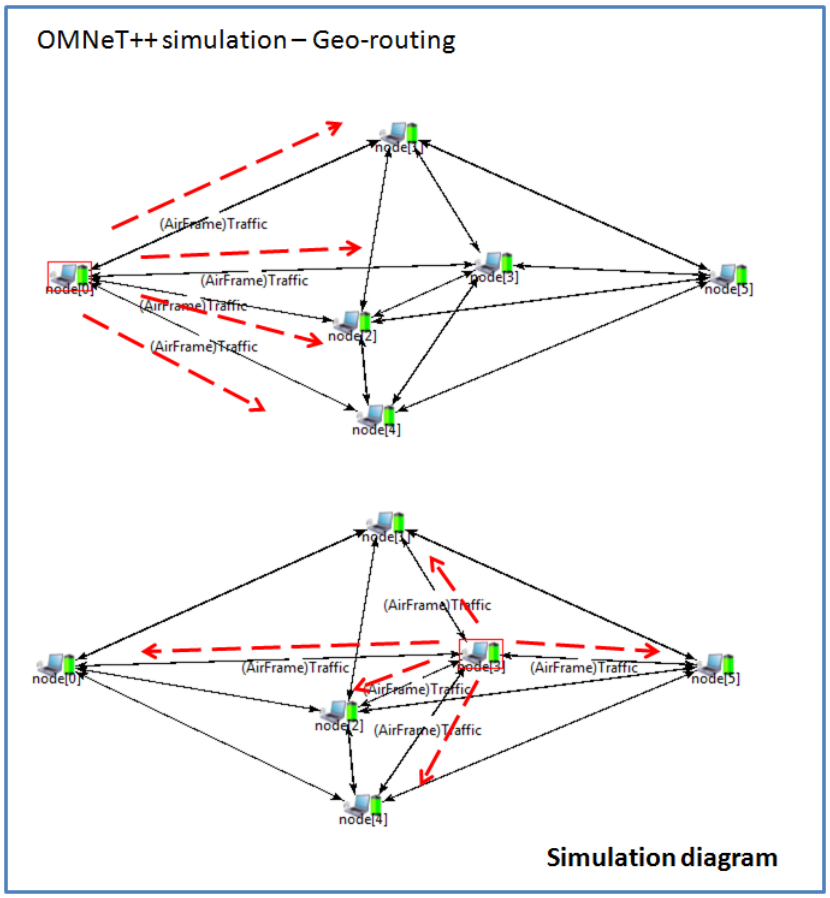

Figure 5.13: Geo-distance factor simulation

The purpose of this simulation was to verify that the computerised model is correctly implemented with regard to the geo-distance factor prioritising condition. The simulation results confirm that the implementation is correct.

5.3.4 Energy factor

The energy factor refers to the energy prioritising condition. The following scenario depicts a situation where the energy aware component of the routing can be observed exclusively. The scenario created in OMNeT++ is depicted in figure 5.14. The simulation is configured that *node[0]* continuously send traffic to *node[4]*. The energy factor is the only condition included in the prioritising calculation here. All the nodes are

equipped with full battery status. The source node is provided with unlimited energy resources in order for it to be able to send traffic continuously.

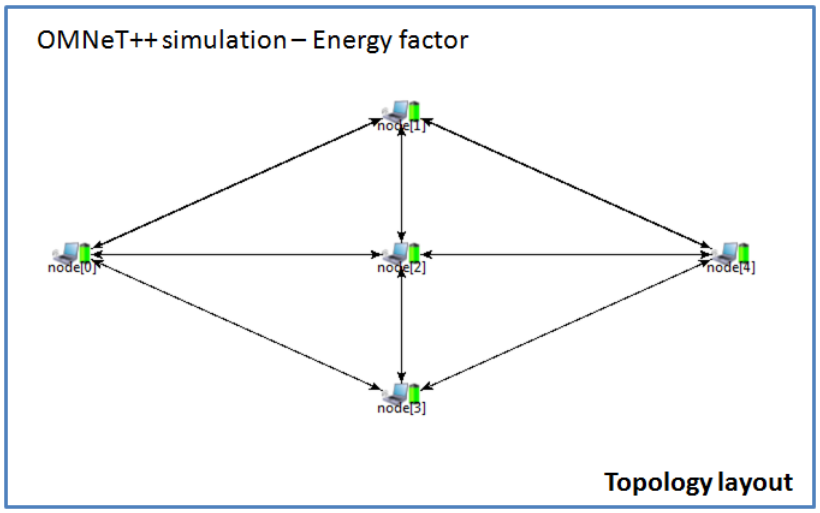

Figure 5.14: Energy factor topology

The forwarder set of *node[0]* is depicted in figure 5.15. Nodes *node[1]*, *node[2]* and *node[3]* are included in the forwarder set since they are closer to the destination *node[4]* than source *node[0]*. We initially expect any one of the forwarder set nodes to forward the traffic since they all have equal battery levels. From the second generated traffic packet we expect only the nodes with the highest energy levels compared to other nodes in the forwarder set to forward traffic.

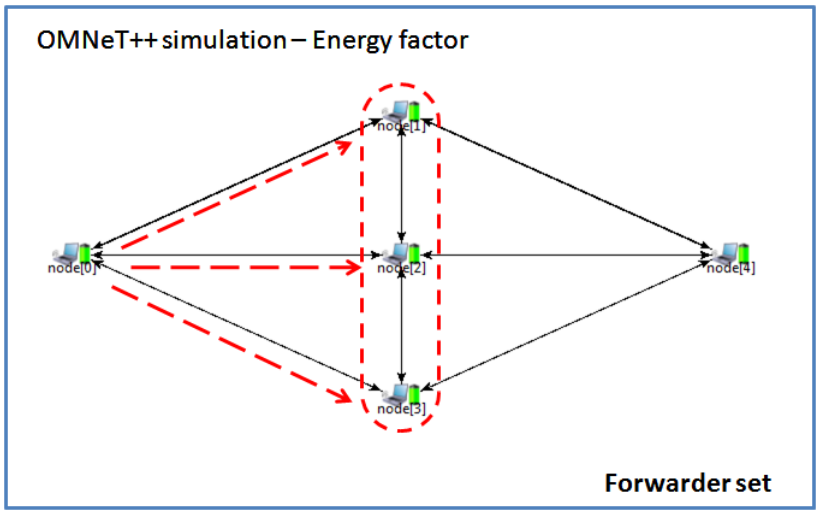

Figure 5.15: Energy factor forwarder set

The first simulation output diagram is depicted in figure 5.16. Source *node[0]* transmits a packet to the forwarder set. Any one of the nodes in the forwarder set can forward the packet, in this case *node[1]* forwards the traffic. The rest of the forwarder set nodes drop matching packets to suppress duplicate forwarding. The energy level of *node[1]* drops with the transmission as indicated in the diagram.

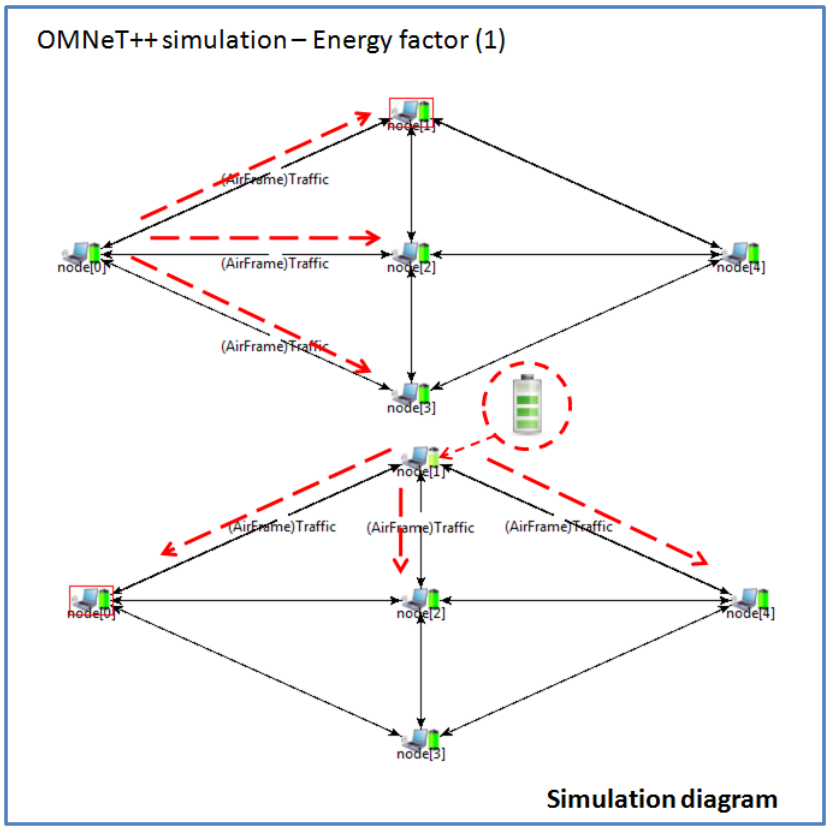

Figure 5.16: Energy factor simulation(1)

Source *node[0]* now transmits a second traffic packet as depicted in figure 5.17. Either one of nodes *node[2]* or *node[3]* should now forward the packet since their energy levels are higher than that of *node[1]*. As shown in the diagram, *node[2]* continues to forward the packet. Other nodes in the forwarder set drop matching packets to prevent duplicate forwarding. The energy level of *node[2]* drops as shown in the diagram.

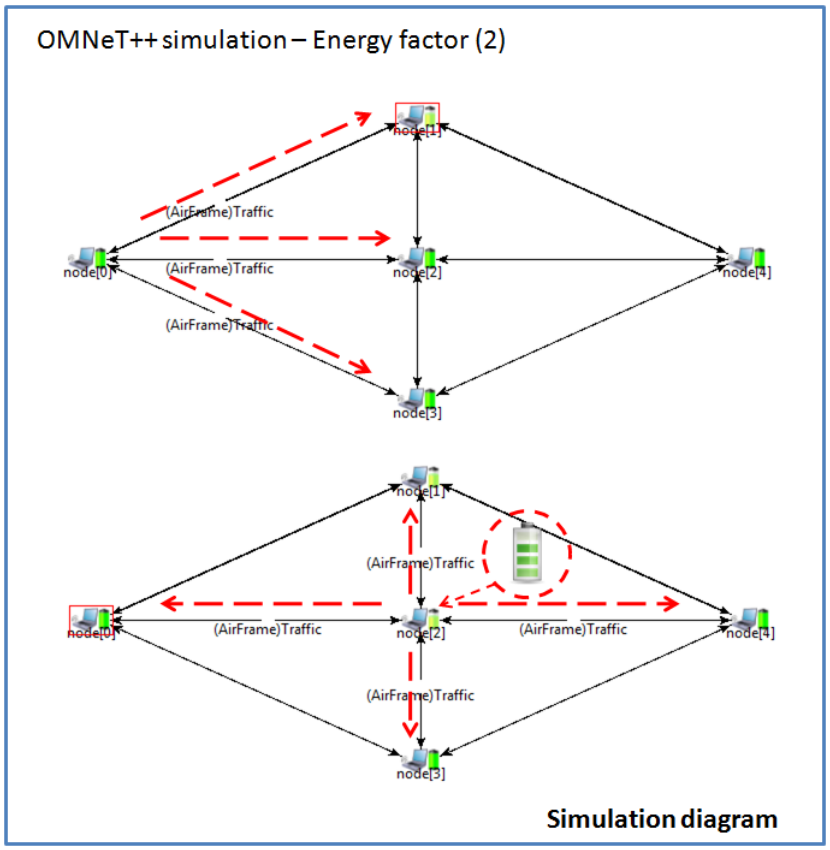

Figure 5.17: Energy factor simulation(2)

The last simulation diagram depicts source *node[0]* sending a third and final traffic packet as depicted in figure 5.18. Since *node[3]* now has the highest energy level compared to other nodes in the forwarder set it should be the primary forwarder. As expected, *node[3]* calculates the smallest delay and continues to send the packet. The energy level of *node[3]* drops as depicted in the diagram.

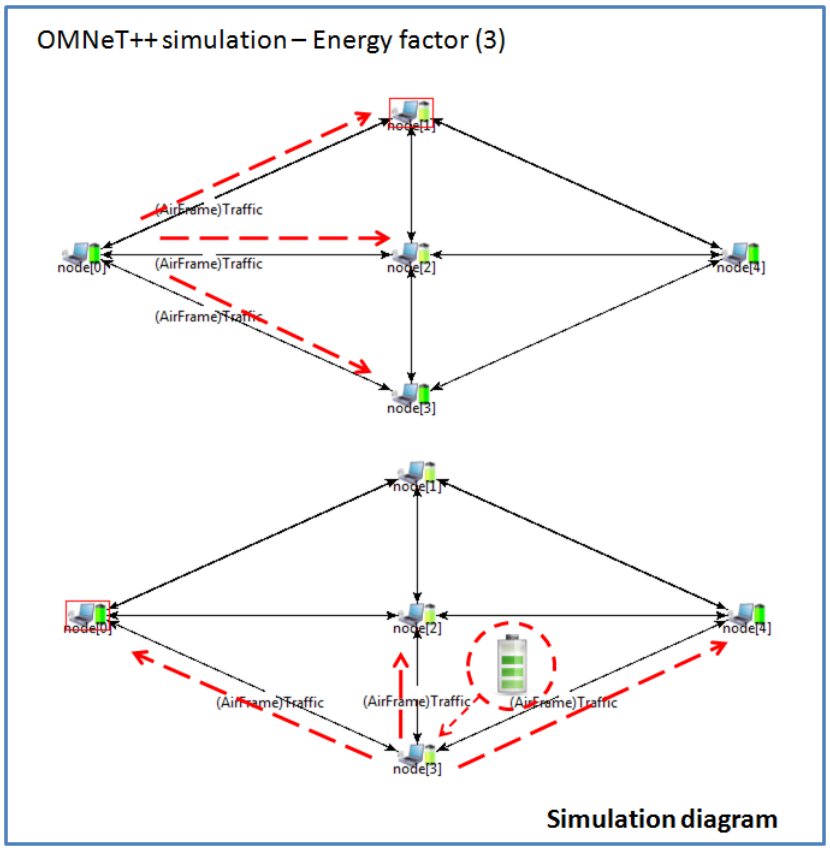

Figure 5.18: Energy factor simulation(3)

The purpose of this simulation was to test the implementation of the energy factor in the prioritising calculation phase. The simulation results verify that the implementation is correct.

5.4 Model comparison

In this section, we aim to validate the implemented network model. Since the coding factor of the developed model is based on CORE, our routing protocol should, in the absence of the geographical and energy conditions, perform exactly the same as CORE. It should, therefore, deliver similar results to that obtained by CORE in [34].

The implementation of CORE, as described in chapter 4, was simulated, and the throughput was measured for different loads. As depicted in figure 5.19, the simulation delivers similar results to that obtained from the original implementation in [34]. The differences in throughput are attributed to the fact that different hardware and software were used for the respective implementations and simulations.

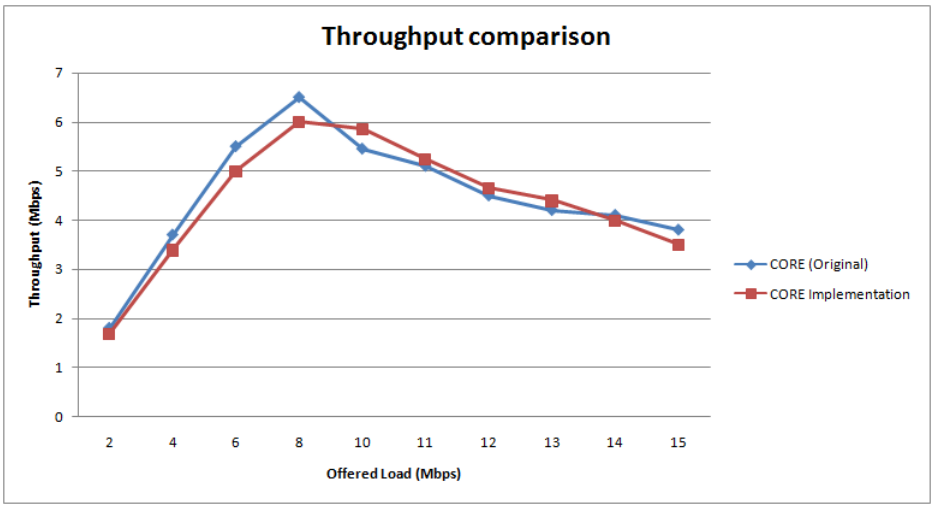

Figure 5.19: Throughput comparison

5.5 Conclusion

In this chapter, validation and verification techniques were performed to confirm the validity of the implemented model. The model was validated and verified by means of various scenarios.

Chapter 6

Simulation and Analysis

In this chapter, we perform various simulations on the implemented model. The simulations reveal information that aid in the optimisation of the model. The simulation results indicate significant increase in network performance with regard to throughput and survivability.

6.1 Introduction

Simulation parameters and results are discussed in this chapter.

6.2 Visual Aspect of the Simulation

A typical simulated network is depicted in figure 6.1. Each node displays its battery status. The yellow warning sign indicates a dead node. OMNeT++ animates every significant event such as transmissions and inter-layer data transfers.

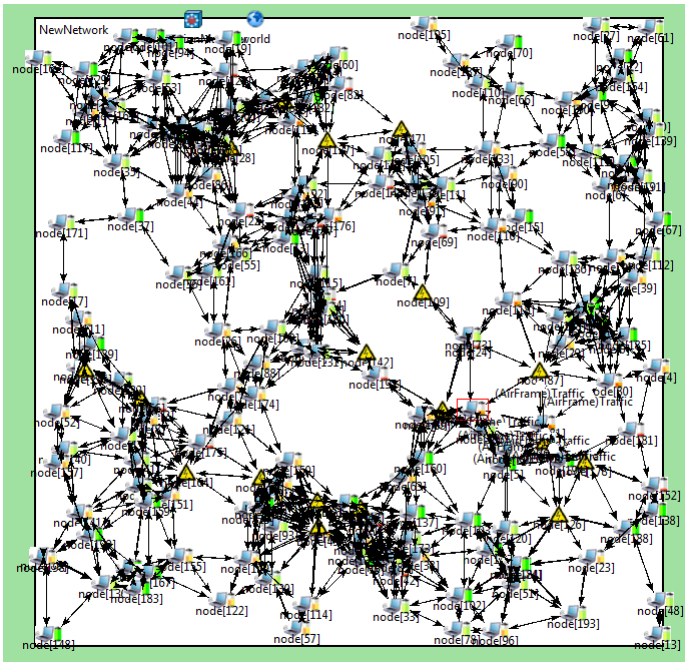

Figure 6.1: OMNeT++ network topology

6.3 Simulation Setting

For simulation purposes, 200 nodes are deployed in an 800 X 800 m^2 square area. The nodes are placed inside the square area at random coordinates with an uniform distribution. The channel rate is configured to 11 Mbps. Variable k, the coding variable,

is set to 10. UDP flows between random source and destination nodes are generated at random intervals, ranging from bitrates of 200kbps to 2Mbps. The offered load is regulated by adjusting the number of data flows. Five random topologies are used for each simulation setting.

In each case, the relevant simulation parameters are recorded from the five topologies, and the averages of those results are calculated.

6.4 Simulation 1 - Coding Factor

In the first simulation, we aim to explore the effects of different time periods assigned to the *coding factor*. An overly long time period may cause inefficient delays that cause packets to pile up, in effect, decreasing the throughput. A very short time period will cause the nodes to miss coding opportunities, cancelling the benefits of NC altogether. The purpose of this simulation is to find a near optimal time period to assign to the coding factor.

The simulations were conducted for various time periods. The throughput and coding parameters were measured. At time periods 1ms, 5ms and 25ms, the effect of different time period magnitudes can be clearly seen. The throughput parameter for each of these periods is depicted in figure 6.2.

The simulation results introduce some interesting phenomenons. As expected, when the time period is very small, as in the case of 1ms, it does not leave enough room for coding opportunities. This phenomenon causes the nodes to miss valuable coding opportunities and in doing so decreases performance. On the other hand, when the time period is excessively large, as with 25ms, the packets pile up at forwarder nodes, also causing a decrease in performance.

The number of packets coded in the simulation is depicted in figure 6.3. It can be seen here that the number of coded packets increases as the coding factor time period

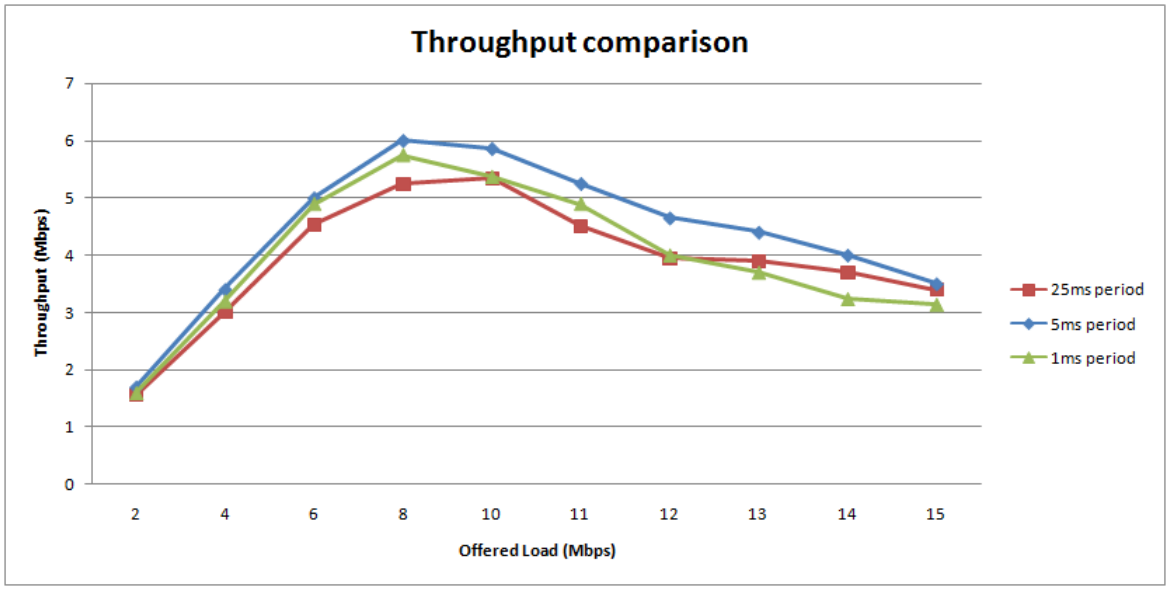

Figure 6.2: Throughput comparison of different coding factor time periods

increases. Although a 25ms period delivers a significant amount of coded packets, the benefits of the coding is nullified as packets pile up at intermediate nodes. On the other hand, when the period is too small, as with 1ms, less coding opportunities appear.

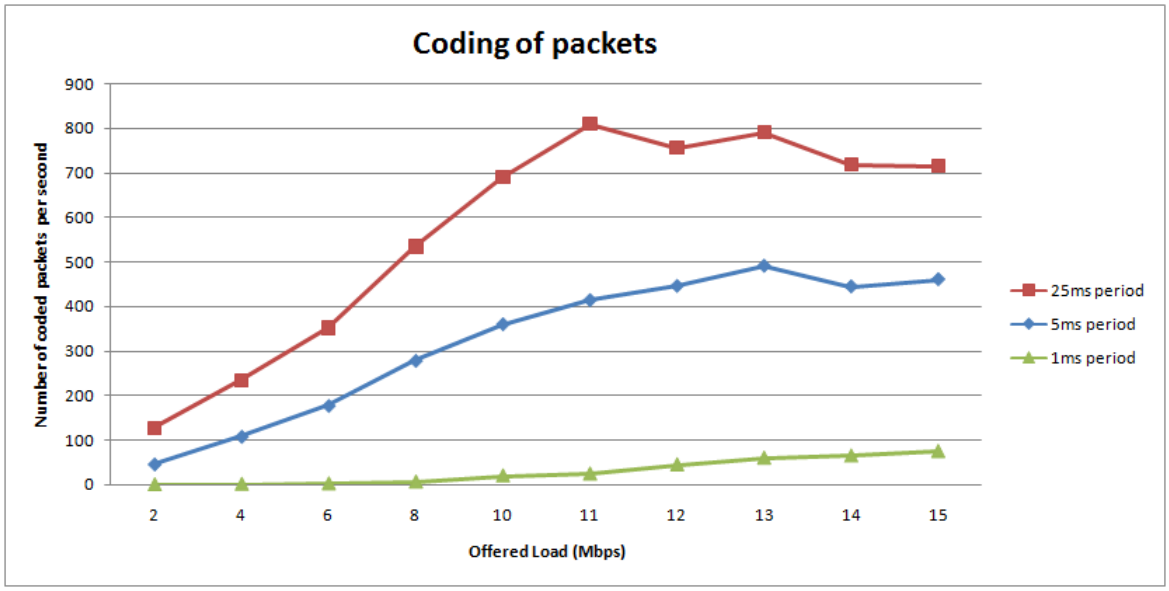

Figure 6.3: Coding comparison of different coding factor time periods

The simulation results are summarised in table 6.1.

Table 6.1: Throughput comparison of different coding factor time periods

	1ms period	**5ms period**	**25ms period**	
Load (Mbps)	Throughput (Mbps)	Throughput (Mbps)	Throughput (Mbps)	
2	1.6	1.7	1.56	
4	3.2	3.4	3	
6	4.9	5	4.53	
8	5.75	6	5.25	
10	5.37	5.86	5.34	
11	4.9	5.25	4.5	
12	4	4.65	3.94	
13	3.7	4.4	3.9	
14	3.24	4	3.7	
15	3.14	3.5	3.4	
	3.98	4.38	3.91	**Average**
8	5.7	6	5.25	**Maximum throughput**

In this case, the 5ms period resulted in the best throughput. It is not extensively long to cause packets to pile up, but it is long enough to allow a sufficient number of valuable coding opportunities at the intermediate nodes.

6.5 Simulation 2 - Distance Factor Incorporated

The *distance factor* merely prioritises forwarder nodes based on geographical distances, causing the forwarder node closest to the destination to forward the packets without delay. Therefore, when implemented separately, the size of the delay periods of the other forwarder nodes only require to be large enough to ensure that the node closest to the destination transmits. The assigned time periods are, therefore, somewhat insignificant regarding the network performance since the purpose of those delays are only to be large enough to prevent forwarder nodes, other than the primary forwarder, to transmit. Other forwarder set nodes will only forward traffic in the highly unlikely event where the primary forwarder fails.

As soon as other conditions are included in the routing process, scenarios arise where the time period assigned to the distance factor plays a more significant role. When a node is prioritised according to more than one routing condition, the magnitude of the distance time period now becomes imperative since it will determine how much priority this condition has. Therefore, with the coding factor incorporated, various time periods for the distance factor will be tested.

The optimal coding factor time period calculated in the previous simulation, that is the 5ms period, will be used as the baseline for this simulation. In this case, different time periods will be assigned to the distance factor, causing changes to its level of priority. For each time period, the throughput and number of coding will be measured.

This simulation will, firstly, give an indication of the results that can be expected when combining NC and GR, and secondly, show how different priorities of these methods effect the performance of a network.

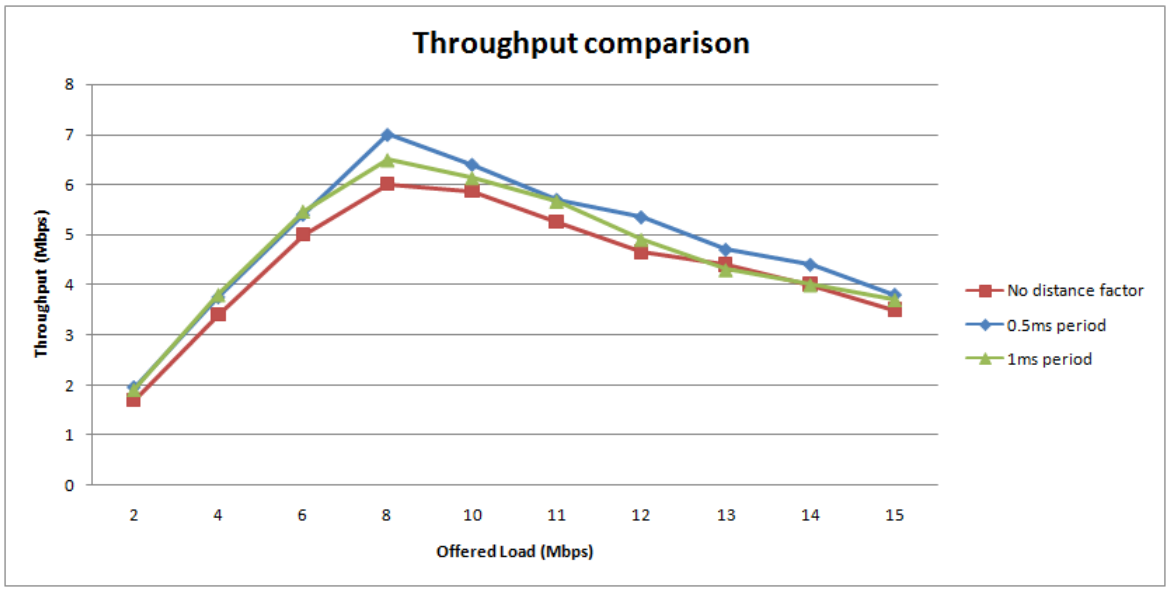

Figure 6.4: Throughput comparison of different distance factor time periods

The simulation results are depicted in figure 6.4. The results are very promising. A maximum throughput increase of 16.67% is achieved when GR is incorporated with NC. These results show that it is indeed beneficial to combine these methods in a single routing protocol. Although NC outperforms GR when they are compared individually, there are, as the results also corroborate, certain scenarios in ONC that benefits from GR.

As depicted in the figure, the 0.5ms time period shows a good increase in overall throughput. When the time period of the distance factor is increased to 1ms, a decrease in performance can be seen. This occurrence shows that when the distance factor receives higher priority, performance is decreased. This phenomenon is attributed to the fact that as the priority of the distance factor increases, it reaches a point where its level of priority and contribution outweighs the benefits it brings to the table. Hence, as the distance factor gains priority, the coding factor decreases in priority.

The amount of coding for the different periods is depicted in figure 6.5. It can be seen that as the distance factor time period increases to 1ms, the amount of coding decreases.

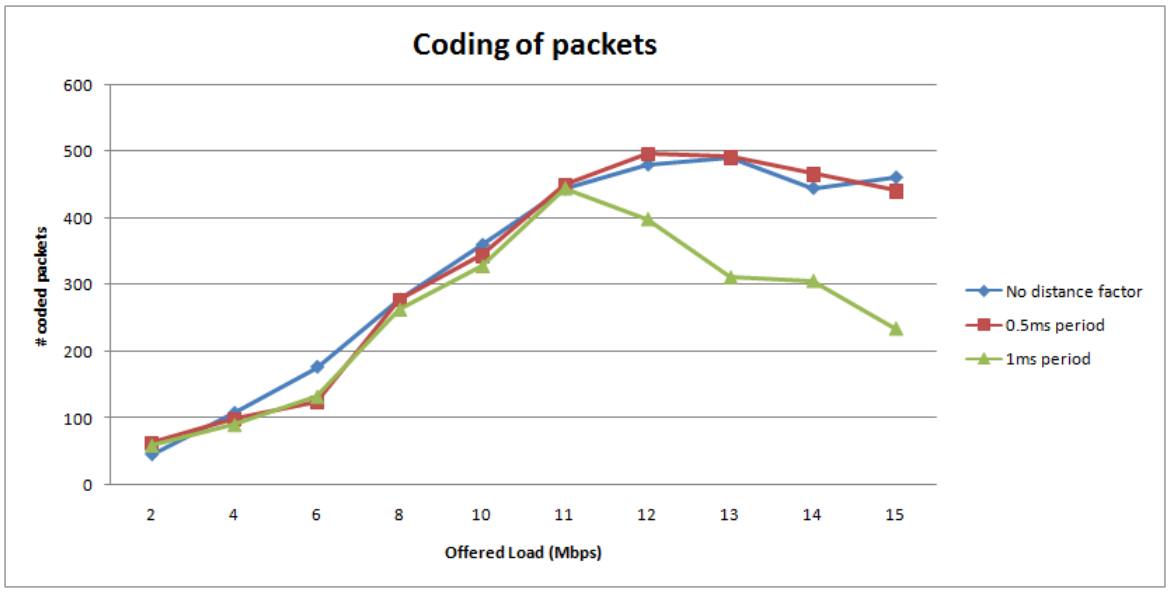

Figure 6.5: Coding comparison of different distance factor time periods

The simulation results are summarised in the table 6.2. In the table, the **percentage throughput increase** column represents a comparison between the **No distance factor** and **0.5ms period** columns.

6.6 Simulation 3 - Energy Factor Incorporated

The purpose of this experiment is to determine the effect of incorporating the energy aware condition into the routing scheme. Here, each node is assigned a random battery status allowing approximately fifteen to a maximum of thirty data transmissions per node. The sender nodes are given unlimited battery power to allow them to continuously produce flows of data. The throughput and the number of dead nodes will be measured at random time intervals. These values will be measured for the routing protocol excluding the *energy factor*, and for energy factor periods of 2.5ms and 5ms.

The simulation results are depicted in figure 6.6. The incorporation of energy based conditions displays significant improvement in node survivability. Without the inclusion of the energy factor, flows between source and destination nodes stop at 17

Table 6.2: Throughput comparison of different distance factor time periods

	No distance factor	**0.5ms period**	**1ms period**		
Load (Mbps)	Throughput (Mbps)	Throughput (Mbps)	Throughput (Mbps)	% Throughput increase	
2	1.7	1.95	1.9	14.7	
4	3.4	3.75	3.8	10.3	
6	5	5.4	5.47	8	
8	6	7	6.5	16.67	
10	5.86	6.4	6.14	9.22	
11	5.25	5.7	5.67	8.6	
12	4.65	5.35	4.9	15	
13	4.4	4.7	4.3	6.82	
14	4	4.4	4	10	
15	3.5	3.8	3.7	8.6	
	4.38	4.85	4.64	10.8	**Average**
8	6	7	6.5	16.67	**Maximum throughput**

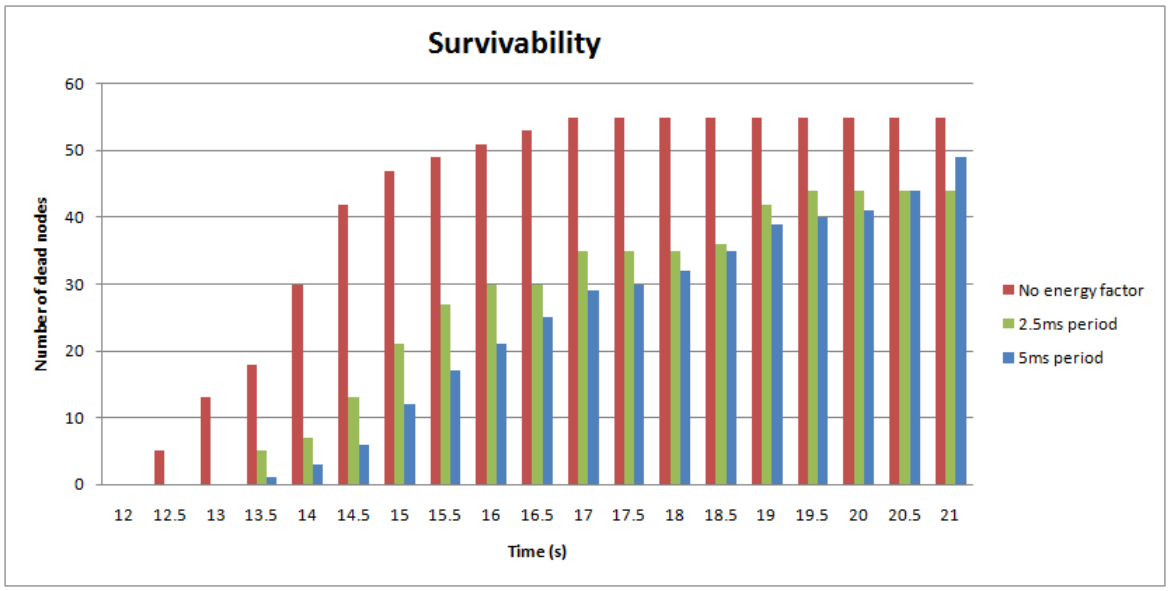

Figure 6.6: Survivability comparison of different energy factor time periods

seconds with 55 dead nodes. For a period of 2.5ms, the flows stop at 19.5 seconds with 44 dead nodes. For a period of 5ms, the flows stop at 21 seconds with 49 dead nodes.

The reason for this phenomenon is that, as the energy condition period increases, the priority of the other conditions decreases. Thus, the coding condition priority decreases. Therefore, as the energy condition period increases, the routing protocol converges to a certain point where the decreased priority of the coding condition causes fewer nodes to perform coding. At this stage, more nodes are required to perform routing, therefore, resulting in a larger number of node deaths.

The throughput of the respective periods is depicted in figure 6.7. The throughput decreases as the energy factor increases beyond a certain point. This phenomenon is attributed to the fact that as the energy factor time period increases, the contribution of the other factors decreases, causing decrease in throughput. From the figure, it can be seen that a time period increase from 2.5ms to 5ms causes a decrease in throughput.

The survivability results are summarised in the table 6.3.

Table 6.3: Survivability comparison of different energy factor time periods

	No energy factor	**2.5ms period**	**5ms period**	
Time(s)	# Dead nodes	# Dead nodes	# Dead Nodes	
12	0	0	0	
12.5	5	0	0	
13	13	0	0	
13.5	18	5	1	
14	30	7	3	
14.5	42	13	6	
15	47	21	12	
15.5	49	27	17	
16	51	30	21	
16.5	53	30	25	
17	55	35	29	
17.5	55	35	30	
18	55	35	32	
18.5	55	36	35	
19	55	42	39	
19.5	55	44	40	
20	55	44	41	
20.5	55	44	44	
21	55	44	49	
	12.2	13.2	13.45	**Time of first dead node (s)**
	17	19.5	21	**Time of source-destination isolation (s)**

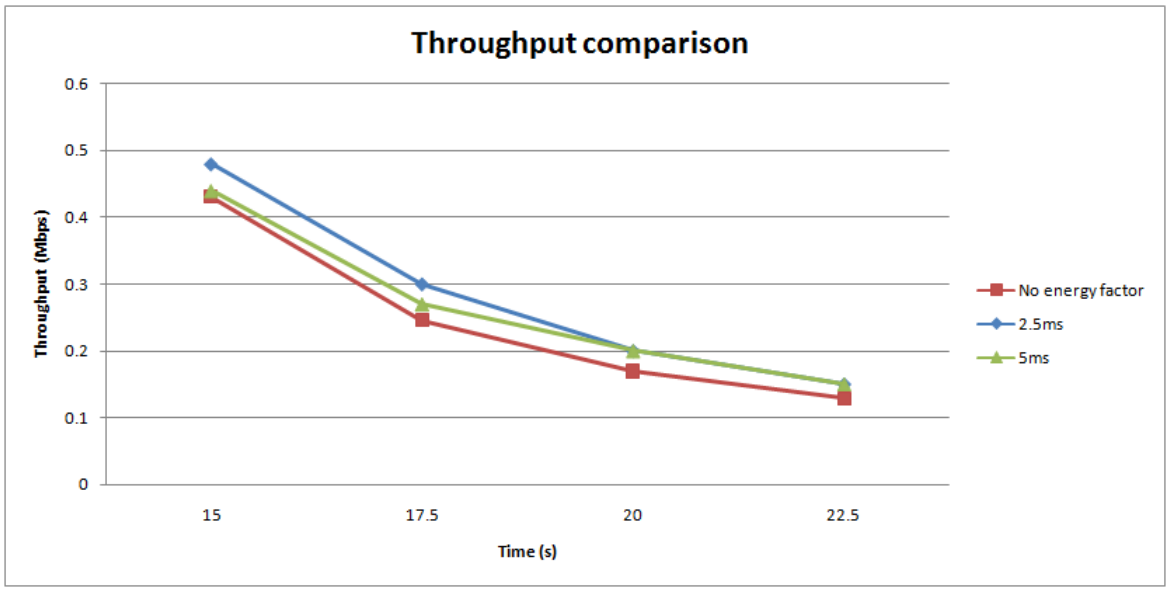

Figure 6.7: Throughput comparison of different energy factor time periods

6.7 Conclusion

In this chapter, we conducted various experiments. In the first experiment, we assigned different *coding factor* time periods to the routing scheme. The results obtained from the experiment revealed the effects of different time periods on network throughput and coding.

The second experiment analysed different priorities of the *distance factor* in addition to the coding factor. The results indicate that when the priority level of the distance factor is increased beyond a certain point, decrease in network performance follows. A significant increase in throughput is obtained when the distance factor is incorporated with an optimal priority.

The purpose of the final experiment was to test the effect that the incorporation of the *energy factor* has on the survivability of a network. The results indicate a promising decrease in the number of dead nodes at measured time intervals. The inclusion of the energy factor also proves to prolong the connectivity time between source and destination nodes.

Chapter 7

Conclusion

In this chapter, we take a brief overview of the research work. We will provide concluding remarks pertaining to the conducted research and give recommendations for future work

---

7.1 Introduction

Firstly, we will discuss the research hypothesis. We will then take brief overviews of the respective chapters constituting the thesis. At each chapter overview, we will discuss the purpose and contribution of the respective chapter. We will conclude with recommendations for future work.

7.2 Research hypothesis

In chapter 1, *Introduction*, we performed a literature survey regarding NC, GR, OR and energy awareness. The purpose of the literature survey was to provide background pertaining to the discovery of an area for research. We presented the proposed research and gave motivation to support the importance and contribution of the research.

The following hypothesis was made:

Using one of the key elements OR is comprised of, namely the *prioritising time delay calculation* phase, we can include NC, GR and energy aware routing into a single specialised routing scheme. This scheme will combine the benefits of the respective incorporated routing conditions.

We will now take brief overviews of the remaining chapters, starting with chapter 2.

7.3 Literature study

In chapter 2, *Literature Study*, we researched various areas pertaining to the research question. The purpose of the literature study was to equip us with the necessary tools required for finding a solution for our proposed research.

7.4 Development and implementation of routing scheme

In chapter 3, *Routing Scheme Design*, we developed the routing scheme. The development incorporated NC based on CORE [34], GR and energy aware conditions into as single specialised routing protocol. OR was used as the baseline for incorporating these routing methods. The development of the scheme was discussed in chapter 3.

The developed routing scheme was implemented in OMNeT++. The implementation details were discussed in chapter 4. A discussion on the respective modules and layers included in the implementation followed.

7.5 Validation and verification

In Chapter 5, *Validation and Verification*, the implementation of the developed scheme was verified and validated. An array of scenarios were created in which the underlying operation of the individual features could be perceived separately. Every scenario was constructed and simulated in OMNeT++ and the results were discussed with regard to the verification or validation at hand. Furthermore, also serving as validation, one leg of the implemented model was compared to an already validated model. The comparison showed similar results. This confirmed the validity of our implementation.

7.6 Simulation results

In chapter 6, we conducted various experiments. The results obtained from the simulated experiments were promising. The combination of the NC and GR showed to be beneficial for the network with regard to increase in throughput. Furthermore, the inclusion of the energy factor indicated a significant increase in the survivability of a network. The increase achieved in the aforementioned network performance metrics

is attributed to the fact that different scenarios or network characteristics benefit from different routing techniques.

7.7 Correctness of hypothesis

A routing scheme was designed. OR was used as the underlying basis for including NC, GR and energy aware routing in a single specialised routing protocol. The design was successfully implemented in OMNeT++ MiXiM. Various carefully planned simulations were conducted using the implemented model. The simulation results indicated increase in throughput and survivability, both of which were improvements that were attributed to the inclusion of the various routing methods. The gathered simulation results corroborate and confirm that our hypothesis is indeed correct.

7.8 Future Work

By incorporating different routing conditions in the *priority based forwarding* delay calculation phase in OR, a routing protocol was developed with improved throughput and survivability. By changing current conditions or even by introducing new ones, many more performance metrics and or network characteristics can be addressed. Future work include the optimisation of different combinations of current and possibly new prioritising conditions for priority based forwarding.

Bibliography

[1] R. Sargent, "Verification and validation of simulation models," in *Winter Simulation Conference (WSC), Proceedings of the 2010*, December 2010, pp. 166 –183.

[2] D. Feng, Y. Zhu, and X. Luo, "Cooperative incentive mechanism based on game theory in manet," in *Networking and Digital Society, 2009. ICNDS '09. International Conference on*, vol. 2, May 2009, pp. 201 –204.

[3] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 69–74, January 2004. [Online]. Available: http://doi.acm.org/10.1145/972374.972387

[4] S. Wang, X. Gao, and L. Zhuo, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, December 2009, pp. 1 –5.

[5] M. Zorzi and R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 337 – 348, 2003.

[6] H. G. Gauch, *Scientific Method in Practice*, 1st ed. Cambridge University Press, 2003.

[7] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, no. 9, pp. S23 – S30, 2005.

[8] P. Yang and B. Huang, "Multi-path routing protocol for mobile ad hoc network," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 4, December 2008, pp. 1024 –1027.

[9] S. Dhurandher, S. Khairwal, M. Obaidat, and S. Misra, "Efficient data acquisition in underwater wireless sensor ad hoc networks," *Wireless Communications, IEEE*, vol. 16, no. 6, pp. 70 –78, December 2009.

[10] A. Mitrokotsa, N. Komninos, and C. Douligeris, "Intrusion detection with neural networks and watermarking techniques for manet," in *Pervasive Services, IEEE International Conference on*, July 2007, pp. 118 –127.

[11] Y. Yoo and D. P. Agrawal, "Why does it pay to be selfish in a manet?" *Wireless Communications, IEEE*, vol. 13, no. 6, pp. 87 –97, December 2006.

[12] E. Spaho, L. Barolli, G. Mino, F. Xhafa, and V. Kolici, "Vanet simulators: A survey on mobility and routing protocols," in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on*, October 2011, pp. 1 –10.

[13] V. Potdar, A. Sharif, and E. Chang, "Wireless sensor networks: A survey," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, May 2009, pp. 636 –641.

[14] S. Wang, X. Gao, and L. Zhuo, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, December 2009, pp. 1 –5.

[15] W. Li, J. Bao, and W. Shen, "Collaborative wireless sensor networks: A survey," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, October 2011, pp. 2614 –2619.

[16] P. K. Sahoo and J.-P. Sheu, "Limited mobility coverage and connectivity maintenance protocols for wireless sensor networks," *Computer Networks*, vol. 55,

no. 13, pp. 2856 – 2872, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611002167

[17] B. A. Forouzan, *Data Communications and Networking*, 4th ed. McGraw-Hill, January 2009.

[18] M. Raciti, J. Cucurull, and S. Nadjm-Tehrani, "Energy-based adaptation in simulations of survivability of ad hoc communication," in *Wireless Days (WD), 2011 IFIP*, October 2011, pp. 1 –7.

[19] S. Zaman and F. Karray, "Tcp/ip model and intrusion detection systems," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, May 2009, pp. 90 –96.

[20] D. Meyer and G. Zobrist, "Tcp/ip versus osi," *Potentials, IEEE*, vol. 9, no. 1, pp. 16 –19, February 1990.

[21] W. Stallings, "Ieee 8o2.11: wireless lans from a to n," *IT Professional*, vol. 6, no. 5, pp. 32 – 37, October 2004.

[22] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, vol. 2, October 1996, pp. 392 –396 vol.2.

[23] M. Sun, K. Sun, and Y. Zou, "Analysis and improvement for 802.15.4 multi-hop network," in *Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on*, vol. 2, January 2009, pp. 52 –56.

[24] W. Woon and T. Wan, "Performance evaluation of ieee 802.15.4 ad hoc wireless sensor networks: Simulation approach," in *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, vol. 2, October 2006, pp. 1443 –1448.

[25] R. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint*

Conference of the IEEE Computer and Communications. IEEE Societies, vol. 1, March 2003, pp. 702 – 711 vol.1.

[26] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, ser. MobiHoc '02. New York, NY, USA: ACM, 2002, pp. 146–156. [Online]. Available: http://doi.acm.org/10.1145/513800.513818

[27] J. Vazifehdan, R. Prasad, E. Onur, and I. Niemegeers, "Energy-aware routing in wireless ad hoc networks with mains-powered nodes," in *Future Network and Mobile Summit, 2010*, June 2010, pp. 1 –12.

[28] P. Chou and Y. Wu, "Network coding for the internet and wireless networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77 –85, September 2007.

[29] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, January 2006. [Online]. Available: http://doi.acm.org/10.1145/1111322.1111337

[30] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 497 –510, June 2008.

[31] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," in *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, March 2006, pp. 62 –66.

[32] M. Grobler and A. Helberg, "Identifying opportunities for deterministic network coding in wireless mesh networks," *SAIEE African Research Journal*, vol. 102, pp. 8 – 15, 2011.

[33] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413 –4430, October 2006.

[34] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "Core: A coding-aware opportunistic routing mechanism for wireless mesh networks [accepted from open call]," *Wireless Communications, IEEE*, vol. 17, no. 3, pp. 96 –103, 2010.

[35] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Computer Communication Review*, vol. 37, no. 4, pp. 169–180, October 2007.

[36] Z. Zhong, J. Wang, S. Nelakuditi, and G.-H. Lu, "On selection of candidates for opportunistic anypath forwarding," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 4, pp. 1–2, October 2006. [Online]. Available: http://doi.acm.org/10.1145/1215976.1215978

[37] C. Westphal, "Opportunistic routing in dynamic ad hoc networks: the oprah protocol," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, October 2006, pp. 570 –573.

[38] Y. Yuan, H. Yang, and H. Starsky, "Romer: resilient opportunistic mesh routing for wireless mesh networks," *IEEE Workshop on Wireless Mesh Networks*, vol. 1, September 2005.

[39] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in omnet++ the mixim vision," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 71:1–71:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416302

[40] J. P. Kleijnen, "Verification and validation of simulation models," *European Journal of Operational Research*, vol. 82, no. 1, pp. 145 – 162, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377221794000166

Appendix A

Conference Contributions from this Thesis

A.1 An Improved Method for the Identification of Opportunities for Deterministic Network Coding

Presented at:

Southern Africa Telecommunication Networks and Applications Conference (IEEE AFRICON)

September 2011

An Improved Method for the Identification of Opportunities for Deterministic Network Coding

Mario J. Engelbrecht, Magdalena J. Grobler and Melvin Ferreira
School of Electrical, Electronic and Computer Engineering
North-West University, Potchefstroom, South Africa
Email: 20704313@nwu.ac.za

***Abstract*—In this paper a new method for the identification of sub-topologies, suitable for the implementation of network coding in a wireless mesh network is presented. When nodes are positioned in certain locations relative to each other they create an environment where network coding opportunities can be identified for instances where certain data flows occurs between nodes in that environment. If nodes forming those sub-topologies can be informed of the occurring coding opportunities, it would improve the performance of the network.**

***Index Terms*—Network coding, Network Topology Identification, Wireless Mesh Networks.**

I. Introduction

Wireless mesh networks (WMNs) are ad-hoc networks that are self-established and self-maintained by the nodes that form it [1]. Network Coding (NC) is a technique that maximizes the throughput in ad-hoc networks [2]. The throughput of a network can be defined as the number of packets successfully delivered in a time unit [3]. There are different ways to perform network coding.

In the case of Random Linear Network Coding (RLNC), packets arriving at an encoding node, are randomly combined as linear combinations of the original packets. This type of NC is independent of the topology of the network, because a coding vector is added by the encoding node, to the header of the encoded packet. The receiver node then waits for a sufficient number of linearly independent packets to reach it, and then decodes the original packets.

In the case of Deterministic Network Coding (DNC), coding opportunities are identified beforehand, and fixed nodes are assigned encoding or decoding roles [4]. There are no coding vectors needed, because decoding nodes will always have prior knowledge of how the packets were encoded. DNC therefore are heavily reliant on the topology of the network and it was therefore a challenge to implement DNC in WMNs, of which the topology may change frequently. In 2011 Grobler and Helberg [5] suggested a method to identify opportunities for the implementation of DNC in WMNs. In their work, they identify the opportunities, by searching for known "Network Coding suitable sub-topologies" within the larger network topology.

An algorithm used to perform this search must be as accurate as possible to prevent it from missing coding opportunities. It must also use as little resources as possible and find topologies quickly as the topology of a WMN may change frequently. In this paper a new search and identification method is proposed. This new method is more accurate and uses fewer resources than the methods described in [5].

The remainder of this paper is arranged as follows: Background on WMNs and NC is provided in section II. Section III describes the method followed to design the new algorithm. Implementation details are given in section IV. The results obtained in section IV are compared to those of [5] in section V in terms of performance and accuracy. A conclusion is drawn in section VI.

II. Background

A. Wireless Mesh Networks

Wireless mesh networks (WMNs) are ad-hoc networks that are self-established and self-maintained by the nodes forming the network [1]. The nodes can either be in fixed locations or be mobile. WMNs consist of mesh clients and mesh routers. Mesh routers form the backbone of the WMN and enables communications between different wireless network access technologies. Mesh clients only have single access technology interfaces and don't have bridge or gateway functions. This allows for simpler hardware and software inside a mesh client than a mesh router. Nodes with wireless network interface cards (NICs) can gain access to a WMN by wireless connections to the mesh routers. Nodes without wireless NICs can also gain access to a WMN by connecting to a mesh router through other access technologies, for example, Ethernet [1].

B. Network Coding

Network coding is a technique that allows the optimization of throughput in networks. It can achieve the throughput predicted by the Min-cut Max-flow theorem, which states that the minimum cut of a network between the source and the destination nodes in a network, where there are no flow capabilities between the source and the destination, is equal to the maximum throughput of the network [3]. This concept is depicted in figure 1. In this example, node S is the source and nodes X and Y are destination nodes. All links in the example have unit capacity. When applying the theorem, it can be seen that the Min-cut of this example, and therefore also the Max-flow (maximum achievable throughput) is 2.

In [6] the author concludes that the fraction of all the data transferred that is actually coded in a network is very low when compared to theory and ideal environments. But even

though the results obtained show network coding not to be as good as in theory, still a throughput gain of more than 10% can be achieved by implementing network coding [6].

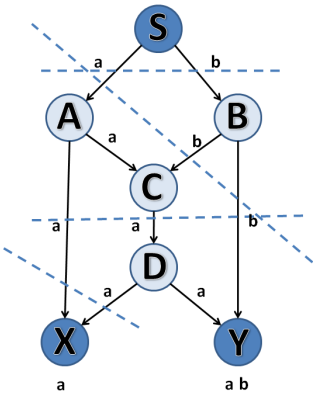

Fig. 1. Min-cut max-flow theorem

This maximum throughput can **only** be achieved through the use of network coding, explanation follows. In figure 1 (an example of traditional routing) the total number of 2 messages were sent by the source and a total of 3 messages were received by the destination nodes after a single round of transmissions. This results in a throughput of 1.5. If network coding is used, as depicted in figure 2, messages *a* and *b* are encoded (XORed) at node *C*. This allows the destination nodes *X* and *Y* to receive the encoded packet, and decode it (XOR) using the information it already received (from nodes *A* and *B*), resulting in a total number of 4 messages after a single round of transmissions. This translates to a throughput of 2.

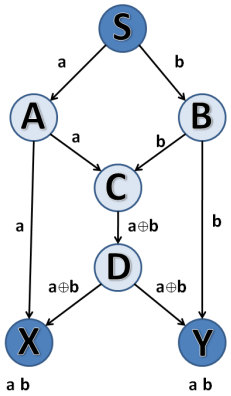

Fig. 2. Network with network coding

1) Opportunistic Implementation of DNC: Depending on the positions of the nodes and the connections formed inside a WMN, certain "known" NC suitable topologies can be used to identify where and how NC can be implemented [5]. In doing this, DNC can be implemented opportunistically each time a "known" NC suitable topology is formed as a sub-topology in a WMN. Certain coding opportunities will arise depending on the data transfer between relevant nodes that cause a flow of multiple data packets through nodes with encoding and decoding capability.

C. Identification of Topologies

In [5], three different methods were introduced to find the sub-topologies within a larger network: Iterative looping, Cross-correlation and Concatenation. Typical sub-topologies that the authors of [5] used, include the "bow-tie" and the "butterfly" topologies depicted in figure 3.

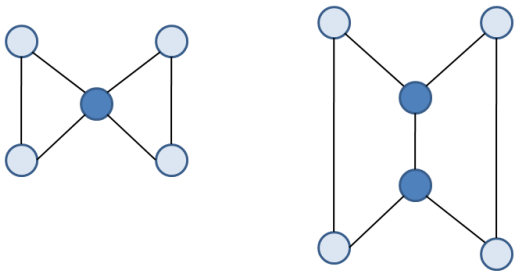

Fig. 3. Network coding topologies a) Bow-tie b) Butterfly

In this paper a new method for identifying these sub-topologies is presented.

III. Methodology

The main idea of the new method concerning the discovery of topologies is that the search begins from within the nodes themselves. The nodes must know to what other nodes they are connected as well as those nodes' connections to other nodes. Each node and its connections are analysed individually to determine whether it exists inside a NC topology. The topology identification method was implemented in MATLAB® and compared to the concatenation method of [5] to determine whether it was an improved method for identifying topologies.

A virtual wireless ad-hoc network with nodes positioned at random coordinates was created. Connections between the nodes are shown. The wireless ad-hoc network was constructed from vertices and edges. The vertices represent the nodes and the edges represent the connections between the nodes. Random coordinates was generated and vertices was placed at those coordinates within a certain square area. Connectivity between the nodes was determined according to the specified communication radius of each node. The nodes have a default color of blue, but whenever a node is identified as a possible network coding node, it turns to red. The connections have a default color of green but turns to red if it is part of a network coding topology.

The connectivity matrix is used as a base for identifying the topologies. The connectivity matrix is modified according to the conditions for a specific topology. Modification will take place until the matrix only contains the relevant data that, through simple analysis, reveals the nodes involved. This procedure takes place for every node with enough connections

to give it the ability to possibly be one of the network coding nodes within a topology.

IV. Implementation Details

The topology identification was implemented in MATLAB®. The steps executed in the identification process for each topology will now be explained.

The following terms are properly defined to simplify the process of understanding the work.

Network coding node - A node which may be assigned network coding capabilities because it exists within a network coding suitable sub-topology.

Possible network coding node - A node with some identified network coding node requirements.

Network coding topology - A topology where network coding opportunities are valid, such as the bow-tie or butterfly topology.

The upper triangle of the connectivity matrix (excluding the diagonal) is manipulated for every possible network coding node by eliminating irrelevant and inapplicable nodes until it reveals whether the analysed node is a network coding node inside a network coding topology or not.

The steps for identifying a bow-tie:

- Step 1: Determine all nodes connected to four or more nodes. These nodes have the possibility of being a network coding node inside a bow-tie topology. For each of them execute the following steps within the connectivity matrix:
- Step 2: Eliminate all the connections of the node itself and also of the nodes not connected to this specific node by replacing the 1's with 0's.
- Step 3: Calculate the sum of each column and save the result in an array.
- Step 4: Calculate the sum of each row and save it in an array.
- Step 5: Any number other than zero represents a one in the results obtained above. Perform the Boolean operation OR between the two results. If the sum of the 1's in the OR result is equal or greater than four then this node is the network coding node in a bow-tie topology. The connections left in the connectivity matrix are the connections between the nodes forming the bow-tie topology.

Bow-tie example:

A randomly generated WMN has the following connectivity matrix:

$$CM = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

We will only perform actions on the upper triangle (excluding the diagonal) hence our matrix looks as follow:

$$CM = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Node 6 is one of the nodes identified in step 1. After step 2 is performed on node 6 the connectivity matrix looks as follow:

$$CM = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The sum of each column results in the following array:

$$Sum_{Col} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The sum of each row results in the following array:

$$Sum_{Row} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The OR result between the arrays looks as follows:

$$OR_{Result} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The sum of the elements in the array is equal to four and therefore a bow-tie is found. The remaining elements inside the modified connectivity matrix show the nodes and the connections that form the bow-tie with node six. In this case nodes 1, 2, 3 and 5 form the bow-tie where the wings are formed with the connections between node 1 and 2 and between node 3 and 5.

The output simulation for the bow-tie example is shown in figure 4.

The steps for identifying a butterfly:

- Step 1: Determine all nodes connected to three or more nodes. These nodes have the possibility of being a network coding node.
- Step 2: Eliminate all the connections of the node identified in step 1 by replacing the 1's with 0's inside the connectivity matrix.
  For each of the nodes identified in step 1 execute the following steps to find the other possible network coding node to form the butterfly topology.
- Step 3: Determine all nodes connected to three or more nodes. For each of them execute the following steps:
- Step 4: Eliminate all the connections of the nodes not connected to this specific node (identified in step 3) or

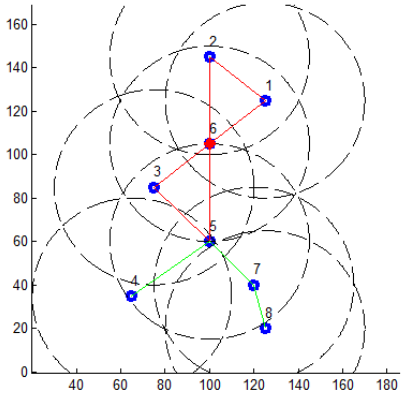

Fig. 4. Bow-tie example output in MATLAB®

the other possible network coding node (identified in step 1) in the connectivity matrix by replacing the 1's with 0's.

- Step 5: Eliminate all the node connections connected to this specific node.
- Step 6: Eliminate nodes connected to both possible network coding nodes.
- Step 7: Eliminate connections between nodes where both nodes are connected to one of the possible network coding nodes.
- Step 8: Calculate the sum of each column and save the result in an array.
- Step 9: Calculate the sum of each row and save it in an array.
- Step 10: Any number other than zero represents a one in the results obtained above. Perform the Boolean operation OR between the two results. Sum all the 1's in the OR result. If the total is equal to four then a butterfly topology is found. The connections left in the connectivity matrix are the connections between the nodes forming the butterfly topology.

Butterfly example:
A randomly generated WMN has the following connectivity matrix:

$$CM = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

We will only perform actions on the upper triangle (excluding the diagonal) hence our matrix looks as follow:

$$CM = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Node 7 is one of the nodes identified in step 1. Node 4 is one of the nodes identified in step 3. After steps 4 to 6 are performed on node 4 and 7 the connectivity matrix looks as follow:

$$CM = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The sum of each column results in the following array:

$$Sum_{Col} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The sum of each row results in the following array:

$$Sum_{Row} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The OR result between the arrays looks as follows:

$$OR_{Result} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The sum of the elements in the array is equal to four and therefore a butterfly is found. The remaining elements inside the modified connectivity matrix show the nodes and the connections that form the butterfly with node 4 and 7. In this case nodes 1, 3, 5 and 6 form the butterfly where the wings are formed with the connections between node 3 and 5 and between node 1 and 6.
The output simulation for the butterfly example is shown in figure 5.

V. Experimental Results

The new search method was implemented in MATLAB®. The same hardware and software platforms are used for the simulations of both of the algorithms. The new presented method was compared to the concatenation method of [5] in terms of performance and accuracy. Random networks consisting of 16, 30 and 50 nodes were generated for testing purposes.

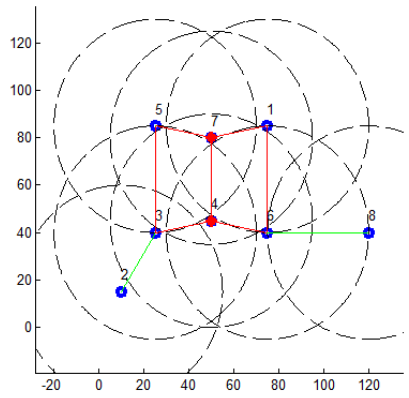

Fig. 5. Butterfly example output in MATLAB®

A. *Performance*

The significance of the performance differences are most effectively shown in the variation of average connections per node. The bow-tie search methods are compared in figure 6, 7 and 8. The comparison of the search times between the two methods can be seen in figure 6 for a network where the average connections per node are between two and five. In figure 7 for a network where average connections per node are between four and six. In figure 8 for a network where average connections per node are between six and eight. In each figure networks one to three represent random networks consisting of 16, 30 and 50 nodes respectively. The amount of time took for the completion of the search and identification process of bow-ties for each of these networks are shown for the concatenation method of [5] and the new presented method respectively.

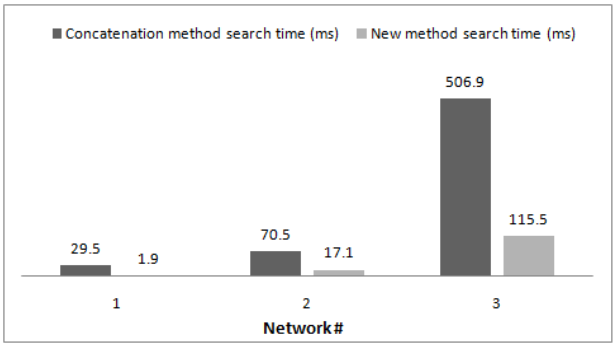

Fig. 6. Comparison of bow-tie search methods in a network with average connections per node between two and four

The butterfly search methods are compared in figure 9, 10 and 11. The comparison of the search times between the two methods can be seen in figure 9 for a network where the average connections per node are between two and five. In figure 10 for a network where average connections per node are between four and six. In figure 11 for a network where

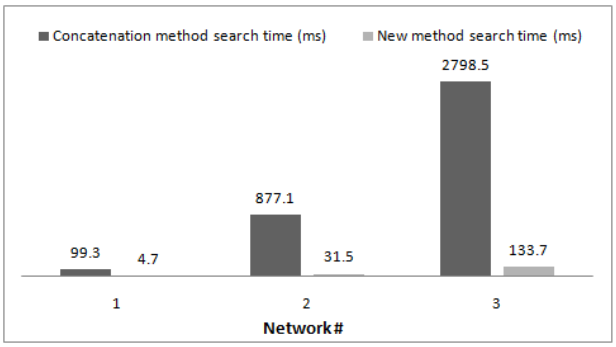

Fig. 7. Comparison of bow-tie search methods in a network with average connections per node between four and six

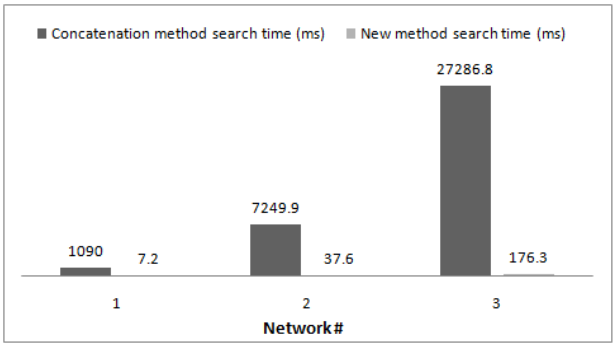

Fig. 8. Comparison of bow-tie search methods in a network with average connections per node between six and eight

average connections per node are between six and eight. In each figure networks one to three represent random networks consisting of 16, 30 and 50 nodes respectively. The amount of time took for the completion of the search and identification process of butterflies for each of these networks are shown for the concatenation method of [5] and the new presented method respectively.

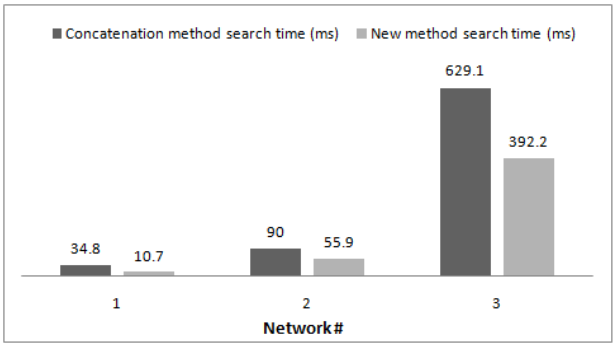

Fig. 9. Comparison of butterfly search methods in a network with average connections per node between two and four

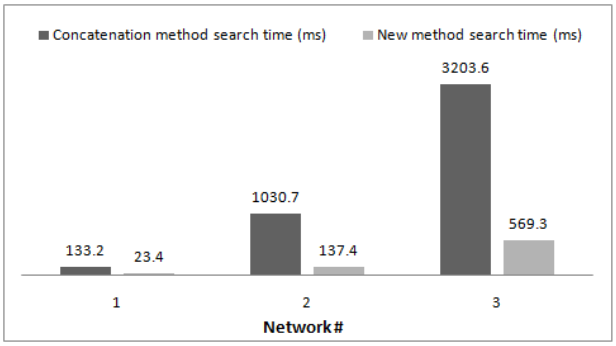

Fig. 10. Comparison of butterfly search methods in a network with average connections per node between four and six

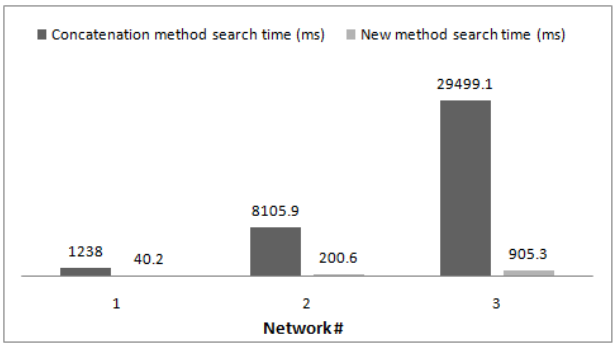

Fig. 11. Comparison of butterfly search methods in a network with average connections per node between six and eight

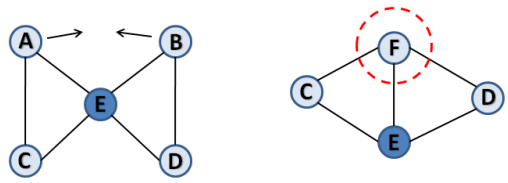

Fig. 12. Merged node scenario

B. Accuracy

The new identification method for a bow-tie is more accurate than the previous method as it only identifies bow-ties if it has four independent surrounding nodes where the previous method would also declare them as a valid bow-tie even when the nodes overlap as indicated in figure 12. In this figure a scenario is depicted where node A and node B are in fact one node as they are merged in node F.

VI. Conclusion

In this paper a new method for the search and identification of topologies suitable for network coding was presented. Both of the methods was implemented and simulated on the same hardware and software platforms. The following conclusions were made from the results. The results indicate a clear performance improvement of method 2 (the new algorithm) above method 1 for different network sizes and average node connections. WMNs changes dynamically as nodes join and leave the network. Therefore there is a consistent change of network topologies. This makes it essential that the fastest search method is to be used when identifying network coding topologies inside WMNs. In this case method 2 (the new algorithm) will be more suitable for practical opportunistic implementation of deterministic NC in WMNs.

References

[1] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, no. 9, pp. S23 – S30, 2005.

[2] S. Wang, X. Gao, and L. Zhuo, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, 2009, pp. 1 –5.

[3] B. A. Forouzan, *Data Communications and Networking*, 4th ed. McGraw-Hill, 2007.

[4] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "Core: a coding-aware opportunistic routing mechanism for wireless mesh networks [accepted from open call]," *Wireless Communications, IEEE*, vol. 17, no. 3, pp. 96 –103, 2010.

[5] M. Grobler and A. Helberg, "Identifying opportunities for deterministic network coding in wireless mesh networks," *SAIEE African Research Journal*, vol. 102, pp. 8 – 15, 2011.

[6] J. L. van Wyk, "Comparing network coding implementations on different open systems interconnection layers," Master's thesis, 2010.

A.2 Unselfish Network Coding Games

Presented at:

Southern Africa Telecommunication Networks and Applications Conference (SATNAC 2011)

September 2011

Unselfish Network Coding Games

Mario J. Engelbrecht and Magdalena J. Grobler
TeleNet Research Group
North-West University
Potchefstroom, South Africa
Email: {20704313, leenta.grobler}@nwu.ac.za

***Abstract*—Nodes in Mobile Ad Hoc Networks (MANETs) are mobile devices and are independent decision makers with regard to the execution of basic network functions. Network coding is a technique that aims to optimise the throughput of a network. Selfish nodes can exploit network coding for their own benefit while being non-cooperative towards other nodes. In this paper a network coding scheme needs to be devised that incorporates rules that gives selfish nodes reason to cooperate truthfully. We incorporate game theory to deliver a perfect environment for the modelling, analysis and optimisation of the described problem.**

***Index Terms*—Game theory, Mobile Ad Hoc Networks (MANETs), Network coding, Selfish nodes.**

I. Introduction

Mobile ad hoc networks (MANETs) are ad-hoc networks that are self-established and lack infrastructure [1]. All of the nodes in MANETs are mobile devices and are independent decision makers with regard to the execution of basic network functions [2]. Therefore if the nodes are operated by rational people they can be configured to behave selfish. Network coding (NC) is a technique that attempts to optimise the throughput of a network by coding packets [3]. NC relies on cooperation and truthfulness between nodes. Therefore in the presence of selfish nodes, NC is destined to fail when nodes exploit the benefits of NC while leaving other nodes at a disadvantage [5].

The remainder of this paper is arranged as follows: In section II, we give background on MANETs, NC and game theory. In section III, the similarity between game theory and MANETs are sketched. Section IV presents the proposed research after which the methodology is explained in section V. A conclusion is drawn in section VI.

II. Background

A. Mobile ad hoc networks (MANETs)

MANETs are ad-hoc networks that are self-established and lack infrastructure [1]. All of the nodes in MANETs are mobile devices such as laptops, PDAs or cellphones. Topologies in the network may change randomly at any given time because of the unpredictable mobility of the nodes. The fact that the nodes are mobile also indicate that they have limited energy and computational resources as well as limited transmission ranges [2]. Due to the limited transmission ranges within MANETs it is common for packets transferred in MANETs to hop between a number of nodes in order to reach its destination.

B. Network Coding

The main idea behind NC is to XOR packets into a single coded packet in order to reduce the number of transmissions required to transfer data. In figure 1, node A and node B both need to send messages to node X and node Y. Conventional routing techniques would require six transmissions to accomplish the task, but by implementing NC at node C it would only require four transmissions [4].

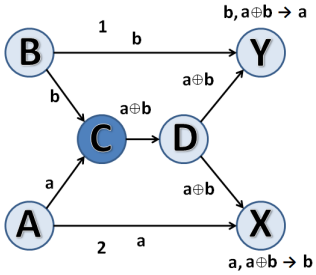

Fig. 1. Network with network coding

C. Selfishness in MANETs

Mobile nodes would behave seemingly selfish in order to protect their limited resources, and because nodes in MANETs each perform basic network functions by choice, they can manipulate the system to their own advantage [5]. When nodes decide to behave selfish, the performance of the network degrades significantly as alternative, maybe even more expensive routes need to be established to compromise for the selfish nodes.

1) Selfishness in network coding: Refer to figure 1. Connectivity on links *1* and *2* determines the possibility of decoding the coded packets at nodes Y and X respectively. Observe the following scenario. The delivery probability on link *1* from node B to node Y is 0.1. Therefore network coding would not be recommended at node C. But because data transfer along link *1* is of no interest to node B, only to node A, node B might lie about its connectivity along link *1* and tell node C that the delivery probability to node Y is 0.7. This will cause node C to perform NC. By doing so node B will increase its throughput by the same amount that node A's throughput will be decreased [5].

D. Game theory

The idea behind game theory is to model decision makers as players and complex decisions as moves as if it were a game. The success of each player's move is determined by the moves of all the other players in the game. A game is categorised depending on the nature of the situation it wishes to imitate [6].

1) Components in a game: According to game theory, a game has three main components. The core unit in game theory is the game G given as $G = \langle N, M, \{u_i\}\rangle$ where

$N = \{1, 2, ..., n\}$ is the set of players,

$M = M_1 \times M_2 \times ... \times M_n$ is a Cartesian product of all the move sets, where M_i is a move set of player i, that is the available moves player i can make, and

$\{u_i\} = \{u_1, ..., u_n\}$ is the set of utility functions [6]. The utility function is a function of all the moves performed inside a game. It gives the level of satisfaction a specific player

experiences with regard to all the moves made by all the players, hence $u_i : M \rightarrow \mathbf{R}$.

2) Nash equilibrium and Pareto optimality: A game may or may not converge to a Nash equilibrium depending on the available strategies the moves result to and the type of players. A Nash equilibrium occurs when none of the players can gain anything by changing his or her decision while all the other players keep their decisions unchanged. A Pareto improvement is made when a player can improve his game without worsening the success of any other player's game. A game is said to be Pareto optimal or efficient when no more Pareto improvements can be made [7].

3) Example: The following example illustrates the use of Game Theory: Two senators where caught accepting bribes. There wasn't enough evidence for a full conviction of both. The judge then devised the game depicted in figure 2 to solve the problem. If this game is played by rational players, as both senators are assumed to be, the game will converge to a Nash equilibrium, leading both to confess [7].

		Senator A	
		Confess	Don't confess
Senator B	Confess	3 yrs, 3 yrs	0 yrs, 10 yrs
	Don't confess	10 yrs, 0 yrs	1 yr, 1yr

Fig. 2. The Prisoner's Dilemma

III. Game Theory and MANETs

Mobile nodes in MANETs are rational of nature and ad hoc networks are competitive playing fields for nodes. Game theory has recently been used to model networks and nodes as games and players because of the similarity between ad hoc networks and games [6].

The rules of the game determine what moves a certain type of player will make in certain situations. Games usually converge to an equilibrium when all the players are satisfied with their decisions. In the same way, rules in a MANET can be changed to produce desired actions from the nodes that in turn produce desired outcomes for the entire network. The underlying motives for changing the rules can be encouraging or enforcing of nature. The components in a MANET can be mapped to the three components of a game as depicted in the table below [6].

Table 1. Game theoretical mapping of MANET elements

Components of a game	**Elements of a MANET**
Players	Mobile nodes
Strategy / Move set	Available decisions or actions related to a functionality
Utility function	Performance metrics (e.g. throughput, packet loss, latency, etc.)

IV. Proposed research

By modelling a MANET NC scenario in game theory, rules can be made, changed or taken away to alter the behaviour of the nodes. By carefully observing the change in outcome (behaviour of the nodes) produced for a changed set of inputs (rules), an optimal set of inputs can be found to produce desired results and develop a NC scheme that gives incentives for nodes to not act selfish.

V. Methodology

A *literature survey* was completed on MANETs, NC, game theory and selfishness in networks. The literature survey led to the following question: can a game theoretic approach aid in the development of a network coding scheme that encourages nodes to cooperate in a MANET?

A *detailed literature study* will be performed to show what is done, how it works and to indicate the challenges and issues involved in the process of attempting to answer the question.

A *conceptual model* of a MANET NC situation needs to be created through *analysis and modelling* using a game theoretic approach. This model will be a mathematical or logical representation of the real problem. The model will be optimised to promote truthful cooperation.

A MANET containing a random number of selfish nodes will be *simulated* in a computer simulation program such as OPNET or Matlab. A network coding scenario will be created inside the network and the performance of the network will be measured with regard to the rules applied.

VI. Conclusion

In this paper, the idea of using game theory to solve the problems that independent rational nodes may cause is proposed. Background on MANETs, NC and game theory was provided. An example of a NC scenario was given and the effect of selfishness was illustrated in that environment. Future work includes the modelling of a MANET NC scenario using game theory as well as the optimisation of the model with regard to cooperation incentives. The model will be implemented in a MANET simulation and performance results will be acquired.

VII. acknowledgements

This work was completed with funding from the Telkom Centre of Excellence at the NWU, Potchefstroom Campus.

References

[1] P. Yang and B. Huang, "Multi-path routing protocol for mobile ad hoc network," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 4, dec. 2008, pp. 1024 –1027.

[2] Y. Yoo and D. P. Agrawal, "Why does it pay to be selfish in a manet?" *Wireless Communications, IEEE*, vol. 13, no. 6, pp. 87 –97, dec. 2006.

[3] S. Wang, X. Gao, and L. Zhuo, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, 2009, pp. 1 –5.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 497 –510, 2008.

[5] J. Price and T. Javidi, "Network coding games with unicast flows," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 7, pp. 1302 –1316, september 2008.

[6] V. Srivastava, J. Neel, A. Mackenzie, R. Menon, L. Dasilva, J. Hicks, J. Reed, and R. Gilles, "Using game theory to analyze wireless ad hoc networks," *Communications Surveys Tutorials, IEEE*, vol. 7, no. 4, pp. 46 – 56, quarter 2005.

[7] A. MacKenzie and S. Wicker, "Game theory and the design of self-configuring, adaptive wireless networks," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 126 –131, nov 2001.

Mario J. Engelbrecht is a Telkom CoE student pursuing a M.Eng degree. He received his B.Eng in Computer and Electronic Engineering from the North West University in 2011. His research focusses on optimisation using game theory and mobile ad-hoc networks.

A.3 Development of an Energy and Geographic Aware Opportunistic Network Coding Scheme

Presented at:

Southern Africa Telecommunication Networks and Applications Conference (SATNAC 2012)

September 2012

Development of an Energy and Geographic Aware Opportunistic Network Coding Scheme

Mario J. Engelbrecht and Magdalena J. Grobler
TeleNet Research Group
School of Electrical, Electronic and Computer Engineering
North-West University, Potchefstroom, South Africa
Tel: +27 18 299 1961, Fax: +27 18 299 1977
Email: {20704313, leenta.grobler}@nwu.ac.za

***Abstract*—Mobile Ad-Hoc Networks and Wireless Sensor Networks are communication networks that consist of nodes with limited energy resources. Therefore, routing protocols implemented in such networks need to be aware of the energy levels in a network to enable them to route in an energy effective manner. Network Coding (NC) is a technique that optimises the throughput of a network. In this paper we investigate the effect of power failure of energy-constraint nodes on network throughput, and propose a NC scheme that not only increases the throughput of a network as opposed to traditional routing mechanisms, but also does so in an energy effective manner. The proposed scheme incorporates geo-routing into one of the leading NC schemes, CORE, and while it delivers more throughput than CORE, it also focusses on energy awareness resulting in extended lifespans of nodes in a network. The NC scheme is implemented and simulated in OMNET++ and the results show significant improvement in both network throughput and lifespan of nodes.**

***Index Terms*—Energy Awareness, Geo-Routing, Mobile Ad-Hoc Network, Network Coding, Opportunistic Routing, Wireless Sensor Network.**

I. Introduction

The evolution of communication networks has led us to an era where you cannot only perform surgery halfway across the world, but do so while being in the comfort of your own home. By eliminating the need for wires, wireless networks revolutionized communication networks by enabling nodes to communicate while being in a mobile state. The concept opened many doors to new applications and possibilities. Wireless Ad-Hoc Networks are networks formed by wireless nodes. They are self-established and do not depend on any infrastructure.

Mobile Ad-Hoc Networks (MANETs) and Wireless Sensor Networks (WSNs) are wireless communication networks that consist of nodes with limited energy resources [1], [2]. Nodes need energy to transmit, receive and process data. When a node is drained of all its energy resources, it cannot perform any functions and is of no use to the network, or in the case of user operated nodes, the operating client. It is imperative that nodes in such networks remain alive for as long as possible, especially in WSNs where each node is responsible for monitoring a specific area or item [3]. Therefore, routing protocols implemented in such networks need to be energy aware. Energy awareness in networks refers to the monitoring of the energy levels of nodes, or the calculation of the amount of energy required per transmission, and using the acquired data as deciding factors when performing routing.

Network Coding (NC) is a technique that optimises the throughput of a network by coding packets [4]. NC can be divided into two categories namely deterministic NC and opportunistic NC [5]. In deterministic NC (DNC), coding opportunities are determined beforehand, and nodes are then assigned encoding and decoding roles. DNC is therefore heavily dependant on the network topology. In opportunistic NC, nodes calculate the local coding opportunities using the available information acquired from neighbour nodes. Opportunistic routing is the preferred option when routing in MANETs, since the topologies in such networks are dynamic.

Opportunistic routing (OR) is a technique that allows routing without prior knowledge of the exact next-hop node; hence OR routes opportunistically as opposed to fixed-path routing protocols [6]. In OR, there is a small pause before each transmission. The pause can be exploited to include routing conditions to be considered before performing further routing.

Geo-Routing (GR) refers to routing based on the geographical coordinates of nodes [7].

In this paper we investigate the effect of energy-constrained nodes on a network, and propose a NC scheme that not only increases the throughput of a network as opposed to traditional routing mechanisms, but also does so in an energy effective manner. The proposed scheme incorporates GR into one of the leading NC schemes, CORE, and while it delivers more throughput as CORE, it also focusses on energy awareness resulting in extended life spans of nodes in a network. The NC scheme is implemented and simulated in OMNET++ and the results show significant improvement in prolonging the lifetime of a node.

The remainder of this paper is arranged as follows: In section II, we give background on MANETs, WSNs, OR, GR and NC. In section III, we discuss related work, specifically COPE and CORE. Section IV presents the NC scheme after which the implementation is explained in section V. The simulation parameters are discussed in section VI, followed by the results and discussion thereof. A conclusion is drawn in section VII and we take a look at future work in section VIII.

II. Background

A. Mobile Ad-Hoc Networks (MANETs)

MANETs are self established, infrastructureless ad-hoc networks that consist of mobile nodes [8]. The term 'mobile node' refers to any mobile device with communication abilities such as a laptop or smartphone. Mobile devices are limited with regard to their transmission ranges, processing capabilities and energy resources [2]. The fact that nodes in MANETs have limited energy resources, necessitates custom routing protocols dedicated to preserve energy. MANET topologies can change randomly at any given moment due to the mobility of the nodes.

B. Wireless Sensor Networks (WSNs)

The primary use for WSNs is to retrieve real-time information from the environment such as temperature, geographical characteristics and particulate matter (e.g. CO_2). A WSN consists of sensor nodes and base stations (sinks). The sensor nodes are responsible for acquiring data from the environment, processing the data and transmitting information to the base stations [9]. The base stations integrate and combine the data collected from the sensor nodes to produce useful information. Sensor nodes are usually battery powered and it is therefore essential that WSNs operate and communicate in an energy-efficient manner in order to keep the nodes alive for as long as possible [1], [10]. Environment, medical healthcare and military monitoring are only a few of the fields included in WSN applications.

C. Opportunistic Routing (OR)

OR, unlike fixed path routing, does not specify each packet's next-hop node [6]. It calculates a forwarder set. The forwarder set of a packet contains nodes that are closer to the packet's destination according to a specific routing metric. Upon packet reception, each node in the forwarder set calculates a delay timer based on further criteria. Further criteria refer to conditions relevant to the objectives the routing protocol needs to accomplish. The objectives can be to increase robustness, optimise throughput or to conserve energy. The nodes with the better conditions will calculate a smaller delay and will therefore transmit sooner that the other nodes in the forwarder set.

When a node forwards a packet, the rest of the nodes in the forwarder set that overhears the packet will drop matching packets to prevent duplicate forwarding and unnecessary transmissions. OR is very effective in high packet-loss scenarios, since the probability of the entire forwarder set failing, is very slim. This incorporates a high level of robustness into the network [11].

D. Geo-Routing (GR)

GR refers to routing based on the geographical coordinates of nodes [7]. Nodes with a smaller geo-distance to the destination node, in comparison with the sender node, forward the packets. GR can only be implemented in networks where the nodes know their own coordinates. Nodes can acquire geographical coordinate information by GPS or localisation techniques.

E. Network Coding (NC)

NC refers to the concept of coding multiple packets into a single coded packet to optimise throughput [4]. This technique enables us to transmit multiple packets in a single transmission; hence transmitting more data for the same number of transmissions or requiring less transmissions to send the same amount of data, when compared to traditional routing mechanisms. NC increases network throughput to the theoretic maximum achievable throughput as can be calculated using the min-cut max-flow theorem [12], [13].

The theorem states that the maximum throughput achievable in a network, is equal to the minimum cut between the source and the destination nodes in a network, where there are no flow capabilities between the source and the destination. The theorem is explained in figure 1. In this scenario, source node S wants to send packets **a** and **b** to destination nodes X and Y. The minimum cut between source node S and destination nodes X and Y, is **two**. Therefore the maximum throughput obtainable in this network is **two**. After a full round of transmissions, **three** packets reached the destination nodes. **Two** packets were sent. This translates to a suboptimal throughput of **1.5** (3/2).

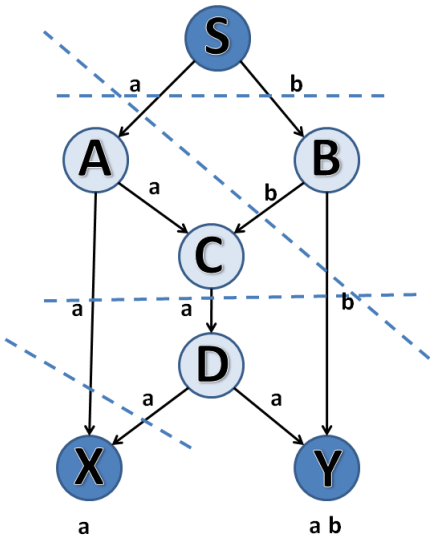

Fig. 1. Min-cut max-flow theorem

When NC is implemented, the throughput increases to **two**, as shown in figure 2. In this scenario, packets **a** and **b** are encoded at node C. The NC packet is sent in a single transmission to node D, and delivered to the destination nodes X and Y in the next transmission. The destination nodes then use the packets they have overheard from previous transmissions in order to decode the NC packet. In this scenario, **four** packets have reached the destination nodes, and it translates to a throughput of **two** (4/2), which is the maximum obtainable

throughput of the network. The following example explains the concept in more detail.

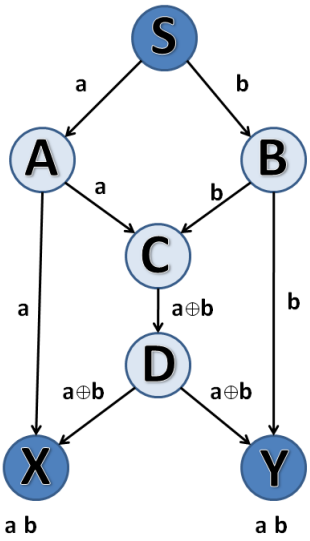

Fig. 2. Network with Network Coding

A single NC packet is created by means of XORing individual packets with each other. A NC packet is decoded by being XORed with a sufficient number of different individual packets it's comprised of. Please refer to figure 3.

Packet (1) and (2) respectively represents packets **a** and **b** from the previous example, depicted in figure 2. The NC packet comprised of the two packets are represented by packet (3). By XORing either of packets (1) or (2) with the NC packet (3) will decode the NC packet and deliver the other packet as illustrated in (4) and (5).

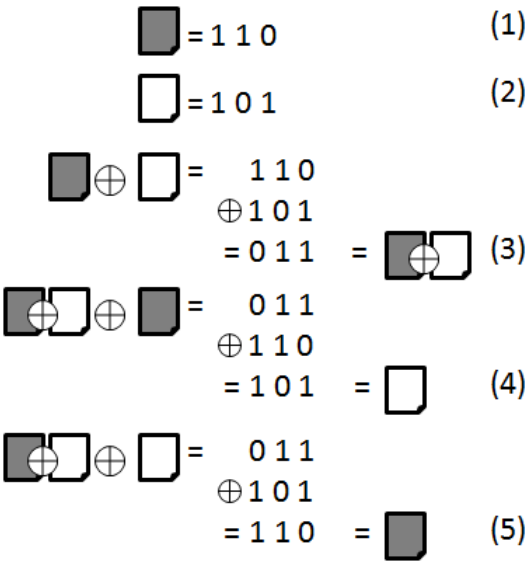

Fig. 3. Network Coding in detail

F. Opportunistic Network Coding (ONC)

In ONC, NC opportunities are calculated dynamically at each forwarder node by using the acquired neighbour information [4], [14]. This type of coding scheme is very effective in MANETs, since the network topology of such networks are dynamic. COPE and CORE are examples of ONC.

III. Related Work

A. COPE

COPE, presented in [4], is the first practical implementation of ONC. The NC scheme proves to deliver significant throughput gain as opposed to traditional routing schemes, in some cases even up to 400% increase. COPE employs three key elements namely *opportunistic listening*, *opportunistic coding* and *learning neighbour state*. The three elements work as follows.

Opportunistic listening - In wireless networks, any node close enough to a node that transmits, may overhear the transmission. This characteristic is exploited by enabling nodes to store and process those overheard packets. Nodes can then use the overheard packets for various purposes. One of those purposes is to decode coded packets. Each node is configured to temporarily store all overheard packets for a default time of T = 0.5s.

Opportunistic coding - Opportunistic coding refers to the intelligence in coding as many packets as possible, so that all the destination nodes can decode the coded packet with their stored packets, in order to retrieve the native packet intended for them. In order to discover and perform the best possible packet combination in opportunistic coding, it is essential that the NC node knows which packets each of its neighbour nodes has.

Learning neighbour state - Reception reports are appended to data packets. The reports contain information regarding the stored packets at the sender node. In this way, nodes can learn what packets are stored at their neighbours and are enabled to perform optimal coding accordingly.

Please refer to figure 4 for an example on COPE. By means of reception reports, node A knows which packets are stored by its neighbour nodes. Using this information, node A can decide to XOR packets $P1$, $P2$ and $P3$. Now, all the neighbour nodes can successfully decode the coded packets and retrieve the native packets intended for them.

Whilst COPE is opportunistic with regard to performing NC, it still lacks opportunistic features with regard to the paths the packets travel. Although the packets are coded opportunistically, the paths that they travel remain fixed. The fixed-path routing mechanism creates a limitation in the process of calculating coding opportunities, since the calculating node may only code packets that can be decoded at the **fixed** next-hops. This allows scenarios where packets are sent on suboptimal paths. CORE alleviate this shortcoming by incorporating OR into the scheme.

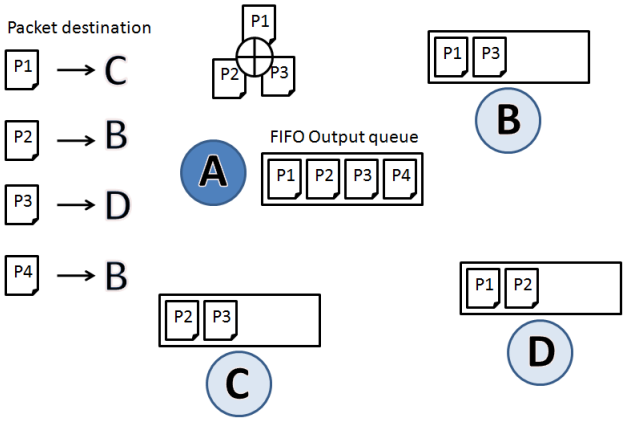

Fig. 4. COPE Network Coding

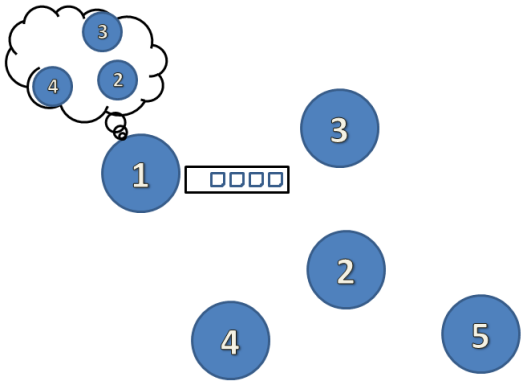

Fig. 5. Forwarder set

B. CORE

CORE, presented in [14], is based on COPE and shares the same three key elements namely *opportunistic listening*, *opportunistic coding* and *learning neighbour state*. Instead of routing packets on fixed paths as in COPE, CORE enables nodes to opportunistically forward packets without specifying the precise next-hop node. As in OR, CORE first selects a forwarder set. In this case, nodes have to be geographically closer to the destination node to be accepted into the forwarder set. The criteria prioritising the forwarder nodes are the number of local coding opportunities each node calculated. The nodes with more coding opportunities will have a smaller delay and therefore transmit packets sooner than nodes with less coding opportunities. Upon hearing forwarded packets, nodes in the forwarder set can cross-reference the sent packets with packets in their own queue and drop them accordingly to prevent duplicate forwarding.

Please refer to figure 5. In this scenario, node 1 has a packet for destination node 5. In COPE, only node 2 would be considered as the next-hop node and therefore only node 2 would be included in the coding opportunity calculation procedure. In CORE, node 1 calculates a forwarder set, in this case nodes 2, 3 and 4, which are all considered when calculating coding opportunities.

IV. Network Coding Scheme Design

We propose a geographic and energy aware NC scheme based on CORE [14]. The scheme adds geographic and energy conditions to the CORE scheme, and is made up of three key elements namely *forwarder set selection*, *priority based forwarding* and *coding opportunity calculation*. The design exploits the *priority based forwarding* aspect to include geographic and energy based conditions.

Forwarder set selection - If a node has a packet to send, it first selects a forwarder set. Nodes must comply with the following prerequisites to be included in the forwarder set:

- Must be a neighbour node
- Must be geographically closer to the destination
- Node may not exist in the packet's previous forwarder set

As in OR, the forwarder set is a list of nodes of which one will eventually become the forwarder of the packet. This set of nodes is the only nodes considered when calculating coding opportunities.

Coding opportunity calculation - When calculating coding opportunities, k number of enqueued packets are considered along with the first packet in the queue (the **primary** packet). The variable k can be adjusted to increase or reduce processing intensity. The deciding node knows what packets its neighbours have due to reception reports. It can therefore cross-reference different combinations of coded packets with the packets its neighbour nodes has and calculate the **code gain** of each combination. The **code gain** refers to the number of nodes that can decode the coded packet and retrieve a packet intended for them. The combination that delivers the most gains is referred to as the **maximum gain**. The packets are coded according to the combination that delivers the **maximum gain**.

Priority based forwarding - The **maximum gain** obtained in the previous step is one of the conditions used to calculate the delay the node must wait before it transmits. The delay is calculated by dividing a time interval, in this case 5ms, by the **maximum gain**. Nodes with more coding opportunities will have a smaller delay; hence transmit their packets sooner.

The delay value is now adjusted correspondingly by observing the energy levels of the nodes in the forwarder set. The adjustment is made to ensure that nodes with similar coding opportunities but with higher energy levels transmit sooner. The delay is adjusted further by taking into account this node's geographical distance to the destination node. Closer nodes will have smaller delays. When a forwarder set node transmits, the other nodes in the same forwarder set that overhear the packet will drop matching ones to prevent duplicate forwarding.

V. Network Model

The network model is discussed in this section. The network model is build in OMNET++. Each node is equipped with an

802.11b MAC layer, with a coding layer stacked on top. The rest of the upper layers are as they appear in the OSI stack. The upper layers are responsible for generating traffic. As for the *MAC layer* and *network layer*, they work as follows.

- *MAC layer* - The 802.11b standard is implemented in the MAC layer. It was modified to send all overheard packets to the network layer for processing. It was also modified to receive packet cancellation control messages from the network layer. The network layer sends cancellation messages to the MAC layer to indicate that the packet to be sent must be dropped.
- *Network layer* - The network layer performs all the coding and routing operations and is responsible for the following:

1 *forwarder set selection* - Calculating the forwarder set of the packet.
2 *opportunistic coding* - Calculating the local coding opportunities and knowing when to search for new opportunities on packet reception.
3 *priority based forwarding* - Calculating the delay timer based on the coding, geographic and energy based conditions.
4 Dropping packets from the queue in order to prevent duplicate forwarding in the forwarder set. This task is accomplished by sending down cancellation messages to the MAC layer.

VI. Simulation Results

Simulation parameters and results are discussed in this section. The NC scheme was simulated in OMNET++. One of the simulated networks is depicted in figure 6. Each node displays its battery status. The yellow warning sign indicates a dead node.

For simulation purposes, 200 nodes were deployed in an 800 X 800 m^2 square area. The nodes were placed at random coordinates inside the area. The channel rate was configured as 11 Mbps. The variable k was set to 10. UDP flows varying from 200kbps to 2Mbps was generated. The offered load was regulated by adjusting the number of flows. Five random topologies were used for each simulation setting.

The first simulation compares the throughput of CORE with and without the geographic condition. By incorporating the geographic condition, a throughput increase of as much as $1.3Mbps$ (23% throughput gain) can be achieved. The results are depicted in figure 7 and are summarised in table 1 with regard to percentage throughput increase, averages and maximum throughput.

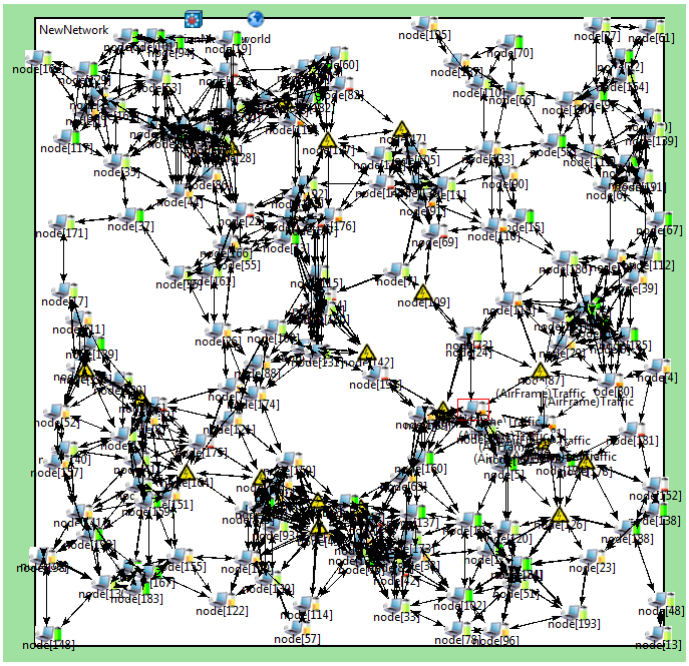

Fig. 6. OMNET network topology

Table 1. Throughput comparison of CORE with and without geographic based conditions

	CORE	**geo-CORE**		
Load (Mbps)	Through-put (Mbps)	Through-put (Mbps)	% Throughput increase	
2	1.7	1.8	5.88	
4	3.4	3.6	5.88	
6	4.3	4.9	13.95	
8	5.6	6	7.14	
10	5.7	7.02	23.16	
11	5.18	5.5	6.18	
12	4.8	5.4	12.50	
13	3.36	4.58	36.31	
8.25	4.255	4.85	13.88	**Average**
10	5.7	7.02	23.16	**Maximum throughput**

The second simulation evaluates the number of node deaths in time intervals. The nodes were given random low battery statuses to avoid unnecessary long simulation times. The energy levels assigned to the nodes allowed an average of five to 20 data transmissions per node. The incorporation of energy based conditions displayed significant improvement in node survivability. The results are depicted in figure 8.

VII. Conclusion

In this paper, one of the key elements in CORE, prioritized forwarding, is exploited to improve network throughput and create an energy aware, geo-aware NC scheme. The scheme combines the benefits of NC with geo-distance and energy

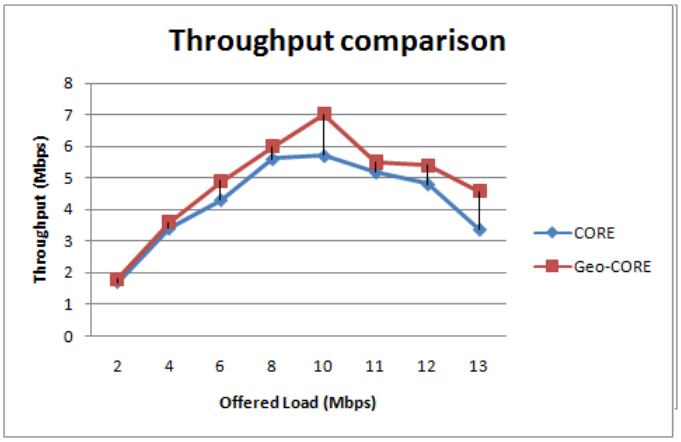

Fig. 7. Throughput comparison of CORE with and without geographic based conditions

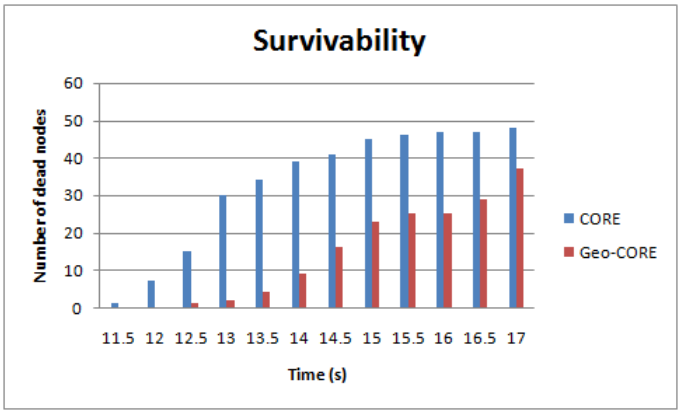

Fig. 8. Survivability of nodes when implementing CORE with and without energy based conditions

aware path routing to create an effective NC scheme for networks where energy efficiency and high throughput are imperative requirements. The simulation results showed a throughput gain of up to 23% and significant increase in node survivability.

VIII. Future Work

By changing the conditions of the *priority based forwarding* delay calculation, a routing protocol was enhanced with improved throughput and energy efficiency. By changing current conditions or even by introducing new ones, many more limitations and network characteristics can be addressed. Future work includes researching different combinations of current and new prioritising conditions for priority based forwarding to further improve the NC scheme, especially with regard to the mobility characteristics of the network.

IX. Acknowledgements

This work was completed with funding from the Telkom Centre of Excellence at the NWU, Potchefstroom Campus.

References

[1] S. Wang, X. Gao, and L. Zhuo, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, dec. 2009, pp. 1 –5.

[2] Y. Yoo and D. P. Agrawal, "Why does it pay to be selfish in a manet?" *Wireless Communications, IEEE*, vol. 13, no. 6, pp. 87 –97, dec. 2006.

[3] P. K. Sahoo and J.-P. Sheu, "Limited mobility coverage and connectivity maintenance protocols for wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 2856 – 2872, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611002167

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 497 –510, 2008.

[5] M. Grobler and A. Helberg, "Identifying opportunities for deterministic network coding in wireless mesh networks," *SAIEE African Research Journal*, vol. 102, pp. 8 – 15, 2011.

[6] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 69–74, January 2004. [Online]. Available: http://doi.acm.org/10.1145/972374.972387

[7] M. Zorzi and R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 337 – 348, 2003.

[8] P. Yang and B. Huang, "Multi-path routing protocol for mobile ad hoc network," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 4, dec. 2008, pp. 1024 –1027.

[9] V. Potdar, A. Sharif, and E. Chang, "Wireless sensor networks: A survey," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, may 2009, pp. 636 – 641.

[10] W. Li, J. Bao, and W. Shen, "Collaborative wireless sensor networks: A survey," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, oct. 2011, pp. 2614 –2619.

[11] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Computer Communication Review*, vol. 37, no. 4, pp. 169–180, OCT 2007.

[12] B. A. Forouzan, *Data Communications and Networking*, 4th ed. McGraw-Hill, 2007.

[13] S. Wang, X. Gao, and L. Zhuo, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, 2009, pp. 1 –5.

[14] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "Core: A coding-aware opportunistic routing mechanism for wireless mesh networks [accepted from open call]," *Wireless Communications, IEEE*, vol. 17, no. 3, pp. 96 –103, 2010.

Mario J. Engelbrecht is a Telkom CoE student pursuing a M.Eng degree. He received his B.Eng in Computer and Electronic Engineering from the North West University in 2010. His research focusses on optimisation of Network Coding schemes and Mobile Ad-Hoc Networks.