MSC ENGINEERING DISSERTATION

APPLICABILITY OF NETWORK CODING WITH LOCATION BASED ADDRESSING OVER A SIMPLIFIED VANET MODEL

Ashton Hudson

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, 2016

DECLARATION

I declare that this dissertation is my own unaided work, other than where specifically acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

Ashton Hudson

Date

ABSTRACT

The design and implementation of network coding into a location based addressing algorithm for VANET has been investigated. Theoretical analysis of the network coding algorithm has been done by using a simplified topology called the ladder topology. The theoretical models were shown to describe the way that network coding and standard location based addressing works over the VANET network. All tests were performed over simulation. Network coding was shown to improve performance by a factor of 1.5 to 2 times in both simulation and theoretical models. The theoretical models demonstrate a fundamental limit to how much network coding can improve performance by, and these were confirmed by the simulations. Network coding does have a susceptibility to interference, but the other benefits of the techniques are substantial despite this. Network coding demonstrates strong possibilities for future development for VANET protocols. The ladder topology is an important tool for future analysis.

ACKNOWLEDGEMENTS

The author would like to express gratitude towards Prof. Ling Cheng, with his advice, guidance and insight provided, this research was possible. Gratitude for the support from the Wits Centre of Excellence for Telecommunications Access and Services (CeTAS) that was provided. Furthermore, the author would like to extend appreciation to the National Research Foundation (NRF) for direct funding of these postgraduate studies, without this funding this would not have been possible.

CONTENTS

LIST OF FIGURES

NOMENCLATURE

Term	**Description**
G	Symbol used to denote a graph
V	Vertex used for the formulation of a graph
E	Edge used for the formulation of a graph
N	The number of nodes in a network, or if in a graph, so denote that it is an intermediate node
h	The symbol used for the Min-cut of a network
$S_{LBA}(N)$	The total number of transmission required to address a node in an LBA only network
$A_{LBA}(N)$	The average number of transmission needed to address any node in an LBA only network
$S_{NC}(N)$	The total number of transmission required to address a node in an LBA and network coding network
$A_{NC}(N)$	The average number of transmission needed to address any node in an LBA and network coding network
$S_{NC2}(N)$	The total number of transmissions for the scenario with two senders
$A_{NC2}(N)$	The average number of transmission for the scenario with two senders

Term	Description
$S_{NC3}(N)$	The total number of transmissions for the scenario with three senders
$A_{NC3}(N)$	The average number of transmission for the scenario with three senders
$S_{NC4}(N)$	The total number of transmissions for the scenario with four senders
$A_{NC4}(N)$	The average number of transmission for the scenario with four senders
P_N	Symbol used to denote this is the N^{th} Packet
$\oplus$	Binary xor operator
R_N	Symbol to denote Receiving node(s)
S_N	Symbol to denote Sending node(s)
e	Symbol denoting interference node

LIST OF PUBLICATIONS

1. A. Hudson and L. Cheng, "Application of Network Coding to Wireless Sensor Networks", work-in-progress paper presented at Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2014.

2. A. Hudson and L. Cheng, "Network Coding Applied to a VANET", presented at IEEE Seminar & 2nd Winter School on Information Theory and Communications 2015

3. A. Hudson and L. Cheng, "Network coding applied to location based addressing over a simplified VANET model", submitting to Mobile Networks and Application, Springer

.

Chapter 1

INTRODUCTION

1.1 Introduction

The amalgamation of network coding, location based addressing (LBA) and vehicle area networks (VANET) is a new, exciting and relatively unexplored field. With network coding, there is a paradigm shift from the routing schemes that have dominated the networking industry. Instead of having path discovery, the network coding technique generates packets together over a finite field and then broadcasts those encoded packets. The encoding and just the knowledge of the number of connections a network has on average is enough to obtain a near optimal throughput through the network; based on this network coding can use this to outperform traditional routing techniques. Network coding is still a new networking technique, but the benefits that it offers has created interest from the research community. The network coding technique offers benefits that would make it an intuitive replacement to existing traditional routing schemes.

Location based addressing is an ingenious way to decide how packets migrate through a network and reach their destination. With location based addressing, the use of an IP address or another fixed address is replaced with one that is specific to a geographical point, and an area around that point. LBA offers the benefit of not having to keep routing tables, but rather each network node knows where a packet needs to be sent to. The intermediate nodes decide whether to forward or drop a packet based on; their location, the destination node's location and the source node's location. This addressing scheme naturally uses multicasting, which blends well with network coding.

The third concept is the VANET, which is a network that provides information and services to road-based vehicles. With vehicles, their relative positions to other nodes, the pace at which nodes are moving and the structure of the network is dynamic. The VANET adds a lot of requirements onto the network because of this unpredictable and dynamic nature that the network nodes have. LBA can be a convenient solution to address and maintain a dynamic network since routing is not required and as a result path discovery is no longer required either. However, the overhead with LBA is that each node has to broadcast out a packet and this may be a problem with scalability. From here, it becomes a question as to whether network coding will actually improve the scalability of the network and improve overall performance.

1.2 Problem Statement and Research Question

The problem statement and research question both encapsulate the importance and the scope of applying network coding to location based addressing. Having correctly identified the problem that is needed to be addressed allows for the appropriate research question to be asked.

1.2.1 Problem Statement

With the increased interest in creating smarter vehicles and road networks, implementing a reliable and robust VANET has become of importance. Due to the dynamic nature of VANET, can traditional routing schemes be improved to manage and take advantage of this behaviour? Location Based Addressing is an existing technique that uses Geocasting and directed flooding to migrate packets through a network, with there being possible ways to improve the standard LBA algorithm. Coupling network coding with location based addressing is to be investigated to discover if a more scalable and reliable network routing algorithm can be introduced.

1.2.2 Research Question

The research question that is being answered in this dissertation is given below:

Can adding network coding improve the performance and scalability of location based addressing over a VANET?

1.3 Research Aims and Objectives

The research aims and objectives are based on attempting to establish the strengths of using LBA, and then discovering what improvements network coding adds when it is used. The simulation results that will be gathered will be compared against models for the specific scenarios decided on.

- Define a model for LBA and LBA with network coding.
- Implement the LBA routing protocol in a simulator.
- Implement a network coding in a protocol and couple it with LBA.
- Simulate these two different algorithms to find their strengths and weaknesses.
- To validate the work, the simulation results will be compared against the models.
- Create an idea of what applications these protocols fulfil.

1.4 Research Methodology

In order to carry out this research, a range of different scenarios will be tested, with varying network setups. Simulations will be carried out to test

the hypothesises, and these simulation results will be compared to theoretical models.
The first procedure is to create a baseline performance test that characterises the VANET that only uses LBA. Once a baseline performance for the VANET running with just LBA has been evaluated, the case where network coding is applied needs to be assessed.

With both results being obtained, then analysis can be carried out where the comparisons between the standard LBA and the LBA with network coding are looked at. In addition, the simulation results will be used to verify the theoretical models that will be proposed.. In order to carry out the simulations, a software simulator will be used, and the specific details of implementing network coding and the LBA algorithm will be programmed. All software developed will be held on a Git Repository for back-up and version control purposes.

1.5 Dissertation Structure

This dissertation is organised in the following structure. A literature review, seen in Chapter 2, is performed to familiarize with the topics that are being addressed. The literature review examines the themes and requirements of network coding, location based addressing and VANET to build a well-rounded knowledge base. Chapter 3 deals with the formalization on the techniques used, the proposed modelling technique of the Ladder topology, the definition of network coding, design work, the scenarios and testing. Chapter 4 provides the detailed derivations and proofs for the network models that describe the behaviour of the standard LBA algorithm, with the simulations results and analysis for this algorithm being included too. Chapter 5 provides the model derivations, proofs and theoretical limits of the gains network coding provides. This chapter then goes on to provide simulation results and based on these, analysis for LBA with network coding is given as well as the overall comparison between LBA with and without network coding. Chapter 6 brings the closing statements, future work possibilities, recommendations for research on these topics and finally the concluding statements.

Chapter 2

LITERATURE SURVEY

2.1 Wireless Sensor Networks

Wireless Sensor Networks (WSN) are becoming increasingly important in recent times, with their diverse range of applications and relative ease of deployment [1][2]. At the core of these networks is a collection of cooperating nodes which establish and maintain the network [1][2]. The nodes are required to also gather information through connected sensors, and then allow for this information to be routed through the network to specialised sink nodes [3][4]. The power of these networks is that they can collate a lot of information, and deliver it reliably through to the desired sink node, which will then forward that information out in the desired format [3][4]. Each node in a WSN is very simple, and is comprised of a power source, a sensor, some memory, a wireless transceiver and at the heart, a micro-controller, and this simple structure can be seen in Figure 2.1 [3]. Traditionally, WSN nodes focus on being very frugal on their power consumption, and this is to provide prolonged network life times. The simple design, power conservative components and extended sleep times are used to reduce battery consumption [2][3][5].

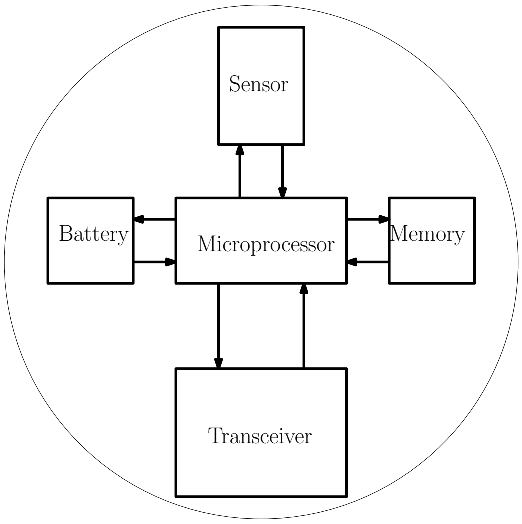

Figure 2.1: Fundamental components that make up a wireless sensor network node

2.1.1 Applications

Wireless Sensor Networks have a diverse range of applications that they can be applied to, and these applications can be broken down into two main groups; those that detect events and those that periodically measure sensed information [3][4][6]. For event detection, their main role is to alert authorities or concerned persons that something specific has happened. Such events include detecting forest fires, floods, earthquakes etc. The scope of event detection is not limited to natural disasters, and included car accidents or changes in a patients vitals.

Conversely, there are applications which require periodically measuring the environment and thus have their information aggregated. This aggregated information comes through to the sink nodes more predictably and regularly than event detection applications. These applications involve measuring the continuous water levels of a river, or the structural integrity of a building [3][4][6]. In the medical field, these applications could be the periodic measurement of a patient's vitals and condition, while in a vehicle-based network, this would be measuring the current number of vehicles on a road.

The range of applications is understandably overwhelming, and these can be seen below [3][4].

- Natural Disaster Detection (Forest Fires, Floods, Earthquakes).
- Infrastructure Integrity.
- Health Care.
- Sharing Information.
- Hazard Warnings.
- Traffic and Road Based Information and Warnings.
- Animal Tracking
- Intelligent Storage, Item Tracking and Stock Taking.

A lot of these applications have a very high coupling between event detection and periodic data flows. One such example is the Traffic and Road based information and warnings applications. For these applications, events such as a vehicle alerting traffic lights to change their colour to favour the driver, or a vehicle alerting roadside lights to turn on while the vehicle passes under them aligns more with event detection. While, a vehicle providing its location to the network periodically, allows for an aggregated view of the traffic flow through a road network. In these cases, a mixture of the two types of applications is required[3][4][6].

2.1.2 Types of Data Collection

There are three different ways to gather and process data in a WSN; these are *data aggregation*, *data collection* and *data selection* [7][8]. The information provided by each of these techniques varies and is dependent on the application that is gathering the information.

Data aggregation is a technique where the sensed data from each individual node is aggregated through the network and that information is provided to the sink node. An example of this is the minimum, average or maximum values from the nodes of the network [7][8]. *Data collection* is when all the data from each and every node is sent through to the sink node of the WSN. Finally, in *data selection*, the sink node indicates to the network what data it would like to be aggregated, as well as some more information about that data. In *data selection* the sink node could request the minimum, the mean, and the maximum, as well as some intermediate data points, allowing for better probabilistic or statistical understanding about the information collected.

2.2 MANET

Mobile Ad hoc Networks (MANET) are networks similar to Wireless Sensor Networks, where by they are self organising networks that allow for information to spread through the network [9–11]. MANET do deviate from WSN in that a MANET is a generalisation of the WSN, since a MANET allows for data communication between all the nodes. With MANETs, their main objective is to establish a stable network with substantially more dynamic nodes than a WSN, and this network must allow for a broader range of applications than a WSN [9][11][12]. One such vital application is establishing a communications network when traditional wired-networks have been brought down, such as in an earthquake [11][12].

2.2.1 Characteristics and Types of MANET

For a MANET to be considered ad-hoc, it needs to have mobile nodes which can dynamically connect to an established network at any point in time [12–14]. Due to this dynamic nature, and the fact that each node has a limited wireless range, nodes will use other nodes to forward their packets to the desired destinations. This means that every node in a MANET is required to act like a router too in order for the network to maintain connectivity [12–14].

The routing protocols that maintain MANET can thus be broken down into two main categories; Table-Driven routing protocols and On-Demand routing protocols [12–14]. In table-driven routing protocols, every node in the network keeps a table of how to forward packets through the network in order for connectivity to be maintained. On-Demand routing protocols will establish the route from the source to the destination as soon as it is requested, thus no node needs to keep a record of how to forward packets through the network.

There are two main types of MANET, open MANET(also know as heterogeneous MANET) and close MANET(also known as homogeneous MANET)[9]. With homogeneous MANET, the nodes that all connect together are the same device, this means that it is difficult to establish a wide range of applications on top of the network, since the devices providing the service must be the same. With heterogeneous MANET, the network is open to a wide range of different devices to connect to and means to connect to the network is more diverse. With heterogeneous MANET, you may have nodes connecting via different means, while with homogeneous MANET, the nodes will all connect with the same interface [9].

2.2.2 Routing with MANET

Routing in a MANET is a major challenge due to its dynamic nature [13][15], and traditional methods of routing applied to MANET show high failures in the links. MANETs are extremely diverse networks and the routing protocol that is being applied to them is highly dependent on how the MANET will behave [12][15][16]. There are two main types of routing protocols; proactive and reactive protocols [16]. With proactive protocols, the routing paths are predetermined and an algorithm discovers the network in a timely manner, updating the routing table. When a link is broken, the nodes further down the network need to have their routing tables updated and because of this proactive routing protocols suffer from tremendous overhead in dynamic networks. Substantial control messages need to be sent out to re-establish the network every time a link is broken and this is what cripples the network when nodes move too rapidly [16]. In low mobility environments, these proactive routing protocols perform well because the bandwidth needed to maintain the network is less than that used to transmit data.

Conversely, reactive protocols are considered 'on-demand' since when a data packet needs to be forwarded, a flood of control messages are sent and this is used to determine the best path [11][13][16][17]. These protocols are able to manage substantially more dynamic mobile networks than the proactive protocols since they are able to recover better from broken links. Despite their better performance in more dynamic networks, reactive protocols are not well suited for bandwidth demanding applications in slower changing networks because the control messages sent out are required consumes a large portion of the bandwidth.

The biggest issues with these protocols is that they are still ill suited for highly dynamic topologies and bandwidth demanding applications, such as applications where traffic incidence are required to be sent backwards [16]. While reactive protocols perform better than proactive protocols for reliability, they are not as effective as proactive for bandwidth utilization. This introduces a third type of routing protocol - one that is as bandwidth efficient as proactive routing, but also allows for dynamic routing like reactive routing. A proposed protocol is the Inter-Vehicle Geocast Protocol (IVG), where it uses location addressing to appropriately forward packets through a network condition [17][18][19]. Location based addressing is becoming a substantially more relevant method due to the accessibility to Global Positioning System (GPS) modules.

More complicated routing protocols have been proposed in [15], which use clever algorithms such as the ant-colony algorithm to find shortest paths and have the paths continuously refreshed. These algorithms offer better performance than algorithms like Ad hoc On-Demand Distance Vector (AODV) and (Optimized Link State Routing) OLSR [15], however, these algorithms do not take into account any packet based encoding.

2.2.3 Challenges with MANETs

MANET can be characterised by the three main topics; Dynamic Topology, Limited bandwidth, energy constrained and limited security [11][13][20][21]. These characteristics present significant challenges to the maintenance of the network, as well as the reliability of communication. MANET are designed to have highly mobile nodes, and these moving nodes force the network to be unpredictable, and this makes the topology rapidly change. Due to the unpredictable and rapidly changing topology, there are challenges in trying to effectively route packets through the network.

A major consideration are the bandwidth constraints on a mobile area networks being significant [13]. Wireless communications are substantially more susceptible to effects of noise, but also to effects of interference and fading, and these effects coupled with an unpredictable network topology create a very challenging network to model. The energy constraints of a MANET are application dependent, as some applications rely on depletive energy sources like batteries, and for these applications there are much tighter restrictions on energy consumption. However, if the nodes have regenerative sources of energy, then it can be expected that the nodes do not have the same limitations.

Security is a major concern for these networks [13][20], as these networks are open to being listened to by malicious nodes, and eavesdropping, denial of service and other malicious behaviour is a serious concern. The networks need to be built with security in mind in order to build a robust, maintainable, and well accessed network. However, the amount of security required is highly coupled to the application that the MANET is trying to provide, with military based applications requiring very high amounts of security[20].

2.3 VANETs

Vehicle Ad hoc Network (VANET) are a specialised form of MANET where the main application goal is to provide connectivity to vehicles and road networks. With the applications focused on transport, these networks are considered substantially more dynamic and mobile than most other MANET, which brings with it the problems seen with highly mobile MANET networks [16][22]. With VANET, the applicability of reactive and proactive routing protocols used for other MANET applications is brought to question. If the data being sent from the application layer is not much larger than the control messages, it makes sense to rather just flood the data throughout the network.

2.3.1 Divergence from MANET

MANET and VANET are very similar, however, one main property that separates the two networks from each other is that the nodes in a VANET move at higher average speeds and there are more nodes in a VANET [22][23]. With a VANET, there is an expected very high number of nodes, considering the density of vehicles and also the number of potential road-side nodes. Furthermore, the high speed of nodes, such as when they are passing static or passing traffic, the time that connections last for is extremely small. Confidentiality with messages is a lesser issue, given that many applications with a VANET require that the information is meant to be broadcast to all vehicles around it[24].

2.3.2 Routing within a VANET

There are a wide range of routing protocols available and many of them have been taken from MANET, however, reliability and applicability of these protocols is questionable [21][23][25].

With VANET, protocols such as AODV and DSR are applicable, but their communication throughput is very poor because of the dynamic nature of the network. As a result, there have been modifications applied to AODV where the protocols apply predictions. Prediction Based Ad hoc On-Demand Distance Vector (PRAODV) builds alternative routes instead of using one in a given time and PRAODV-M tries to predict the lifetime of a different routing paths.

VANET leads itself well to another type of routing - Geographical Routing. With this, a network node knows its location with the aid of GPS, and then the routing can be performed based on this information. There are two well known protocols, Greedy Perimeter Stateless Routing (GPSR) and Greedy Perimeter Coordinator Routing (GPCR) [21][23][25]. With these protocols, they perform well compared to reactive protocols like AODV and DSR. The use of geographical routing is a big shift from traditional methods.

Broadcast routing is a good method for relaying important information out through the network though, as it performs very well in densely packet scenarios. However, there are issues with flooding the network with packets because as the network grows larger, the performance of these routing protocols decreases rapidly [21][23][25].

2.3.3 Applications of VANET

The applications of a VANET based system are extremely diverse, and depending on the application a particular networking protocol will be better or worse suited. In Table 2.1 some applications are shown, along with the requirements on what that application needs to be implemented [22][26][27]. The range of applications that can be applied to VANET is not an easy list to exhaust, and as VANET becomes more popular, the range of applications is expected to increase too.

The major applications that are being considered are centred around providing information to drivers and the vehicles in order to improve the quality driver's experience and the safety for road vehicles [24][26][27].

Table 2.1: Applications, examples and the requirements to implement the application for VANET [24][26][27]

Application	Examples	Requirements
Road conditions	1. Change in road direction (hair pin bend, concealed drive ways)	can be slow, tolerates delay
	2. Wet, sleet or icy road	can be slow, tolerates delay
	3. Visibility; fog, low light	can be slow, tolerates delay
Incident	1. Pre-collision detection; sudden stopping, loss of control	must be fast, reliable, low latency
	2. Post-collision detection; broken down car, lane closure updates	can be slow, tolerates delay
Traffic conditions	1. Traffic flow conditions	can be slow, tolerates delays
	2. Abnormal vehicle	can be slow, tolerates delays
	3. Slow moving lane and lane closures	can be slow, tolerates delays
	4. Traffic control conditions; speed limits, maximum weight, special vehicle requirements	can be slow, tolerates delays
Vehicle updating road	1. Information traffic lights to control flow better	can be slow, tolerates delays
	2. Updating road illuminating lights to turn on and off	fast, does not tolerates delays
	3. Updating authorities of traffic flow	can be slow, tolerates delays

2.4 Location Based Addressing

Location based addressing (LBA) is a technique whereby routing decisions for packets travelling through a network are made based on the physical location of the network nodes[28][29]. This differs from traditional methods of routing where the shortest paths are determined first and then the packets are forwarded through those paths. With LBA, a sending node broadcasts out its packets and the nodes that overhear them will decide whether to forward them based on whether they are between the sending node and the destination location. Location based addressing is very similar to Geocast Routing [28–30].

2.4.1 Location Based Addressing Applications

The applications for location based addressing are mostly condensed around scenarios where the networks nodes have access to a GPS, but the network is also regularly dynamic. Location based addressing sees applications in the Urban and Highway traffic areas, where nodes are attached to vehicles and as a result, they are moving too fast for effective routing [28–30]. LBA via geocasting is an easy way for packets to migrate through a network reliably, and this allows for VANET applications that do not require specific node resolution.

The applications range from updating vehicles around a car about a hazardous activity; such as sudden braking. In this case, the vehicle suddenly braking will send warning packets backwards to alter on coming vehicles. LBA and geocasting are powerful here because they will only need to address the packet to the locations behind the node[29][31].

2.4.2 Location Based Addressing with VANET

With VANET, many of the nodes will be attached to vehicles and it can be reasonable to assume that they can have access to GPS or other location based services[29][31]. This is why LBA becomes a natural extension for

VANET since it can be difficult to route with a VANET, but LBA allows for a more dynamic routing through the network. The path is no longer required to be determined with LBA, and so if the network is highly dynamic, it does not impact LBA. LBA also opposes using the traditional routing schemes, as it does not look to discover the paths, and as a result, the routing protocols needs to be developed specifically for the application[29][31].

2.5 Network Coding

Network coding is a paradigm shift in the way that we carry data through a network [32][33], where it opposes the current and established methods of routing packets through a network. Network coding uses the way that nodes naturally overhear packets, especially in a wireless medium, and instead of having nodes decide whether to forward or drop the received packets based on a complicated routing table, nodes encode received packet together using the XOR process [32–34]. As outlined in many papers [32–35], network coding allows for the possibility of reaching the maximum throughput of a network topology, even outpacing the performance of optimal routing strategies. This performance is achieved by the ability for packets to be adequately mixed together and multiple senders to have their packets move down a network at the same time.

Network coding is not a silver bullet though, as it is notoriously difficult to implement [36], and does require a header that keeps track of the packet combinations. Furthermore, with simple Network coding, binary XOR is used to encode packets and this means there maybe a high probability of linearly dependent packets coming through the network. When two packets are linearly dependent, the packets that reside within them cannot be extracted. Linear Network coding is very similar to normal Network coding, except that the coefficients of the packets are over a linear finite field. Thus, the number of ways packets can be combined increased massively.

There are many examples of how Network coding working better than routing [32–34][35, 36] with topologies that have peculiar structures, such as the Butterfly Diagram. What Network coding truly promises, in all networks, is a reliable way to move information through a network. The rate at which Network coding can carry packets through a networks is directly proportional to the Min-Cut of the network. The Min-Cut is the minimum number of connections you need to cut to disconnect any node from the network between the senders and receivers. As the network's Min-Cut increases, the number of packets that can be linearly added, and then transmitted increases too on a one-to-one ratio. This has been proven by the Min-Cut Max-Flow theorem [35][37].

Since the foundation of Network coding relies on the linear combination of packets, the natural algorithm to find the packets is to use Gaussian Elimination to solve for the matrices. Each packet requires a header which keeps track of the coefficients that each encoded packet contains, this means that the header will have to be quite large, however, the coefficients range of values are limited over a fixed finite field. Since the senders are often random, the header will be a sparsely populated vector, and this means that it can be compressed efficiently - saving the overhead space that maybe a problem [35][36].

2.5.1 Applications of Network Coding

Network coding has a large potential to be applied to many different networking scenarios and even to situations that are not directly connected to networking but rather can take advantage of the technique. Decentralized networks and the way they are maintained can be improved, almost intuitively with network coding [38]. From the paper written by Christina Fragouli and Emina Soljanin [38], there are many different examples provided and these will be discussed subsequently.

With decentralized networks, distributing and disseminating information through

the network is a major challenge, as making sure you have received the packets correctly, and received enough packets to construct the data is a challenge. If there is a graph defined by: $G = (V, E)$, where V is the set of vertices and E is the set of edges defined by: $E \subset V \times V$. In this graph, there is a node $S \in V$ which wants to send to a node $E \in V$. If the network has a min-cut of h, then the maximum rate for the network will also be h by the Min-Cut Max-Flow theorem [35][36]. This means that routing through the network, would require discovering h disjoint paths, but once these paths have been discovered then the nodes would know where to forward packets to. However, if the topology changes, then the network would have to find a new path between the nodes S and R.
However, with random network coding being applied, the nodes between S and R would simply just have to encode the packets in the exact same way - each node simple encodes the packets randomly, and then each node just transmits out to every other node it is connected to. By doing this, the network is able to transmit at the min-cut rate. This extends to the fact that the min-cut rate is maintained in network coding even in random networks, and randomly changing network, as long as the min-cut is held constant. Routing fails to offer this level of robustness, and does not lend itself quite as well to handling the reliable transmission of information at random network coding.

From the work provided in [38], it can be seen that network coding can be a powerful tool in a network which has a randomized structure, and offers performance at the min-cut rate. This is a better and more reliable method than traditional routing as it no longer requires that nodes need to discover the path, but rather nodes can just broadcast random packets and the information is still reliably transmitted. Furthermore, for random networks, it has been said that with well-behaved random network graphs, the min-cut values tend to keep around a reasonable average value with a high probability.

Content distribution is the process where information, data, media etc. is delivered to a large number of users in a network. With traditional networks, large scale servers are used to provide the content, but with the advent of peer-2-peer protocols, the networks are now more scalable as the nodes that are downloading from the network are required to upload and maintain it too. However, peer-2-peer networks suffer from the problem of files becoming rare in the network, network coding allows for this problem to be mitigated efficiently [38]. Network coding allows for data to disseminate through a network

Wireless networks are another area where the application of network coding offers an alternative to the current routing methods. Wireless networks differ from the wireline networks in that they share the medium of communication and that the channels vary over time [38]. Wireless networks have the intrinsic ability to offer broadcasting as a means to sends messages out, where as with wired communications, this is not possible or feasible. Few wireless protocols do not exploit the advantages that wireless networks offer due to their designs focussing on simplicity and scalability [38]. Network coding naturally offers a means to communicate through broadcasts, and offers savings over physical resources.

With wireless networks, WSN and Ad-hoc networks are the first to be expected to start exploiting network coding because they have have much more flexible protocols. Network coding offers gains over efficiency, improvements over delay as well as adaptability to a dynamically changing network and this is due to the fact that resources are not wasted in deciding where packets must go. According to [38], network coding in a network scenario where there is no network knowledge achieves the same performance as routing that has perfect knowledge. From this, it is clear that network coding is a more intuitive method in unpredictable networks.

Multiple unicast sessions are another benefit offered by network coding. In these scenarios, there are multiple senders sending to multiple receivers, and routing traditionally handles this by buffering or queuing packets, and then forwarding them when the network permits. With network coding, packets are encoded on the fly and multicast out as they are required; this means that packets do not need to be queued extensively. Furthermore, network coding allows for the packets to migrate through the network at the same time and this allows for scenarios where queues grow very large to be mitigated. The benefits for network coding in multiple unicast sessions can be thought to extend to the world of WSN and MANET since it is very likely that multiple nodes are going to be sending messages at the same time, and it can be difficult to manage the queues [38].

2.5.2 Network Coding over WSN

Network coding has been applied to many different WSN scenarios, and a lot of different algorithms have been proposed that apply network coding in different ways, all with varying performances. Some of the most prominent algorithms developed are AdapCode [39], CoZi [40] and R-Code [41]. With AdapCode, linear network coding is employed, and the focus of the algorithm is to provide reliable data dissemination through a network and compete with the current algorithm called Deluge. With AdapCode, the simulation results show 40% less packets being sent that the Deluge algorithm, as well as the algorithm has been adapted to allow for low memory usage; since it needs to run on microprocessors.

CoZi is another network coding based dissemination algorithm, but it was applied to the ZigBee protocols. In this coding algorithm, it shows substantially faster throughput throughout the network in various different node densities. The latencies that are produced by this algorithm is substantially better than the regular ZigBee methods. Finally, R-Code is an adaptation of AdapCode, and it sees even better performance, with less overhead and much shorter delays than AdapCode.

2.5.3 Network Coding over VANET

VANETs are similar to WSN as discussed earlier, but given what is known about network coding and the conditions that the VANET will be subject to the networking protocols, some of the protocols discussed above become a little limited. VANETCODE is a network coding based algorithm that is designed for the dissemination of packets through a network [42]. This coding strategy focuses on the spreading of information throughout the network, and not necessarily on the point-to-point communication in the network. However, VANETCODE offers very strong gains over regular routing in terms of latency and the reduction of collisions.

Other network coding strategies that have been applied to VANET focus on using network coding to improve the downloading rates through the network to inform nodes in the network [43]. The main focus of research that has been done is viewing whether using network coding to disseminate information is better than current methods, and the evidence does show this.

2.6 Simulation Software

A major component of this research is validating whether or not the algorithms developed work, and to do this validation, a simulation software is required. There are a deluge of different simulation software available for VANET, WSN, and wireless networks [44]. Castalia was decided as the simulation framework that will be used because of its excellent documentation, its community and support, it is built on top of OMNeT++ and because of both Castalia and OMNeT++ having licences that allow for the free use of the software for academic purposes.

Castalia has strong support for mobile nodes, and implements a well designed layered structure, making the development inside specific layers relatively easy to set up and work on. Castalia also has a wide range of implemented protocols in the different layers, allowing for rapid development. Furthermore, with a little extra work, location information was made available to the routing layer.

Chapter 3

FORMALIZATION AND TECHNICAL INFORAMTION

3.1 VANET

VANET is one of the newer concepts for mobile networks, and with vehicles slowly becoming more advanced in recent times. VANET has the added benefit over traditional MANET in that often the vehicles will add more functionality that is not feasible in the other networks, and one of these features is having the vehicles location being constantly updated via GPS. The main assumption that can now be made is that all nodes in a VANET, especially more advanced VANET, will have location information available to the node and to its networking protocols.

With VANET, the range of applications is extremely diverse and the variety of these applications means that the routing protocol that is used to implement that application must be carefully chosen. For example, if accident or sudden incident information needs to be sent through the network, it is vital that the routing protocol can handle sudden spikes in the packets being sent. In this case, reliability and burst throughput are essential, but sustained throughput or continuous information is not important. However, if communications between vehicles involves continuous throughput, then a level of packet loss is tolerated, something such as a vehicle being updated about the road conditions, or a vehicle informing the road network of its location.

3.2 Simplified VANET Model Based on Ladder Topology

A challenge with developing these algorithms and networking protocols is knowing whether they are working correctly, and what the actual performance benefits are. In order to tackle this challenge, a model needs to be created that is able to give estimates on the performance of the network protocol, and this stands as the first line of validation. This section discusses the proposed model that will be used to create the estimated performance of the network.

The *Ladder Topology* is the proposed topology which will be used to simplify the modelling for analysis. Since LBA employs certain techniques that allow for a directed acyclic graph to be implemented, the ladder topology is simply a tool used to create the theoretical performance of the network. For the ladder topology to be formulated, the following assumptions need to be made:

- Node densities are relatively consistent.
- Nodes consistently have at *least* connection to two other nodes.
- The min-cut of the network is maintained at 2.
- All packets move directionally due to LBA being able to decide the forwarding direction.
- Packets can only move in one direction, but the directionality is only limited to that packet.

- Two packets can have different directions.
- The network needs to maintain a acyclic nature to be analysed.
- The network will be limited to only ‘two-lanes’.
- The network will have an undefined length.
- Thus, the network will be defined as $2 \times N$.

From these assumptions, the graph that is generated is $G = (V, E)$, with the vertices defined by equation 3.1, and the edges of the network defined by 3.2.

It is important to note that the network is two nodes wide, but has an undefined total number of nodes, denoted by N.

$$V = \{1, 2, \ ... \ , N\} \tag{3.1}$$

$$\begin{aligned} E = &\{\{1,3\},\{1,4\},\{2,3\},\{2,4\}, ... \\ &... \ \{N-3, N-1\},\{N-3, N\},\{N-2, N-1\},\{N-2, N\}\} \end{aligned} \tag{3.2}$$

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \tag{3.3}$$

This simplified model allows for the performance between scenarios with and without network coding used to be estimated. If the model and the simulation results agree, then the research is on the right track, if they do not agree then the model needs to be adapted, or the algorithm needs to be improved.

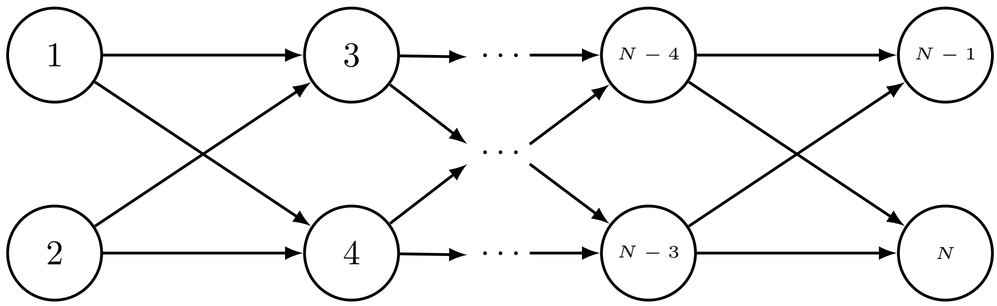

Figure 3.1: The graphical representation of the Equations (3.3) and (3.2). This image is also the generalization of the plots.

In Figure 3.1, the graphical representation of the ladder topology shown in matrix 3.3 and equation 3.2. From this figure, it can be seen why it is called the ladder topology as it resembles the structure and form of a ladder.

3.3 Location Based Addressing and Network Coding

LBA, as discussed in Chapter 2.4, opposes the traditional way to route packets through a network, since nodes intuitively know whether a packet needs to be forwarded, dropped, or can be passed to the application layer. Network coding, as discussed in Chapter 2.5, on the other hand is a method to efficiently allow packets to 'flow', such that the maximum rate for the network is achieved. With network coding, the topology of the network is less important than the min-cut and max-flow; such that if a min such is maintained, network coding allows for the maximum rate for the network.

Coupling network coding with LBA may not seem ideal at first, however like network coding, LBA relies on broadcasting and this allows for a perfect. Furthermore, LBA allows for network coding to efficiently group together packets, allowing for a 'train' of packets that can progress through the network - including a grouping of unicast packets. The combination of LBA to network coding becomes an intuitive combination since the two techniques focus on slightly different aspects of networking.

LBA focuses on ways to address nodes, but also offer a way for nodes to know more about the network than they would otherwise; since they have access to extra information. With LBA the routing decisions are now based on known information about the network, not necessarily about information other nodes have said about the network. Furthermore, LBA can be used to create a directional network - something that is not possible with naive broadcasting/multicasting. Network coding on the other hand does not care about the direct addressing used in a network, nor does it care about the directionality of the network.

3.3.1 Network Coding

Network coding is the technique that was proposed by Raymond Yeung which maximises the throughput of a network [35]. This work showed that scenarios seen in Figure 3.2 where the min-cut of the network is two, but routing cannot allow for the two packets P_1 and P_2 to progress through the network at the same time. However, if the intermediate node N, as seen in Figure 3.2, encodes both packets into a novel packet $P_1 \oplus P_2$, then the nodes R_1 and R_2 can both decode and receive both packets.

Figure 3.2 is often the first example shown to prove the power for network coding, as it offers a solution to a problem that cutter routing techniques simply cannot solve. Given the Max-Flow Min-Cut Theorem [37], min-cut is a way to determine how many packets, or signals can move down a fixed network path. This can be seen in Figure 3.3, where the min-cut of the network is two as the minimum number of connections that need to be severed to disconnect any node from the network is two. By the Max-Flow Min-Cut Theorem, this implies that two packets can move through the network simultaneously, without having to reduce the packet rate of the network.

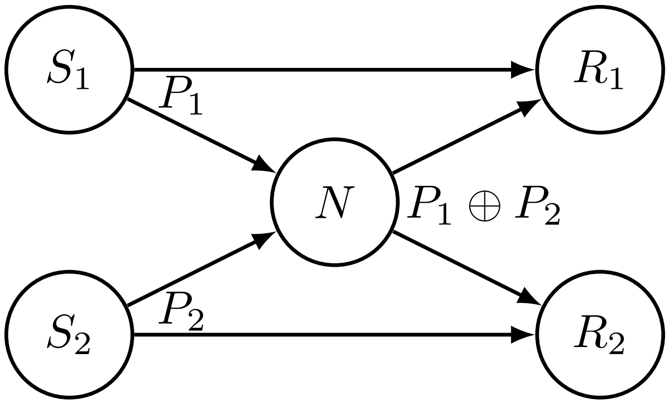

Figure 3.2: Butterfly diagram showing how network coding allows for packets to move more efficiently through a network by allowing them together at intermediate nodes.

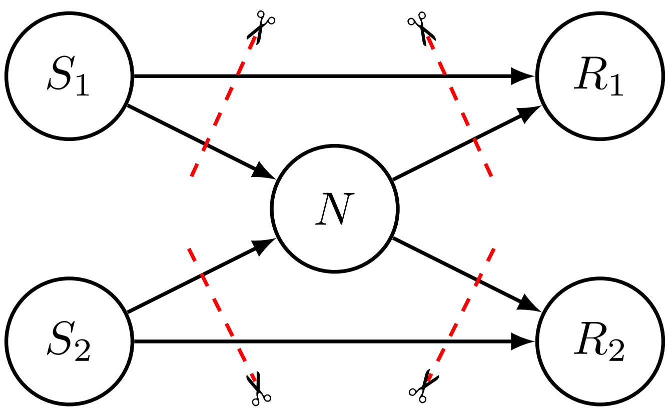

Figure 3.3: The Min-Cut of a network is defined as the minimum number of connections that are required to completely disconnect a node from a network. In this figure, the min-cut is two.

Network coding derives from the fact that there is a discrepancy between the Max-Flow Min-Cut Theorem and scenarios such as the butterfly plot shown in Figure 3.2. Routing simply does not allow for the rate of flow to reach the theoretical maximum.

The Max-Flow Min-Cut Theorem is as follows[45] :

Theorem 1. *For a graph defined as $G = (V, E)$ with unit capacity edges, a sending vertex defined as S and a receiving vertex R. If the minimum cuts required to disconnect R from S is h such that h is the min-cut, then the maximum rate that data can move between S and R is h.*

From Theorem 1, which is a unicast scenario, the cases with which multicasting is used can be considered. In this, a graph, G, is an acyclic directed graph with unit capacity edges. The min-cut of the network is maintained at h. There are h senders $S_1, \ldots, S_h$ sending to N nodes $R_1, \ldots, R_N$ [36]. For this, a unit capacity edge is defined as a symbol from a finite field, $\mathbb{F}_q$ of size q, being sent down a transmission in one time unit [36]. With this in mind, the theorem that defines and formalizes network coding can be discussed[46]:

Theorem 2. *For an acyclic directed graph defined as $G = (V, E)$ that has unit capacity edges, has h unit rate sources that are all on the same vertex of the graph and there are N receivers. It is then assumed that the min-cut of each receiver is also h. Then in this scenario, there exists a way for the intermediate nodes to linearly combine the transmitted packets over a finite field $\mathbb{F}_q$ that transmits the information sent from the sources at a rate equal to h.*

Theorem 2 can be said that given there are multiple packets being sent through a network, there does exist a way to encode packets by linearly combining them, using finite field, at intermediate nodes which runs at a rate equal to the minimum cut of the receiving node. From this proof, the multicast scenarios for the ladder topology can be defined as seen in Figure 3.4. In this scenario, the min-cut rate h for each receiver is 2, and the intermediate nodes N_1 and N_2 encoding the sent packets P_1 and P_2 to allow for the maximum rate of 2 to be maintained. The intermediate nodes acknowledge that the min-cut is two and know that they only need to encode the two packets together and transmit only one randomly encoded packet in order for the receivers to correctly receive the packets.

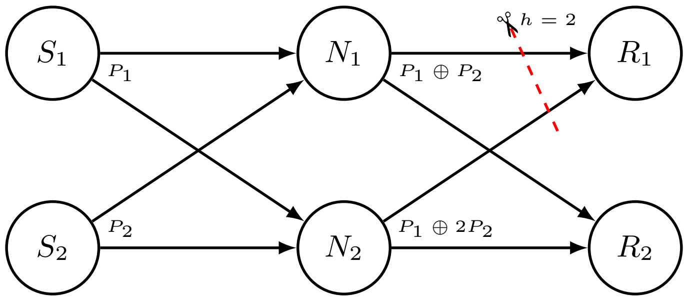

Figure 3.4: Network coding example over the ladder topology, showing the min-cut between nodes being two, and thus a maximum encoding rate of $h = 2$ being maintained. The encoding rate is also being demonstrated, and both receivers R_1 and R_2 are receiving packets P_1 and P_2 simultaneously.

From this, the operation of network coding over the ladder topology has been defined. In cases when there is only one sending node, there are no advantages offered. However, as more senders are sending packets, the opportunity to implement network coding increases and this can be seen from Theorem 2 and Figure 3.4.

3.3.2 Formalisation of Location Based Addressing

Location based addressing is a method of addressing nodes not by static or dynamically set addresses, such as an IP address, but rather, a node's address is the exact area that it is currently occupying at an instant. When a node wishes to send a packet, it sends that packet to a specific location with the knowledge that that area would be interested in what the packet contains. To further generalise the location addressing, the area around a specific point is regarded as an acceptable place for a packet to be addressed to. This generalisation allows for LBA to effectively multicast packets to whole areas without having to specify individual nodes, or have to work out what the subnet is. Having an effective multicasting method that uses location and area is far more in tune with how VANET work as information needs to be conveyed to zones of relevance rather than to specific nodes.

It becomes clear that with LBA, the network loses resolution to specific nodes. LBA does gain the advantage of being able to convey information to areas, zones, or specific points without having to decide how to get that information there. This lowers the overhead required to route packets through an ever changing network since simple decisions can be made by intermediate nodes on whether to forward or drop a packet. This advantage blends itself well to applications that are in highly dynamic networks where nodes are rapidly changing their position and routing is not viable. Applications where information needs to be sent to places relative to the sending node, such as accident or incident information, are also well suited to LBA.

LBA does have limitations though, as it is very difficult to target specific nodes for communication. In these situations the sending node will need to know exactly where the destination node will be when the packet arrives. This is less of a problem if the destination node is a fixed node, perhaps a relay node on the road side, but communication between two moving vehicles is extremely difficult. LBA is also not well suited if there are strict power limitations of the transceivers as they must always be listening and/or broadcasting. LBA is also not well suited to be included into many existing routing protocols as it is a different paradigm to how those protocols address nodes.

In Figure 3.5, an example of how LBA works has been presented. In this example, the nodes 4, 5 and 6 all fall under the area that the sending node, S, specified. In this case the nodes 4, 5 and 6 will send the packet up through to the application layer. Nodes 7 and 8 have been excluded from the network as decided by the nodes 3 and 5 respectively, as neither 3 or 5 multicast out the packet. The dashed lines between the nodes shows how they are within connection distance. This shows the crux of how LBA works, as each node makes a decision on whether to forward a packet or not, which effectively excludes parts of the network. With LBA, every node can over hear a packet and it is that node's responsibility to keep broadcasting out packets.

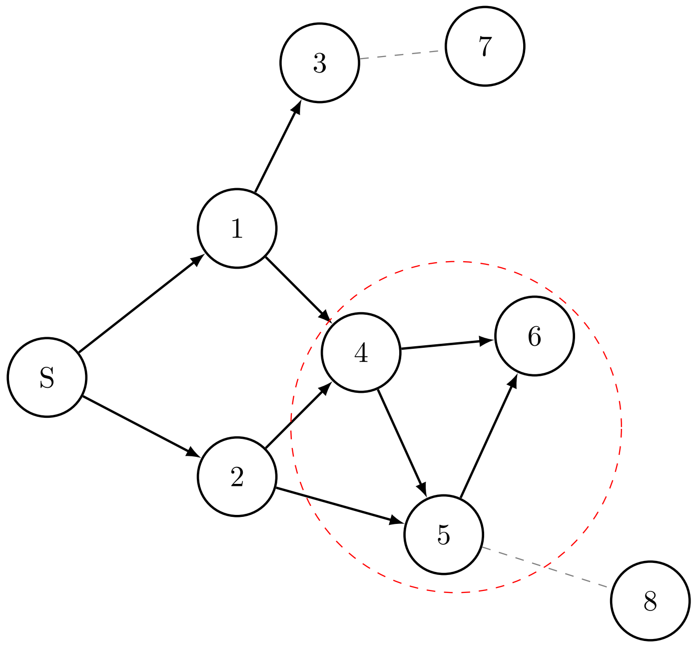

Figure 3.5: Graph showing how nodes 4, 5 and 6 are being addressed because they fall under the area specified by the red circle. Nodes 7 and 8 are excluded from this network as the nodes 3 and 5 have no reason to send to those nodes.

3.3.3 Formalisation of Location Based addressing with Network Coding

Applying network coding to the location base addressing algorithm is a process of overlapping the two techniques into a simple and clean design. With network coding, as mentioned in Chapter 2.5.1, a graph is defined as $G = (V, E)$, with V the set of vertices and E the set of edges of the network and $E \subset V \times V$. Inside this graph, there exists a Min-Cut of h, and this Min-Cut is also the Max-Flow of a network.

Merging network coding with LBA affords certain luxuries to network coding that improves the algorithms performance. With network coding, it is a difficult task to encode packets together if the addresses for the destination nodes are different. If different source nodes are sending packets to the same location, there is no problem since the destination node's address does not have to be encoded and all intermediate nodes can simply look at this plain text address and make a decision based on that. This changes when different destinations are to be encoded together. One way to solve this is to encode all the addresses together and have intermediate nodes decode on the fly, this is not very efficient though as continuous decoding and encoding is required. An alternative is to just append the destination's address to the encoded packets, but this may not be a scalable solution since the packets size will slowly start increasing as more packets with different destinations get encoded together.

LBA offers the ability for entire areas to be addressed, and furthermore, if specific locations within that area need to be addressed, the general destination of the node can be what all intermediate nodes look at when forwarding the packets. This allows for a little extra overhead to be added, but an unlimited number of nodes can be encoded together. LBA affords this flexibility and scalability, which is a slight change to the network coding algorithm.

3.4 Design of the Routing Layer

In order to test the feasibility of the combination of LBA and network coding, a new networking protocol would have to be developed. The routing layer is the main focus for this, with both the Application layer and the MAC layer being treated as agnostic, thus allowing for simple applications to be built on top of the LBA and network coding protocol, and allowing for a pre-existing MAC layer to be used.

3.4.1 MAC layer used

For its availability, the IEEE 802.15.4 is the MAC and physical layer being used. It is a standard that is available in the routing software, and because of its popularity with many wireless protocols, it was decided as a stable layer to build on. Some considerations that need to be made are that if broadcasting is used, then the protocol by-passes its own congestion control. Since all congestion control now needs to be handled with the routing layer - the network coding algorithm gets to control when to send, decode and how long to wait. This is ideal in many ways since with network coding requiring time for the buffer to fill up, this can be implemented without the MAC layer trying to implement congestion control.

3.4.2 Design of LBA Routing Algorithm

In Listing 3.1, the pseudocode for LBA decision making has been included. The pseudocode just shows how a packet from the MAC layer is handled, with the function *forward()* simply sending the packet out again. The function *drop()* drops the packet as it can not be forwarded and the packet is not destined to this node. Finally, the function *pushToApplicationLayer()* sends the packet up to the application layer since the packet is meant for this node.

3.4.3 Design of LBA Routing Algorithm that uses Network Coding

To implement network coding, buffering is essential, since there needs to be a small amount of time to wait for packets to come in so that the messages can be solved. Furthermore, intermediate nodes will be able to see if the packet is meant for them, so they will create a buffer than continuously encodes packets together. This increases the complexity of the system substantially from just the LBA pseudocode seen in Listing 3.1.

The network coding algorithm comprises of three components; a route decision making algorithm, an outBuffer and a packet solver. The route decision

Listing 3.1: Decision making pseudocode for location based addressing algorithm

```

void routingDecision(Packet packet)
{
    if( this.location < packet.source.location && ...
        this.location > packet.destination.location )
    {
        forward(packet);
    }
    else if( this.location > packet.source.location && ...
        this.location < packet.destination.location )
    {
        forward(packet);
    }
    else if( this.location < packet.source.location && ...
        this.location < packet.destination.location )
    {
        drop(packet);
    }
    else if( this.location > packet.source.location && ...
        this.location > packet.destination.location )
    {
        drop(packet);
    }
    else if( this.location = packet.destination.location )
    {
        pushToApplicationLayer(packet);
    }
}
```

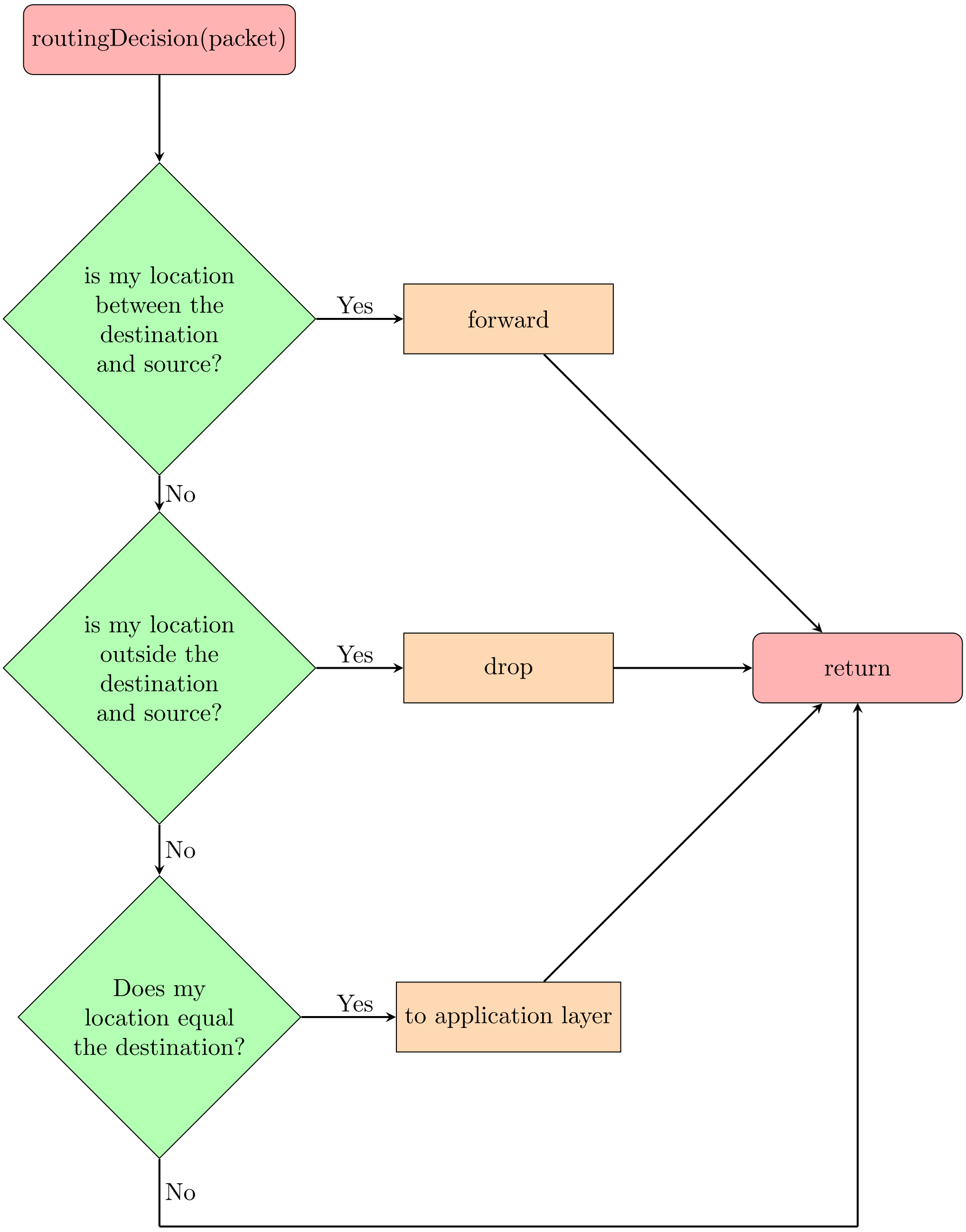

Figure 3.6: Flowchart demonstrating the algorithm flow for the routing decisions made when LBA is used. This flowchart captures the simplified process detailed from the pseudocode shown in Listing 3.1.

Listing 3.2: Decision making pseudocode for location based addressing algorithm with network coding

```

void routingDecision(Packet packet)
{
    if( this.location < packet.source.location && ...
        this.location > packet.destination.location )
    {
        sendToObuffer(packet);
    }
    else if( this.location > packet.source.location && ...
        this.location < packet.destination.location )
    {
        sendToObuffer(packet);
    }
    else if( this.location < packet.source.location && ...
        this.location < packet.destination.location )
    {
        drop(packet);
    }
    else if( this.location > packet.source.location && ...
        this.location > packet.destination.location )
    {
        drop(packet);
    }
    else if( this.location = packet.destination.location )
    {
        sendToPacketSolver(packet);
    }
}
```

making algorithm is the same as the LBA decision making algorithm is shown in Listing 3.2.

For the *outBuffer* component, all packets are added into a buffer. As packets are being added, the general destination value is calculated and stored. When the node triggers that it is time to forward the packets, it then requests packets from the *outBuffer*. The *outBuffer* code simply multiplies all the packets inside the buffer by randomly generated numbers and this creates the encoded packet. The general destination is added to the packet, and then the packet is sent to the MAC layer. Once enough packets have been sent, Listing 3.3: Simplified outBuffer pseudocode

```
addPacket(Packet packet)
{
    vector.push(packet);
    updateGeneralDestination(packet.getDestination());
}
getPacket()
{
    packet = new Packet();

    for(i = 0; i ==vector.size()-1; i++)
    {
        rand = getRandomNumber();
        multipliedPacket = ...
            FiniteFieldMultiplication(vector[i],rand);
        packet = XOR(packet,multipliedPacket);
    }

    packet.GeneralDestination = GenDest;
    return packet;

}
flush()
{
    vector.empty();
}
```

the *outBuffer* is 'flushed', clearing it and allowing it to accept a new batch of packets. The simplified *outBuffer* code is show in Listing 3.3, with the basic operations being shown. The functions *flush()* and *getPacket()* functions are called depending on when a timer triggers. When enough time has elapsed after the first packet has been received, the timer starts to request for packets to send out. After further time has elapsed, and all the packets have been requested, the vectors are flushed and the buffer is now cleared.

The final component of the network coding algorithm is a packet solver. To reliably solve packets, the packet solver is the intermediate step between the routing layer's decision making algorithm and the application layer. In this case, the function *pushToApplicationLayer()* in Listing 3.1 is replaced with another function called *packetSovler(packet)*. The packet solving code involves having a buffer that just stores all packets destined to this node, and the packet solver simply measures how many packets it needs to decode all received packets and waits until it receives enough packets. Once a square matrix of packets has been populated, Gaussian Elimination is used to solve the packets. Then, packets are run passed the LBA route decision making algorithm again since some packets maybe destined to different nodes, but the general destination told this node to decode the packets. If a decoded packet is destined for this node, then it must be sent up to the application layer. This code can be seen in Listing 3.4, with the corresponding flowchart being shown in Figure 3.8.

3.4.4 Further design aspects

Further design aspects that need to be considered, but are not formally apart of the network coding or location based algorithms are buffers that are used before the MAC layer, and the way packets are handled. The MAC layer normally handles congestion control, however, since all packets are broadcast to circumvent the addresses used in the IEEE 802.15.4 protocol. When broadcasting is used, all congestion control is disabled, and this meant there were a lot of packet collisions. To get around this the packets were sent at random intervals. This is not as efficient as working with a contention based protocol, but implementing a new MAC layer is not within the scope of this research.

3.5 Scenarios and Testing

To test how whether network coding and location based addressing is working, the models are used as the baselines. Furthermore, network coding with location based addressing will be compared against the standard location based addressing algorithm. By pitting the two algorithms against each other, the strengths and weaknesses of the two strategies can be shown.

The main scenarios that will be testing the performance of these strategies are examining how they handle an increasing number simultaneously sending nodes (so that the networks are put under considerable strain), then how well they handle interference and the recovery time of interference.

From these scenarios, the throughput, stability, resistance to interference and load on the nodes can be inferred.

3.5.1 Scenarios

There are two major types of scenarios that are being investigated for location based address only and location based addressing with network coding; first how the networks handle an increasing number of simultaneously sending nodes with an increasing network size. Second, how each network handles an interfering node that moving towards and away from the network at vary distances. These scenarios test different aspects of the two routing strategies, with the first testing how scalable that networking strategy is for scenarios where there is a lot of traffic on the network. The second scenario tests how well the strategy is able to handle and recover when there is a large amount of varying interference on the network.

The first scenario that is investigated is the one where the size of the network is increased from two nodes to eighteen nodes. At first only one sending node is used, and then after each successful completed run of the scenario, another sending node is added, with the added nodes sending messages at the same time to each other. The simulation will end when the network is no long able to manage the number of packets being sent. With these scenarios, models can be generated that will give the ideal performance from the network - which provides a baseline for the simulations to be compared to.

The second major set of scenarios that will be tested are interference tests which investigates how well the two different routing strategies can recover when there is external interference on the network. For this scenario, the networks have 18 nodes in them, and there are between one and three simultaneously sending nodes. However, an interfering node is now introduced, and it moves towards the network, then turns back and moves away from it. The distance travelled by the interfering node is varied, so as to simulate an interference moving towards and away from the network. In this scenario, the interfering uses a throughput test for its application layer, a bypass routing layer and the MAC layer used is different from the other nodes.

In each of these scenarios, the nodes are 10 meters apart in the x-axis, and 15 meters away from each other in the y-axis, and they use an antenna with a fixed power output of -3dBm. The distance between the nodes and the transmitter power used was chosen to allow for the ladder topology to be effectively implemented and tested. If this were to be implemented, an adaptive power method could be implemented so that the ladder topology is consistently implemented. These details can be seen in Table 3.1 for all the nodes in the network, except for the interference node, while Table 3.2 shows the conditions that the interfering node will be under.

Table 3.1: The conditions of the nodes that will be in the main network for both just LBA as well as LBA with network coding.

Environment	**Value**
Number of Nodes	1-18
Transmission Power	-3dBm
MAC layer	802.15.4
Distance between nodes: x-axis	10m
Distance between nodes: y-axis	15m
Maximum network length	150m

3.5.2 Testing

The testing process for these scenarios is simply a task of running the different scenarios within the simulator and then recording the details from each simulation. The results from Castalia will be reflected against what the performance that the models generated predict, which should show some consistency. If there are discrepancies between the models and the simula-

Table 3.2: The conditions that the interfering node will be under for the second major scenario that will be tested.

Environment	**Value**
Transmission Power	-3dBm
MAC layer	TMAC
Starting position (meters)	(50,75), (100,75), (200,75), (300,75), (400,75), (500,75)
Ending position (meters)	(0,75)
Speed (m/s)	5

tion results, either the simulation has not been correctly implemented or the model requires revision.

Testing the two routing strategies that have been designed and implemented is paramount to determining how valid they are as viable routing protocols for a VANET. Thus, another part of testing is looking deeper into the data gathered and generating the strengths and weaknesses of each strategy. This will allow for the effective analysis and will help create conclusions about the applicability of using location based addressing with and without network coding.

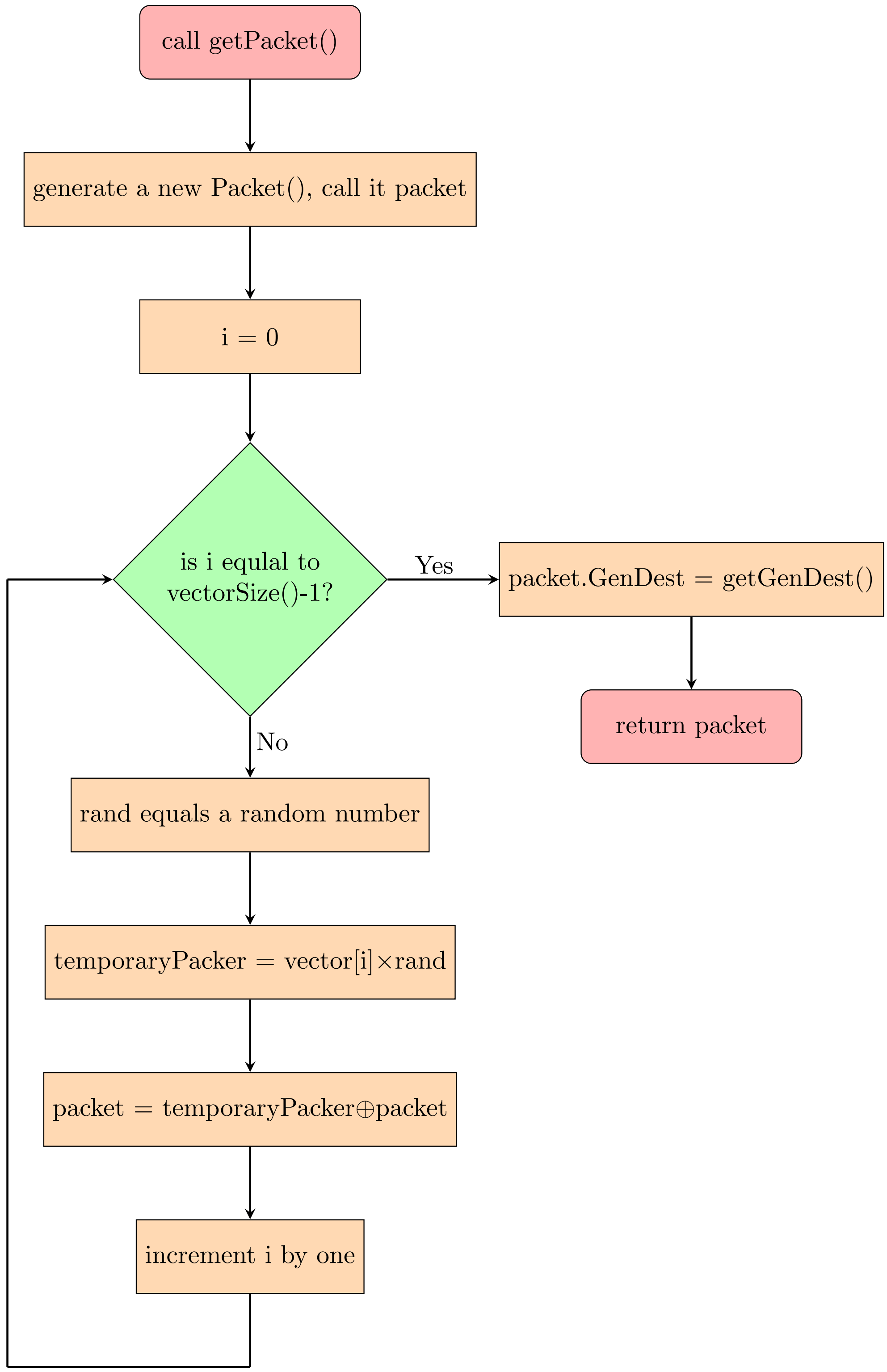

Figure 3.7: Flowchart of the *getPacket()* algorithm seen in Listing 3.3. The general flow of the algorithm is being shown.

Listing 3.4: Simplified packet solver pseudocode

```
addPacket(Packet packet)
{
    matrix.add(packet);
    if(matrix.isSquare())
    {
        solve();
    }
}
solve()
{
    solvedMatrix = Gaussian.solve(matrix);

    // If the matrix has linearly dependent components,
    // then it cannot be solved.
    // The Gaussian.solve() Function will return a
    // reduced matrix which will allow it to be
    // further populated.
    if(solvedMatrix! ==NULL)
    {
        matrix = solvedMatrix;
        return;                  // matrix cannot be solved,
                                 // so return and wait for
                                 // more packets.
    }

    for(i = 0; i = solvedMatrix.size()-1, i++)
    {
        Packet newPacket = solvedMatrix[i]; // Getting the
                                            // solved packet.
        if(this.location = newPacket.destination())
        {
            pushToApplicationLayer(newPacket);
            // packet can be sent to application layer now.
        }
        else
        {
            routingDecision(newPacket);
            // If the packet isn't destined for this node,
            // re-process it.
        }
    }
}
```

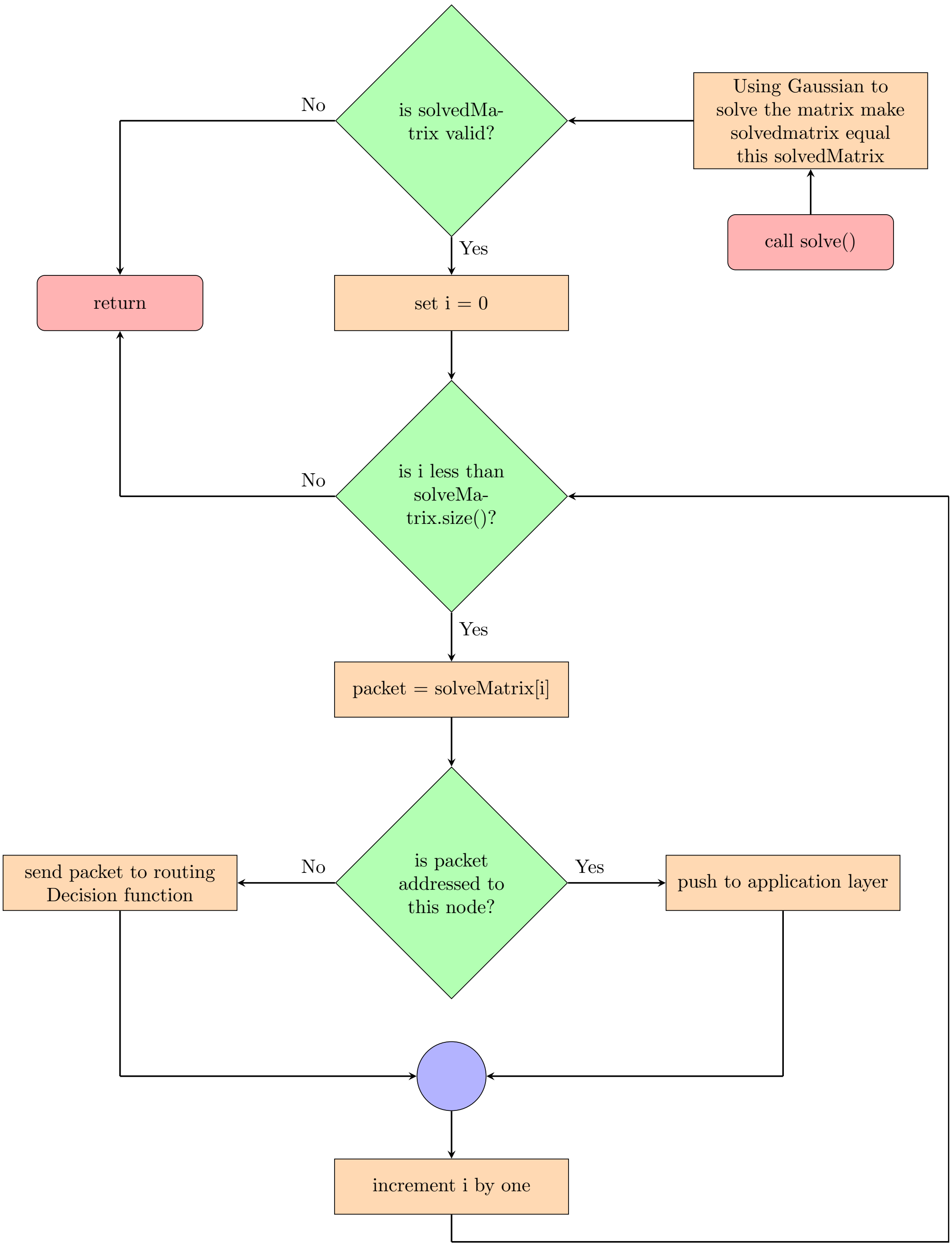

Figure 3.8: A flowchart of the way solved matrices are handled in the routing layer code. This flowchart expands on the pseudocode shown in Listing 3.4.

Chapter 4

APPLICATION OF LOCATION BASED ADDRESSING TO VANET

4.1 Naive Multi-casting using Location Based Addressing

For this scenario, the baseline performance for LBA will be established analytically and with the help of simulations. This is the focus for the simple case where by the network is analysed without network coding being turned on. This analysis allows it to be observed if the simulations are working, and then that will confirm if the gains generated from turning on network coding will be beneficial.

4.1.1 Impracticably of Routing

Routing is an impractical method to send packets through a VANET as discussed in earlier chapters. Because of this impracticality, routing will not be looked at.

4.1.2 Naive Multi-casting

Naive multi-casting is simply when all the nodes in the network broadcast the packets with very little control. This technique will not be analysed because of its fundamental impracticality. Since LBA is the main focus of this research, and then subsequently having network coding applied to the LBA.

4.2 Modelling Naive Multi-casting using Location Based Addressing

It is vital to establish a model that predicts the performance of the network under certain circumstances. Some assumptions need to be made before the model can be established and the model will be used to verify the simulation results - where by if both the model and the simulation results agree, then analysis can begin. Using the topology defined in Equations (3.1) and (3.2), the expected performance of the network can be determined.

4.2.1 Assumptions

The assumptions that are required to be made to create the expected performance are defined below:

- The network topology does not change in packet time.
- The average number of *transmissions* required to cover the whole network can be used.
- The simulation will be set up to allow for the nodes to maintain the ladder topology - and the min-cut of two.
- As the transmission power increases, the number of sends will remain static.

4.2.2 LBA expected performance formula derivation and proof

Based on the assumptions from Section 4.2.1, the theoretical limits of how the system can perform can be derived and proven. To prove that the system is working, the average number of sends that are required to address a network of varied length are required and then this is compared against the simulation results. This requires that a formula of the expected number of sends required to reach a node in a network needs to be derived, then an averaging formula for this is required. Theorem 3 details the transmissions required to address any node in an LBA, Ladder topology network, with the Equation (4.1) derived from this Theorem.

Theorem 3. *The total number of sends, S, that are required to reach a specific node, N, in an LBA network using the Ladder topology is given by Equation* (4.1)*. Figure 4.1 is the graphical representation of this equation.*

$$S_{LBA}(N) = \left\lfloor \frac{N-1}{2} \right\rfloor \times 2 + 1 \tag{4.1}$$

Proof. To prove the formula in Theorem 3, induction is used. Firstly, from Figure 4.2, it can be seen that to send a packet to nodes 1 or 2, only one send is required. It can also be seen that sending to nodes 3 and 4 will require three sends from the nodes in total. Finally, it can be seen that to send to nodes 5 and 6, a total of five sends is required from all the nodes to reach those nodes. This behaviour is assumed to continue for a network of any length and is seen in Table 4.1.

From Table 4.1, it becomes clear that a more elegant formula is required to define the behaviour, and this is where Equation (4.1) was derived empirically. Induction is used to prove this formula works for all cases. The proof requires that the cases for odd and even are dealt with separately.

When N is odd:

Table 4.1: The increase in the number of sends required for a packet to reach a node with LBA using a Ladder topology

Node value	Sends required to reach
1	1
2	1
3	3
4	3
5	5
6	5
$\vdots$	$\vdots$
N	$\begin{cases} N & \text{if } N \text{ is odd} \\ N-1 & \text{if } N \text{ is even} \end{cases}$

If $S(N)$ is valid, then $S(N+1)$ must equal to $S(N)$ and $S(N+2)$ must be $S(N)+2$.

For the condition $S(N+1)$, it can be made that $S(N+1) = \lfloor \frac{N}{2} \rfloor \times 2 + 1$. However, N is odd, thus the division gets rounded down:

$\lfloor \frac{N}{2} \rfloor = \frac{N-1}{2}$, which can be manipulated to be: $S(N+1) = \frac{N+1-1}{2} \times 2 + 1 = S(N)$, which holds true.

For the condition $S(N+2)$, it must be shown that $S(N+2) = S(N) + 2 = N + 2$.

With N being odd, $N-1$ is even, and thus: $\lfloor \frac{N-1}{2} \rfloor = \frac{N-1}{2}$, for odd N values. Using this, we can say that $S(N+2) = \lfloor \frac{N+2-1}{2} \rfloor \times 2 + 1$, and since $N+2$ is odd too, thus:

$$S(N+2) = \frac{N+2-1}{2} \times 2 + 1$$

Finally, from this it is shown that:

$$S(N+2) = N + 2 = S(N) + 2$$

When N is even:

If $S(N)$ is valid, then $S(N)$ must equal to $S(N-1)$ and $S(N)$ must be equal to $N-1$.

For the condition $S(N-1) = S(N)$, the equation starts off as: $S(N-1) = \lfloor \frac{N-2}{2} \rfloor \times 2 + 1$. SInce $N-2$ is also even, thus the division by 2 has no decimals, this allows for the formula to become $\lfloor \frac{N-2}{2} \rfloor = \frac{N-2}{2}$ and thus, from this it is proven that $S(N-1) = N-1$.

While, with $S(N) = \lfloor \frac{N-1}{2} \rfloor \times 2 + 1$, $N-1$ is odd, thus the division by 2 gets floored, thus resulting in the equation now becoming: $\lfloor \frac{N-1}{2} \rfloor = \frac{N-2}{2}$.

From this, the equation becomes $S(N) = \frac{N-2}{2} \times 2 + 1 = S(N-1)$ for even N

Thus, Equations (4.1) holds for all conditions.

□

Theorem 4. *The average number of sends, A, required to address the entire LBA using the Ladder topology, will follow the behaviour seen in Equation (4.2).*

$$A_{LBA}(N) = \frac{1}{N} \times \left\lceil \frac{N^2}{2} \right\rceil \tag{4.2}$$

Proof. Provided Equation 4.1, the average number of sends required is simply the summation of the sends required for a packet to reach a node divided by the number of nodes that are in the network. This trend can be seen in Table 4.2.

Table 4.2: The average number of sends required to send packets to a network of length N, provided that all nodes are uniformly addressed

Number of nodes	Average number of sends required
1	$\frac{1}{1} = 1$
2	$\frac{1+1}{2} = 1$
3	$\frac{1+1+3}{3} = 1.\dot{6}$
4	$\frac{1+1+3+3}{4} = 2$
⋮	⋮
N	$\frac{S(1)+S(2)+S(3)+\cdots+S(N)}{N} = \sum_{i=1}^{N} \frac{S(i)}{N}$

The formula for this behaviour seen in Equation (4.2) from Theorem 4

has been empirically derived, and by induction it is proven to work for all cases. To prove by induction, the odd and even N values are considered separately. Thus, assuming $A(N)$ is true, it must be proven that:
$A(N+1) = \frac{1}{N+1} \times \left\lceil \frac{(N+1)^2}{2} \right\rceil$
If N is odd, then $N+1$ is even, thus $S(N+1) = N$.

$\sum_{i=1}^{N+1} \frac{S(i)}{N+1} = \frac{1+1+3+3+\cdots+N+N}{N+1}$

It is easy to see that this can be arranged as two summations, creating:

$= \frac{2\sum_{i=1}^{\frac{N+1}{2}}(2i-1)}{N+1}$

Using the identity that $\sum_{i=1}^{n}(2i-1) = n^2$ the obtain

$= \frac{2\left(\frac{N+1}{2}\right)^2}{N+1}$

$= \frac{1}{N+1} \times \left\lceil \frac{(N+1)^2}{2} \right\rceil$ thus, proven for when N is odd.

The case when N is even is now considered:

First, the right hand side of the equation generates the following when N is even:

$A(N+1) = \frac{1}{N+1} \left\lceil \frac{(N+1)^2}{2} \right\rceil$

$= \frac{1}{N+1} \left\lceil \frac{N^2+2N+1}{2} \right\rceil$

$= \frac{1}{N+1} \left(\frac{N^2+2N+2}{2} \right)$

If N is even, then $S(N+1) = N+1$, now computing the left side of the equation

$\sum_{i=1}^{N+1} \frac{S(i)}{N+1} = \frac{1+1+3+3+\cdots+(N-1)+(N-1)+(N+1)}{N+1}$

Thus, this breaks down into the two summations:

$= \frac{1}{N+1} \left(\sum_{i=1}^{\frac{N-1}{2}} (2i-1) + \sum_{i=1}^{\frac{N+1}{2}} (2i-1) \right)$

$= \left(\left(\frac{N}{2}\right)^2 + \left(\frac{N+2}{2}\right)^2 \right) \frac{1}{N+1}$

$= \frac{1}{N+1} \left(\frac{N^2+2N+2}{2} \right)$, thus proven for when N is even. □

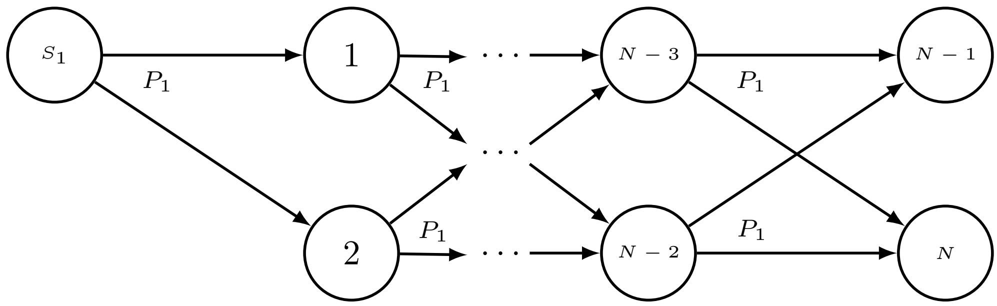

Figure 4.1: The generalized case of a packet being sent down an LBA network with the Ladder topology being maintained. The network is of arbitrary length and the sending node S_1 is sending packet P_1 to node N.

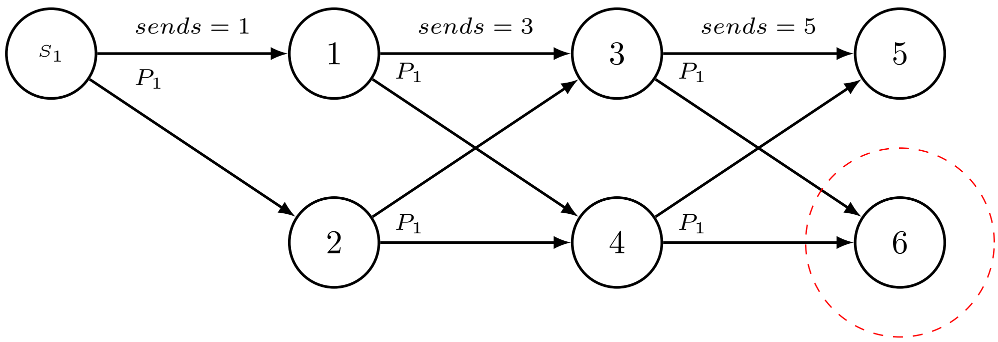

Figure 4.2: A specific case of a sending node S_1 sending packet P_1 to node 6 with LBA and the ladder topology being utilized. The accumulation of transmissions is also being shown.

4.3 Simulation Results and Analysis

After running the simulations, the network performed similarly to how the model predicted. Refer to Appendix B for all the tabulated results used to plot the figures in this section.

4.3.1 One sender with LBA-only

In this scenario, there is only one sending node that is sending to a varied number of nodes, with an example of this scenario seen in Figure 4.3. In this case, the network coding algorithm will perform the same since there is only on packet moving down the network at a time. In Figure 4.4, the model is compared to the simulation results and an adjustment to the model is also included. The adjusted formula can be seen in equation 4.3, since the simulation takes into account random event and slight delays in the network, its performance is expected to drift slightly from the ideal model.

$$A_{LBA}(N) = \frac{1}{N} \times \left\lceil \frac{N^2}{2} \right\rceil + \frac{N}{5} - 0.7 \tag{4.3}$$

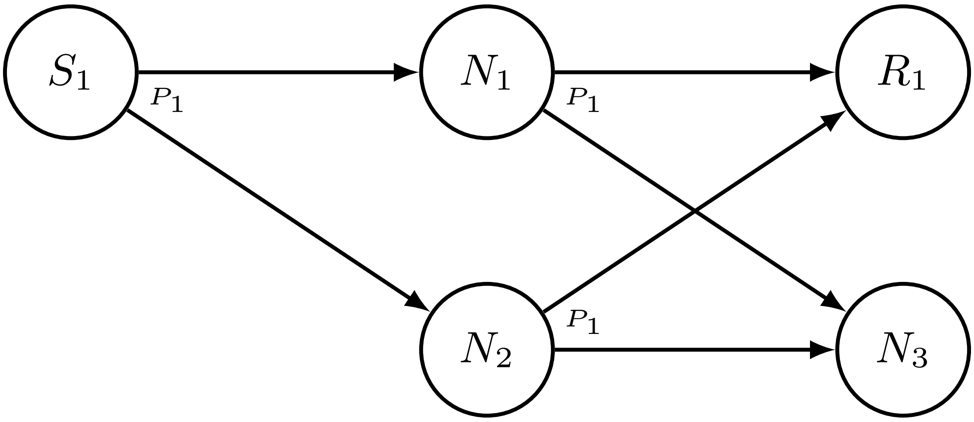

Figure 4.3: The scenario example of one node sending packets through a network. The network can be of arbitrary length, and in this example, only node R_1 is receiving packets, with the intermediates nodes N being the network sustaining nodes.

4.3.2 Two simultaneous senders with LBA-only

In this scenario, the network is tested with two packets moving down it simultaneously as seen in Figure 4.5. The performance of each packet should not change since no network coding strategies are employed. The adjusted formula for this scenario is the same to that of the one sender scenario. The simulation results are shown in Figure 4.6.

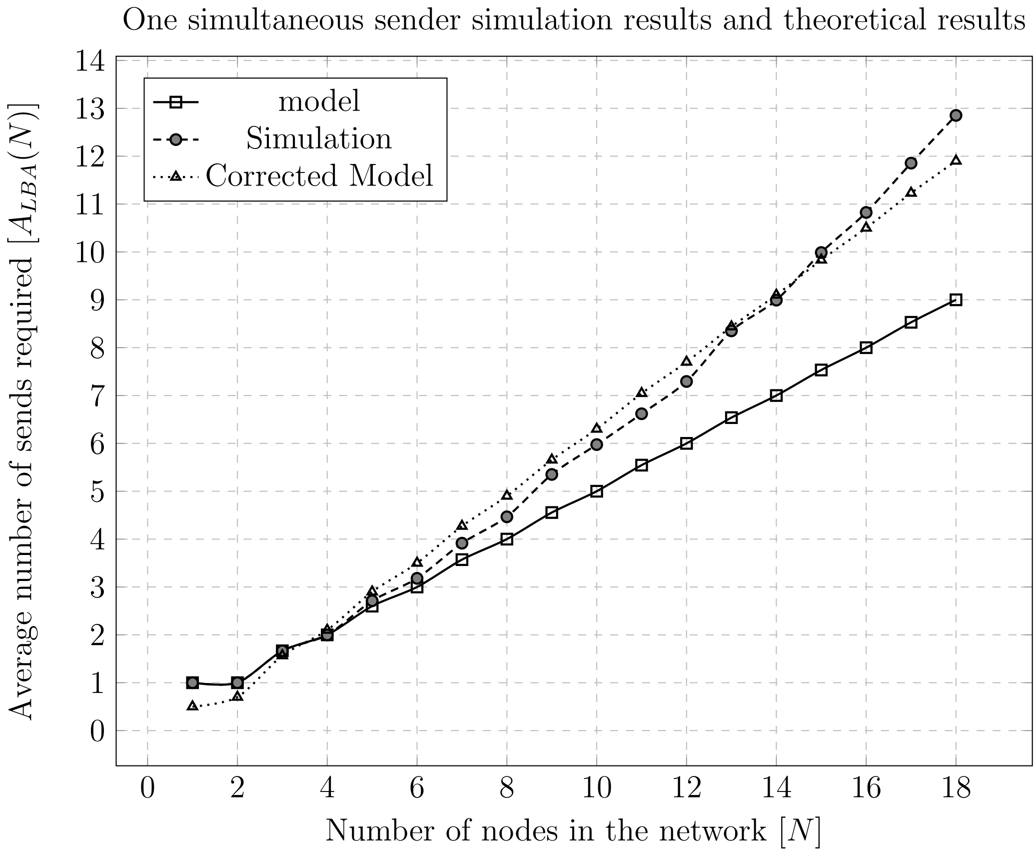

Figure 4.4: The results of having one sender in an LBA network with a ladder topology. The simulation results drift from the theoretical results, however, it is only a slight gradient drift. The corrected formula can be seen in Equation (4.3).

4.3.3 Three simultaneous senders with LBA-only

With this scenario seen in Figure 4.7, three senders are set up and used to send packets through the network. In this scenario, the simulation results, seen in Figure 4.8, began to drift significantly from the model results, and this is due to the network becoming too busy. Three nodes sending their packets at the exact same time starts to cause a large number of dropped packets too as the MAC layer is unable to handle the collisions in the packets. Equation 4.4 shows the adjusted corrected model for three sending nodes.

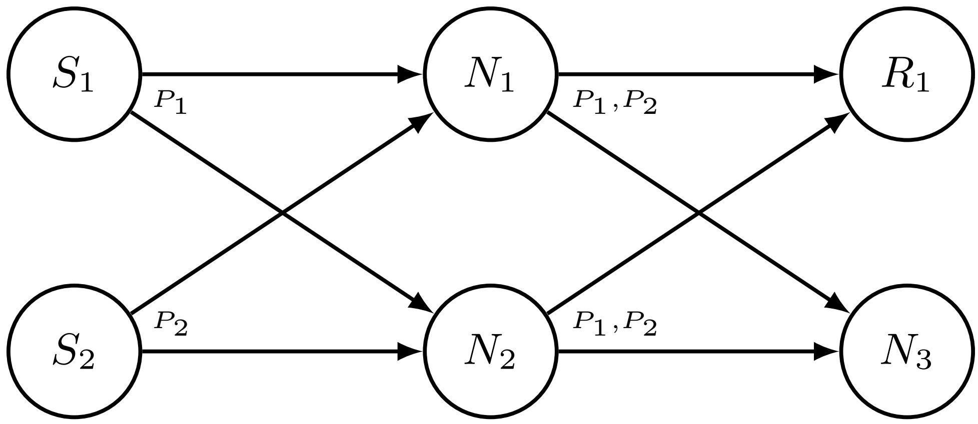

Figure 4.5: The scenario example of two nodes sending packets through a network simultaneously. The network can be of arbitrary length, and in this example, only node R_1 is receiving these packets, with the intermediates nodes N being the network sustaining nodes.

$$A_{LBA}(N) = \frac{1}{N} \times \left\lceil \frac{N^2}{2} \right\rceil + \frac{N}{5} - 0.7 \tag{4.4}$$

4.3.4 Four simultaneous senders with LBA-only

The scenario with four simultaneous senders has been considered and can be observed in Figure 4.9. The simulation software ran into problems while trying to run simulations with more than seven nodes, and this maybe due to issues with the discrete time event based simulator running out of resources due to the chaotic nature of this scenario. However, from the results gathered, seen in Figure 4.10, it is clear that the simulation diverges a lot from the model when there are four senders.

4.3.5 Comparison of different sender scenarios

Here the scenarios, seen in Figure 4.11, show how adding more simultaneous nodes to the network impacts performance. With one and two senders, the

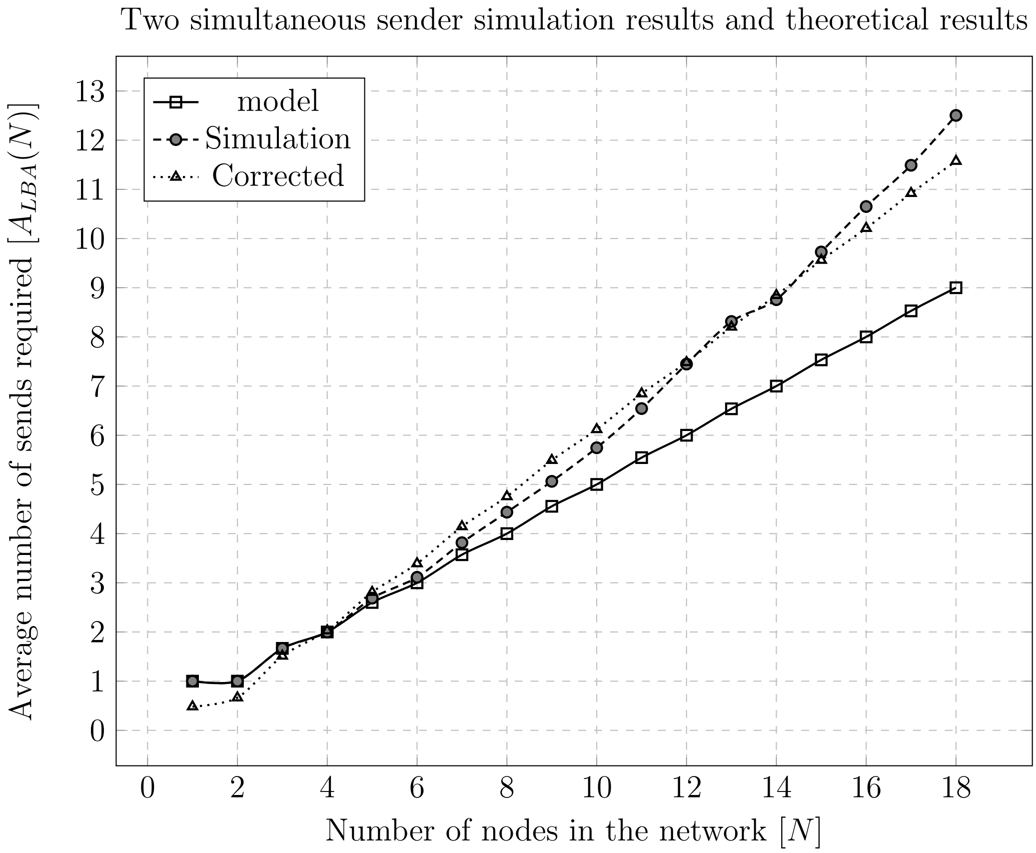

Figure 4.6: The results of having two simultaneous senders in an LBA network with a ladder topology. The simulation results drift from the theoretical results, however, it is only a slight gradient drift. The corrected formula can be seen in Equation (4.3).

average number of sends all the nodes are required to send for packets to move through the network are very close. However, with three simultaneously sending nodes, the performance begins to deteriorate rapidly, and this is due to the network essentially being 'flooded' with packets.

4.3.6 LBA Interference patterns

Understanding how interference influences the behaviour of the topology allows for better analysis. For this simulation, an interfering node broadcasting at high power using a different MAC layer protocol is brought in and out of

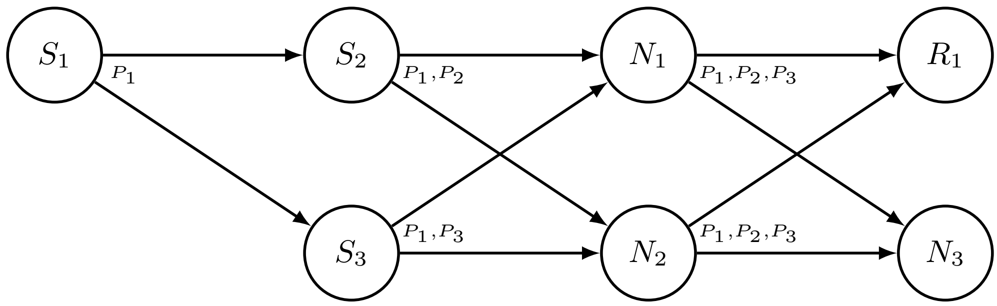

Figure 4.7: The scenario example of three nodes sending packets through a network simultaneously. The network can be of arbitrary length, and in this example, only node R_1 is receiving these packets, with the intermediates nodes N being the network sustaining nodes.

the of the network with the distance that the node travels perpendicular to the network being varied. This simulation begins with the node starting at 500 meters away, and then it proceeds to move towards and away from the network. This interference tested how well the network can recover from something causing a large packet loss. Figure 4.12 shows how the interfering node moves closer to the network and begins to ruin the way packets are received at nodes. This Figure 4.12, the interfering node is given the symbol e. The results can be seen in Figure 4.13.

In Figure 4.13, it can be seen that the more interference there is, the worse the throughput. Both two and three sending nodes scenarios have very similar performance structures, and it can be said that the networks performance is quite consistent. With the case of three senders, the throughput never climbs past 71% due to how noisy the network is with three sending nodes regardless of external interference.

From this, it is observed that the network does experience a rapid drop off when the interfering node travels only 100 meters and less to and from the network. However, the throughput for two sending nodes converges to the throughput for three sending nodes to 54% when the interfering node is only moving back and forth 50 meters from the network. Given that the interfering node is moving perpendicular to the centre of the network, this

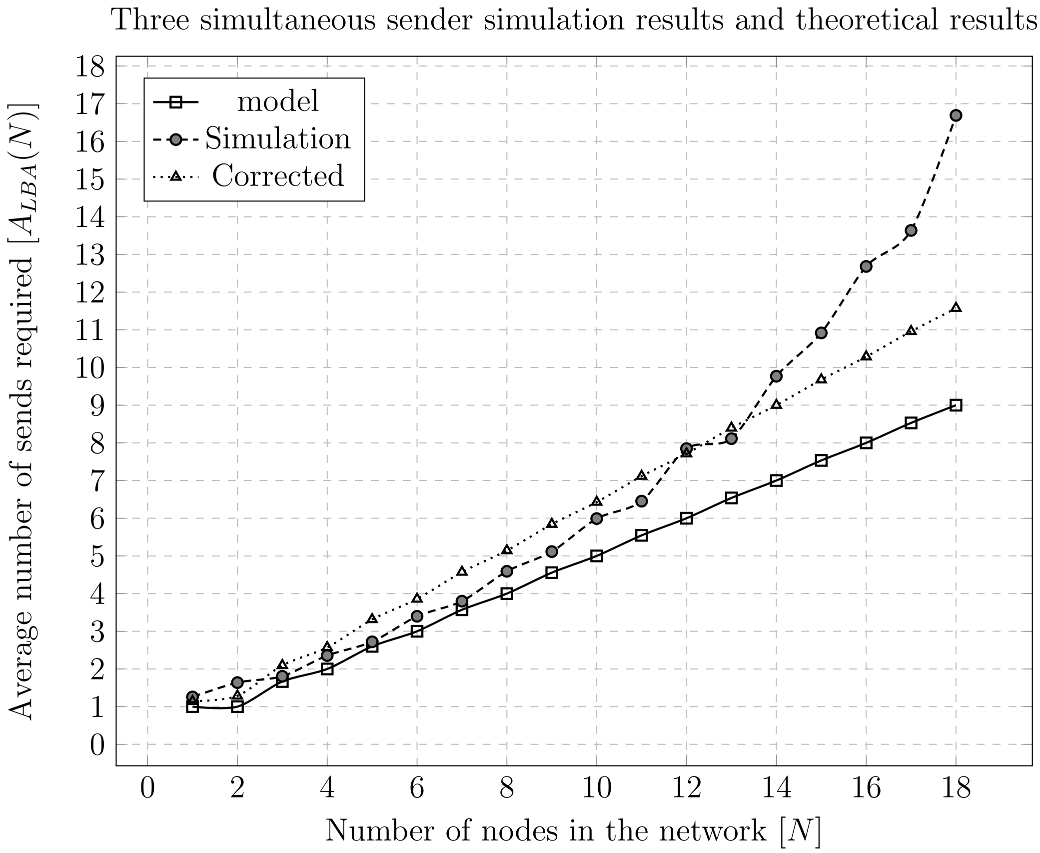

Figure 4.8: The results of having three simultaneous senders in an LBA network with a ladder topology. The simulation results drift from the theoretical results, but as with the other scenarios, it is only a gradient shift. The corrected formula can be seen in Equation (4.4).

makes sense that half the network is being cut off, and the interference is too high for packets to get passed.

4.4 Analysis of LBA protocol

With the results that have been shown, the analysis of Location Based Addressing for a VANET can be made. The analysis is an important part of verifying that the simulations and the models either agree or disagree, and then to what degree. Furthermore, explanation of certain behaviour needs to be detailed and explained, this becomes a big part of the validation, since

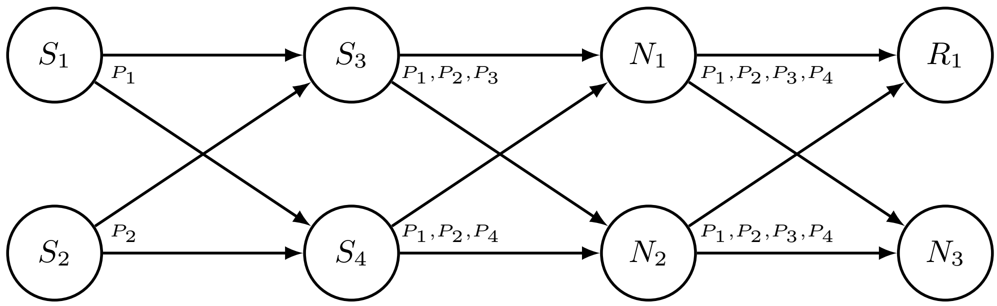

Figure 4.9: The scenario example of four nodes sending packets through a network simultaneously. The network can be of arbitrary length, and in this example, only node R_1 is receiving these packets, with the intermediates nodes N being the network sustaining nodes.

a consistent explanation needs to be created. Finally, critical analysis on the actual performance and how valid this application is needs to be established.

4.4.1 Single Senders with LBA

In this scenario, there is only one sending node that sends packets to every other node in the network, and this scenario is the same for both LBA and network coding applications. Here, the model predicted a linear growth in how many packets were needed to be sent; and this linear growth grows as the network gets longer; this relationship can be seen in 4.2. The simulation results seen in Figure 4.4 shows that a linear growth is exactly what happens; however, the gradient for the average number of sends is slightly different. The simulation results show that there is another factor that needs to be put into play, and that the average number of times a node needs to send a packet increases faster than what the model in 4.2 predicts. The best fit that was matched is in 4.3, which shows a slight shift in the gradient and the crossing point on the x-axis.

This difference in performance between simulation and model is expected though, as the model does not take into account the channel quality, the chance that more nodes may over hear packets and broadcast twice, and the

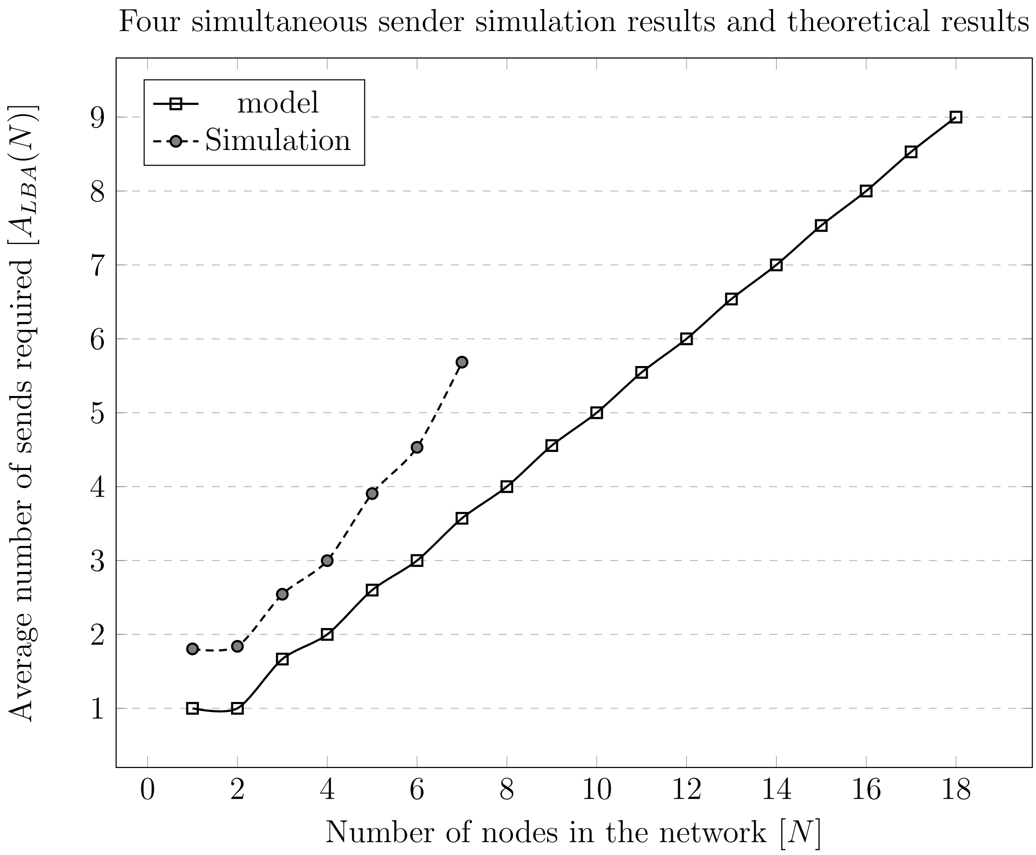

Figure 4.10: The results of having four simultaneous senders in an LBA network with a ladder topology. This simulation was unable to complete an entire cycle, and crashed when there were more than seven nodes in a network.

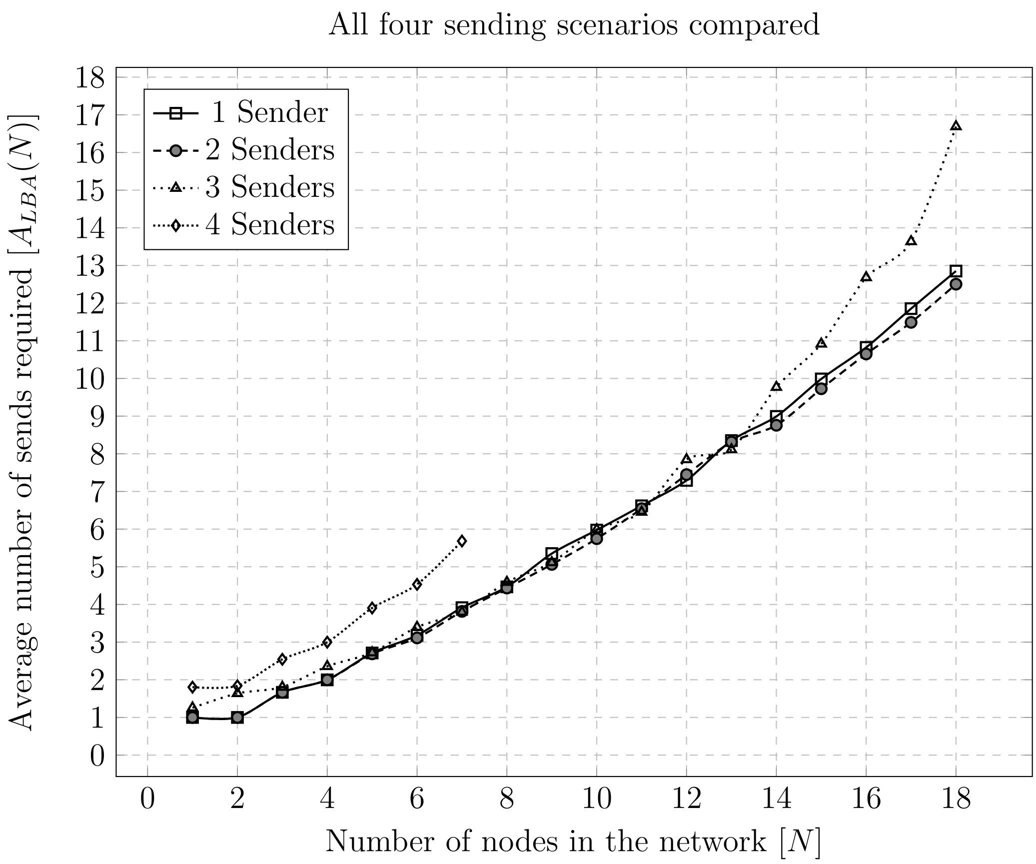

Figure 4.11: The four sending simulations are being compared against each other in this figure. The simulation results show a very close coupling between the nodes for the initial results and shows only minor drifts and deviations.

model assumes that the ladder topology will always be maintained. However, despite the simplification of the model, it does predict the linear relation in how the network will scale as the network grows and the simulation does not drift rapidly from model.

With minor corrections to the model, corrections that allow for more flexibility in terms of random nodes sending out more than one packet, then a very close fit model is made, as seen in Figure 4.4 and in Equation 4.3. From this scenario, the ladder topology is shown to be a powerful tool in showing how the network scales, and gives a good estimate on performance. Minor adaptations to the model can be made to factor in the chance of there being more than one broadcast per node.

The adjusted formula in 4.3 suggests that there is a $\frac{1}{5}$ chance that a node will broadcast out two packets instead of just one, since the simulation result grows at a rate of $\frac{1}{5}$ faster than the predicted model. The slight shift down in the adjusted formula of -0.7 is merely to adjust for the fact that there are almost never any double broadcasts for the first three sent nodes. This can be seen in that the simulation and the model both share exactly the same results for up to the first three nodes that were sent to.

This further strengthens that the ladder topology is applicable in this scenario, since a minor adaptation to the ideal model allows for a very close relationship to what the simulation generates.

4.4.2 Multiple senders with LBA

Two scenarios were looked at for the multiple senders, two senders and three senders. The reason for stopping at three senders was that the simulation would crash when running the LBA-only protocol when four or more senders where generated. The crashes were attributed to the fact that the network became unstable due to too many interfering packets and this instability resulted in the simulator not being able to handle the number of discrete events. With four sending nodes, the simulation could only handle up to a network length of seven.

The results gathered from multiple sending nodes are very interesting. For the scenario where there are two nodes simultaneously sending packets through the network, the exact same performance is generated as the case when there is only one sending node. The adjusted formula remains the same for having two senders as it does for when there is only one sender, implying that the ladder topology is still applicable, and there is an $\frac{1}{5}$ chance that a node will broadcast a packet twice.

The scenario where there are three simultaneously sending nodes is very different though; as the simulation results drift significantly away from the model, especially as the network grows longer. The simulation results show that when there are three simultaneously sending nodes, the number of sends each node makes on average grows exponentially, and not linearly as predicted. This sudden change in behaviour is not due to the model failing, but rather, it is due to the rapid loss in control the network has over the amount of packets being sent. With three sending nodes, each node needs to be broadcast three packets, and this seems to be breaking point of the network, as when the network grows larger, the throughput of the network also falls significantly.

This scenario reveals the major drawback of using a naive multicasting approach; it does not scale very well. As more nodes join the network to transmit a message, the more the nodes are all transmitting and competing to use the available bandwidth. In Figure 4.14, the packet loss of each of the three scenarios is presented, and this shows how the Location Based Addressing based protocol begins to have problems with the third scenario with the higher number of nodes. The sudden increase in packet losses implies that there is simply too much happening in the network for the packets to correctly make their way through the network.

4.4.3 LBA's simulation performance against expected performance

From the results gathered in the LBA scenarios, it can be seen that for relatively well behaved and quiet networks, the Ladder topology and the model derived from it performs extremely well in giving an estimate in performance. The model is also easily adapted to take into account for the minor details that are not present in the ideal model.

The major oversight in the ideal model is that it does not account for the random chance that an intermediate node in the network may over hear a packet, and thus break the ladder topology. In the event of this happening, those random nodes will have a $\frac{1}{5}$ chance of broadcasting a message twice, given that its routing protocol does not detect an error and this 20% chance is easily added into the model as seen in Equation 4.3.

For the LBA-only protocol, the network is unable to scale effectively when there are more than two simultaneously sending nodes. For the cases when there are three or more sending nodes, the protocol keeps having every node broadcast all those packets and the MAC layer simply cannot handle the collisions. This coupled with the fact that MAC layer used does not implement collision detection for broadcast packets, and all the collision handling had to be done in the Routing layer in the simulations.

This demonstrates how scalability is impacted without using a routing protocol to help manage how the packets behave through the network. The ideal model based on the Ladder topology becomes less able to predict how the network would react when there are many nodes trying to send out messages. As a result of this, the network can no longer be said to be predictable for cases where there are many sending nodes; furthermore, the throughput drops off very quickly due to the sudden increase in packet losses.

4.4.4 Scalability of LBA

Scalability for this protocol is very important as the network may extend to be very long with many nodes. In this simulation, only up to 18 nodes are considered, and the general trends are used to determine the scalability as the simulation software has limitations on how large a network can be given the custom routing software.

The scalability of the LBA protocol is very good for when the number of sending nodes is very low - only one or two nodes will be sending at a time. In these bounds, the Ladder topology based model predicts the results very well, and with a minor correction factor, the model predicts the behaviour exactly. At this level, it can be said that LBA is suitable as a simple protocol for VANET given that only sparse information events are being sent out.

Another scenario where the scalability is not a problem is when there is a sudden, short-lived event that has no reason to travel far back. In this case, if a node was only alerting the immediate nodes behind it, then there are no problems with scalability shown by the simulation. This limits how well the application of an LBA-only based protocol for VANET can scale, and to what applications it can scale to.

For the resistance to interference seen in Figure 4.13, this protocol performs quite well. While in the scenarios where the interfering node is very close to the network, only half the network is cut off, and the errors do not propagate further. This improves the argument for this protocol being scalable as it does hold good resistance to massive amounts of interference.

4.4.5 Practical applications of LBA

From looking at how well LBA scales, the applications are limited in a VANET environment, but there are scenarios that it maybe well suited. With LBA showing good resistance to external interference, as well as having low packet loss and scalability for longer network scenarios when there are fewer senders, this shows that an LBA based network protocol can be used as a non-vital, sparsely updating application within VANET. Such examples of these applications would be road condition changes, whereby the state of the road vehicles are travelling on are updated. Conditions such as whether the road is wet, or the state of the traffic light ahead.

LBA may also be useful for emergency scenarios whereby a vehicle updates the nodes nearby of a sudden event change. This is because LBA allows for many nodes to send simultaneous messages in a very short period of time. However, there is a limit to how far back packets can move, and this may be a limiting factor on its applicability but that can be built into the specifications of the protocol.

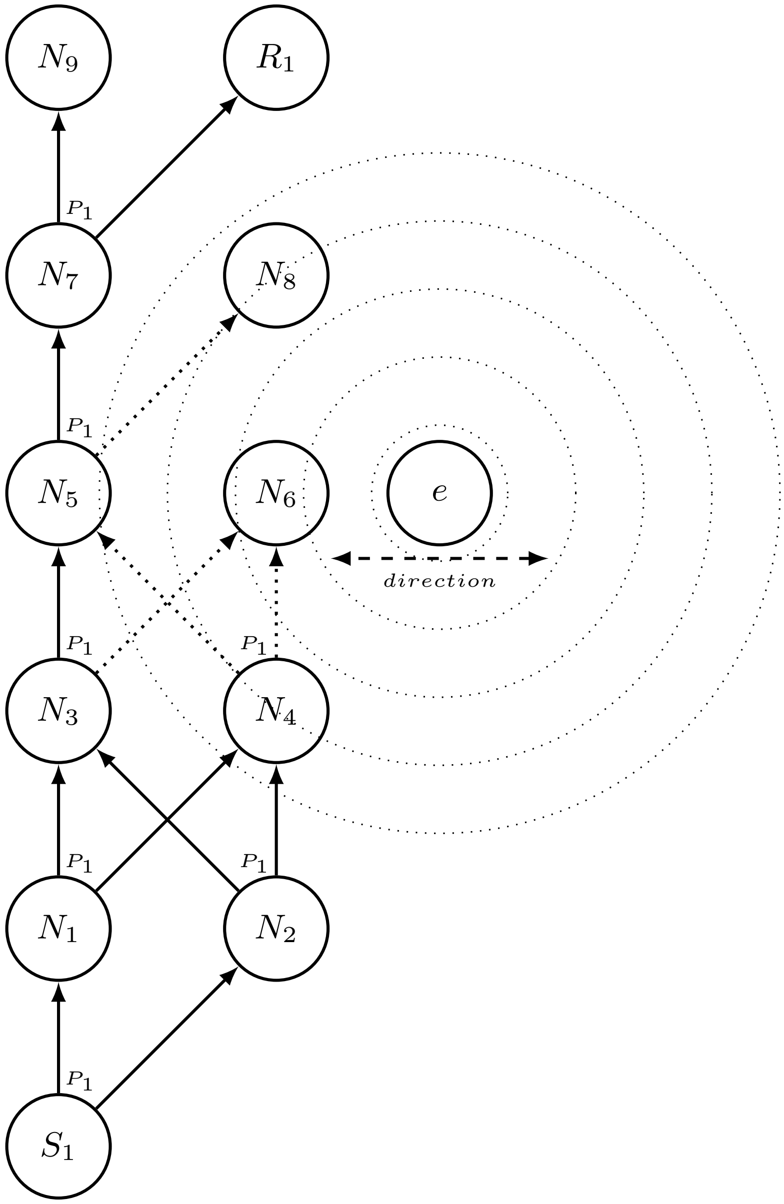

Figure 4.12: Example of how the interference test will work. An interfering node, e, will move towards and away from the network while packets are being set through the network. In this case, e creates enough interference to prevent the message from N_3 and N_4 from reaching node N_6, effectively disconnecting it. Node N_8 also is disconnected from the network as it does not receive the message from N_5 due to interference. Packet P_1 does reach its destination though, as there is no complete disconnection.

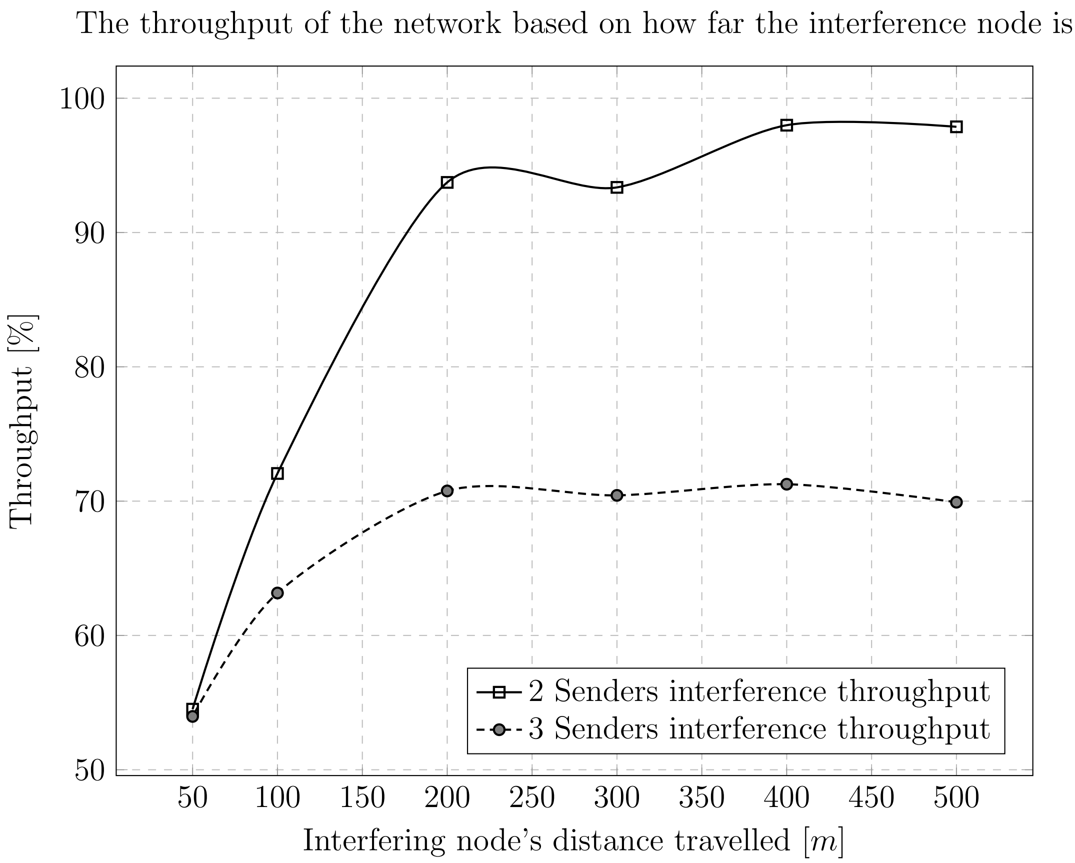

Figure 4.13: Interference behaviour of a node moving in an out of the network where LBA-only is used. In this scenario, the interference node moves to and from the network at a set distance and speed. The test simulates an interference that the network may drift into and out of.

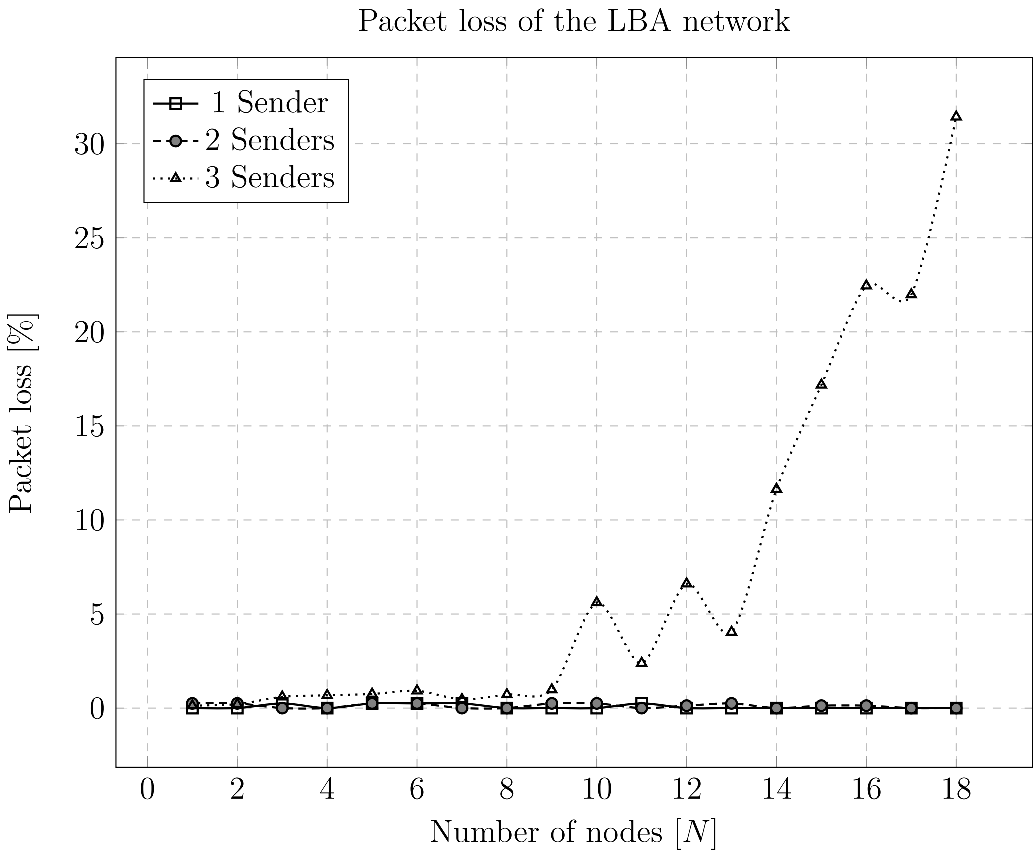

Figure 4.14: These are the simulation results which show the packet loss of the network versus number of simultaneously sending nodes. From this, it is clear that the standard LBA algorithm is unable to handle long networks with many sending nodes.

Chapter 5

APPLICATION OF NETWORK CODING AND LBA TO VANET

5.1 Network Coding with Location Based Addressing

For this scenario, the application of network coding to the LBA over the VANET is analysed and with the help of simulations to help prove the concept and feasibility. Using the assumptions and ladder topology to create a formula to predict the expected performance, which will subsequently be used to compared the results of the simulations.

5.1.1 Advantages and Potential Gains of Network Coding over the Naive LBA Approach

Network coding has been proven to operate at the min-cut of a network in ideal conditions, and this implies that the network will be running at the maximum throughput. LBA has been shown in equation 4.2 that it operates at a rate that is proportional to the number of senders, implying that if there are N senders, then the number of packets being sent will be N times larger.

However, with network coding applied, the expected gain for a network with a min-cut of two will mean that only *half* as many packets will be required to be sent. If the min-cut of the network increases, then number of simultaneous messages can be increased. Thus if the min-cut is 3, then three senders can transmit their messages simultaneously, while with normal LBA, all three messages will need to be transmitted individually.

5.1.2 Advantages over Routing

The main advantage that network coding has over routing is that there is no need for network discovery. With routing it is vital for nodes to understand which node to route a packet to, and this can be a problem for highly dynamic networks since that knowledge of where to forward a packet to keeps becoming outdated.

5.2 Modelling Network Coding with Location Based Addressing

To model this section, the same approach was taken as in 4.2, with the exception being that the rules used to make routing decisions take network coding into consideration.

5.2.1 Assumptions

The assumptions made for this scenario are the same as seen in the section 4.2.1. With one addition, in that the sending nodes will transmit messages at very similar times, so that it is very likely that the messages will be added to the buffers and can be encoded together.

5.3 Expected Performance

Applying network coding to these scenarios prove that there is no simple formula that can be applied generally. The major aspect of network coding is that is allows for the opportunity to divide the number of sends required by the min-cut of the network, however, this saving in min-cut only tends as the network grows in length.

The theoretical performance for each scenario needs to be performed differently, and the closed solution is different, thus three separate closed form solutions are required for the scenarios with two, three and four sending nodes. The logic for deriving each formula is the same though.

For the scenario with three simultaneous senders, refer to Appendix C.

5.3.1 Two simultaneous senders theoretical performance

This is the theoretical performance of the scenario where there are two simultaneously sending nodes in the network. Theorem 5 describes how many sends are required for messages to reach the designation node.

Theorem 5. *The total number of sends for two sending nodes, S_{NC2}, that are required to reach a specific node, N, in a LBA network using the Ladder topology is given by Equation* (5.1)*. Figure 5.1 is the graphical representation of this equation.*

$$S_{NC2}(N) = \left\lceil \frac{N}{2} \right\rceil \tag{5.1}$$

Proof. To derive the formula seen in Theorem 5, the scenario for two senders is considered. Table 5.1 contains the results generated from analysing Figure 5.1.

From Table 5.1, the total sends required for a packet to reach a specific node in the network can be considered below.
The proposed formula that will satisfy this behaviour can be seen in Equation 5.1, to show this formula works for all scenarios, the odd and even scenarios are considered differently.

When when N is odd:

Table 5.1: The total number of packets sent per sending node during to reach a node in a LBA network using network coding and the ladder topology for the scenario where there are two sending nodes.

Node	2 Senders
1	1
2	1
3	2
4	2
5	3
6	3
$\vdots$	$\vdots$
N	$\begin{cases} \frac{N+1}{2} & \text{if } N \text{ is odd} \\ \frac{N}{2} & \text{if } N \text{ is even} \end{cases}$

If $S(N)$ is valid, then $S(N+1)$ must equal to $S(N)$ and $S(N+2)$ must also be equal to $S(N)+1$.

For the condition $S(N+1)$, the equation becomes $S(N+1) = \lceil \frac{N+1}{2} \rceil$. However, since N is odd, therefore $N+1$ is even, thus meaning that $\lceil \frac{N+1}{2} \rceil = \frac{N+1}{2}$. However, the ceiling function does the same as adding a one onto the N before dividing by two, so we can say that $S(N+1) = \frac{N+1}{2} = S(N)$

For the condition $S(N+2)$, the equation becomes $S(N+2) = \lceil \frac{N+2}{2} \rceil$. However, since N is odd, this means N+2 is also odd, and thus the same logic as before follows, we need to add a one to the $N+2$ term, making it $N+3$, since this makes it even and divisible by 2. From this, the equation $\lceil \frac{N+2}{2} \rceil = \frac{N+3}{2}$ is created, and $\frac{N+3}{2} = \frac{N+1}{2} + 1$, and this means that $\lceil \frac{N+2}{2} \rceil = S(N)+1$, as expected.

When when N is even:

If $S(N)$ is valid, then $S(N+1)$ must equal to $S(N)+1$.

This follows that $S(N+1) = \lceil \frac{N+1}{2} \rceil$. Since N is even, $N+1$ is now an odd number, thus the ceiling function breaks down into the equation $\lceil \frac{N+1}{2} \rceil = \frac{N+2}{2}$.

We can manipulate the equation to $\frac{N+2}{2} = \frac{N}{2} + \frac{2}{2}$, which has the final form:

$$S(N+1) = S(N) + 1$$

Thus, Theorem 5 holds for all conditions. □

Theorem 6 describes how many sends are required on average for a network for a given length. The average number of sends is what is recorded in the simulations.

Theorem 6. *The average number of sends required for an LBA network with network coding, a Ladder topology and only two sending nodes is given in Equation* (5.2)*.*

$$A_{NC2}(N) = \frac{1}{2}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N} + 1\right) \tag{5.2}$$

Proof. Provided Theorem 6, the average number of sends required is simply the summation of the sends required for a packet to reach a node divided by the number of nodes that are in the network. This trend can be seen in Table 5.2.

Table 5.2: The average number of sends required to send packets to a network of length N, provided that all nodes are uniformly addressed

Number of nodes	Average number of sends required
1	$\frac{1}{1} = 1$
2	$\frac{1+1}{2} = 1$
3	$\frac{1+1+2}{3} = 1.\dot{3}$
4	$\frac{1+1+2+2}{4} = 1.5$
⋮	⋮
N	$\frac{S(1)+S(2)+S(3)+\cdots+S(N)}{N} = \sum_{i=1}^{N} \frac{S(i)}{N}$

The formula for this behaviour seen in Equation (5.2) from Theorem 6 can be easily derived.
With $S(N) = \lceil \frac{N}{2} \rceil$, the average can be expressed as Equation (5.3).

$$\frac{1}{N} \sum_{i=1}^{N} \left\lceil \frac{N}{2} \right\rceil \tag{5.3}$$

To find the closed form solution of this summation, the conditions for where N is even and odd need to be considered separately. Furthermore, the identity where $\sum_{i=1}^{N} i = \frac{N(N+1)}{2}$ will be used.

For the cases where N is even:

Since the ceiling function allows for a duplication of the integers, we can just half count and multiply the summation by two:

$$\frac{1}{N} \sum_{i=1}^{N} \left\lceil \frac{N}{2} \right\rceil = \frac{2}{N} \sum_{i=1}^{N/2} i$$

This new expression can be written into its closed from as: $\frac{2}{N} \sum_{i=1}^{N/2} i = \frac{2}{N} \left(\frac{\frac{N}{2}(\frac{N}{2}+1)}{2} \right)$, which after some manipulation, it can be written as $\frac{2}{N} \sum_{i=1}^{N/2} i = \frac{1}{2}(\frac{N}{2} + 1)$

However, since N is also even in this case (as defined), the formula can also be expressed in the form as seen in Equation (5.5).

$$\frac{1}{N} \sum_{i=1}^{N_{even}} S(i) = \frac{1}{2} \left(\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N} + 1 \right) \tag{5.4}$$

For the cases where N is odd, there is an asymmetry, as one summation of consecutive integers will count up to $\frac{N-1}{2}$ and the other summation of consecutive integers will count up to $\frac{N+1}{2}$. Thus, for odd N values we will have the equation look like: $\frac{1}{N} \sum_{i=1}^{N} \lceil \frac{N}{2} \rceil = \frac{1}{N} \left(\sum_{i=1}^{\frac{N-1}{2}} i + \sum_{i=1}^{\frac{N+1}{2}} i \right)$

Using the closed form solution for both of the new summations, this allows for the equation to to become:

$\frac{1}{N}\left(\sum_{i=1}^{\frac{N-1}{2}} i + \sum_{i=1}^{\frac{N+1}{2}} i\right) = \frac{1}{N}\left(\frac{1}{2}\left(\left(\frac{N-1}{2}\right)\left(\frac{N-1}{2}+1\right)\right) + \frac{1}{2}\left(\left(\frac{N+1}{2}\right)\left(\frac{N+1}{2}+1\right)\right)\right)$

After some manipulation, $\frac{1}{N}\sum_{i=1}^{N}\lceil\frac{N}{2}\rceil = \frac{1}{2N}\left(\frac{N^2+1}{2}+N\right)$, can be stated. However, N is odd; because of this, $\frac{N^2+1}{2} = \left\lceil\frac{N^2}{2}\right\rceil$, and this means we can adjust this to assert: $\frac{1}{2N}\left(\frac{N^2+1}{2}+N\right) = \frac{1}{2}\left(\left\lceil\frac{N^2}{2}\right\rceil\frac{1}{N}+1\right)$. With Equation (5.2) being the final equation for odd N.

$$\frac{1}{N}\sum_{i=1}^{N_{odd}} S(i) = \frac{1}{2}\left(\left\lceil\frac{N^2}{2}\right\rceil\frac{1}{N}+1\right) \tag{5.5}$$

With this is being the final form of the equation, this means that for all N, Equation (5.6) applies for the scenario with two senders.

$$A_{NC}(N) = \frac{1}{2}\left(\left\lceil\frac{N^2}{2}\right\rceil\frac{1}{N}+1\right) \tag{5.6}$$

□

5.3.2 Three simultaneous senders theoretical performance

Theorem 7 is what describes how many transmissions are required for a packet to reach a node. This theorem is used to establish the foundations that Theorem (8) to be built on.

Theorem 7. *The total number of sends for three sending nodes, S_{NC3}, that are required to reach a specific node, N, in a LBA network using the Ladder topology is given by Equation* (5.7)*. Figure 5.3 is the graphical representation of this equation.*

$$S_{NC3}(N) = \frac{1}{3}\left(\left\lceil\frac{N}{2}\right\rceil\times 4+1\right) \tag{5.7}$$

Theorem (8) describes the average number of sends required to address a network of various lengths.

Theorem 8. *The average number of sends required for an LBA network with network coding, a Ladder topology and three sending nodes is given in Equation* (5.8)*.*

$$A_{NC3}(N) = \frac{1}{3}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{2}{N} + 3\right) \tag{5.8}$$

5.3.3 Four simultaneous senders theoretical performance

Theorem (9) describes the how many transmissions are required to address a specific node in an LBA network with the ladder topology.

Theorem 9. *The total number of sends for four sending nodes, S_{NC4}, that are required to reach a specific node, N, in a LBA network using the Ladder topology is given by Equation* (5.9)*. Figure 5.5 is the graphical representation of this equation.*

$$S_{NC4}(N) = \left\lceil \frac{N}{2} \right\rceil + 0.5 \tag{5.9}$$

Theorem (10) describes the average number of transmissions required to address any node in a network of a specific length.

Theorem 10. *The average number of sends required for an LBA network with network coding, a Ladder topology and four sending nodes is given in Equation* (5.10)*.*

$$A_{NC4}(N) = \frac{1}{2}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N} + 2\right) \tag{5.10}$$

5.3.4 The theoretical performance gains of network coding

For the theoretical gains of network coding, the average number of sends for the standard LBA algorithm is divided by the average number of sends of the LBA algorithm with network coding. The limit is taken to infinity to solve for how the network will behave provided that it is sufficiently long. In Theorem (11), the network coding is show to effectively half the number of transmissions required.

Theorem 11. *Provided the Min-Cut Max-Flow Theorem, applying network coding to LBA over the Ladder topology will provide an asymptotic approach to this max-flow. Provided that the LBA algorithm without network coding will transmit once for every node, network coding should tend to half as many sends due to encoding as the networks length increases, and in doing so, will reach the max-flow, or when there are an even number of sending nodes. Equation* (5.11) *is the definition.*

$$\lim_{N \to \infty} \frac{A_{LBA}(N)}{A_{NC2}(N)} = 2 \tag{5.11}$$

In Theorem (12), the solution shows that a sub-optimal solution is produced. This is because there are three packets being encoded over a network with a min-cut of two.

Theorem 12. *For the case where there are 3 sending nodes, the asymptotic performance only approaches 1.5 as opposed to 2 as seen in Theorem* (11)*. The expression for this can be seen in Equation* (5.12)*.*

$$\lim_{N \to \infty} \frac{A_{LBA}(N)}{A_{NC3}(N)} = \frac{3}{2} \tag{5.12}$$

5.4 Simulation Results and Analysis

For these scenarios, the performance of how network coding being applied to the network would impact performance, and how they differ from the idealistic performance. Refer to Appendix B for all the tabulated results used to plot the figures in this section.

5.4.1 Two simultaneous senders with LBA and network coding

In this scenario, two simultaneous senders are sending to random nodes in a network. The length of the network is increased and the number of times each packet needs to be broadcast by the other nodes of the network is recorded. The average number of sends required from each node is then calculated, and this can be seen in Figure 5.2. There is a slight difference between the performance of the simulation to the model, and the corrected model is shown in 5.13.

$$A_{NC2}(N) = \frac{1}{2}\left(\frac{1}{N}\left\lceil\frac{N^2}{2}\right\rceil + 1\right) + \frac{N}{11} - 0.3 \tag{5.13}$$

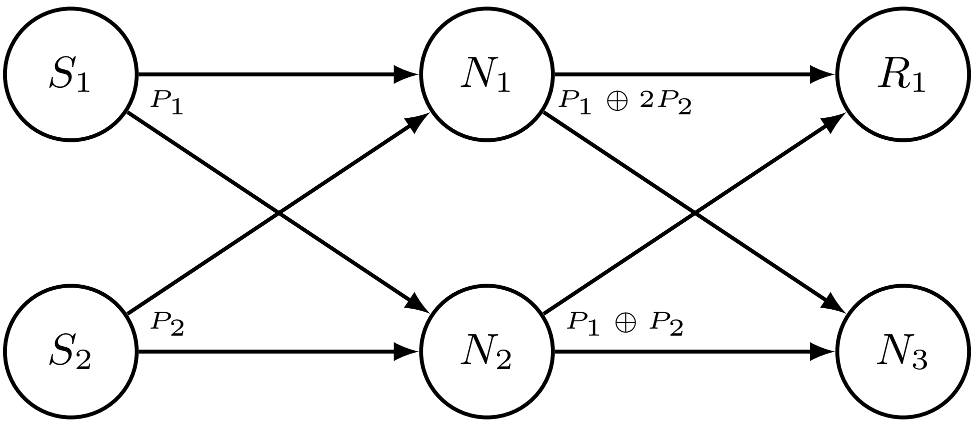

Figure 5.1: Example of how two simultaneous senders in an LBA network with network coding. The intermediate nodes N_1 and N_2 encode the received packets and are only required to send one encoded packet. As a result, node R_1 receives enough novel packets to decode.

5.4.2 Three simultaneous senders with LBA and network coding

In this scenario, three senders send their packets out simultaneously to a network of varied length. This scenario yielded very close results to the expected

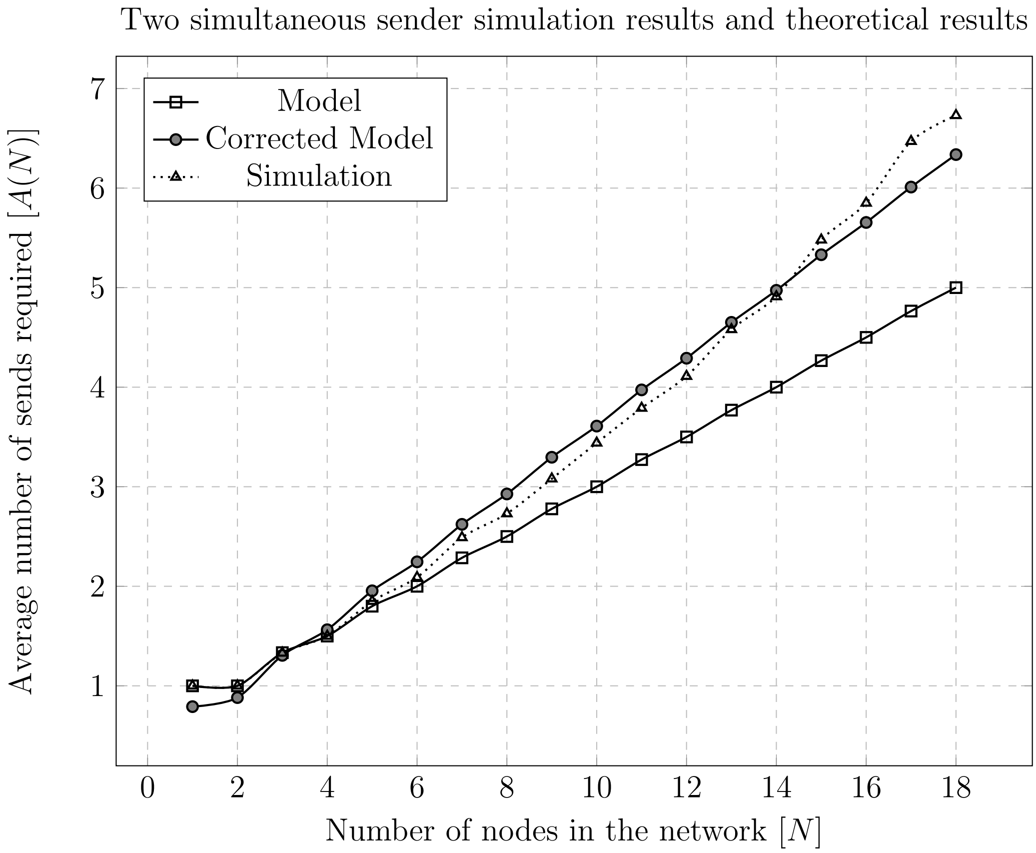

Figure 5.2: The simulation results against the theoretical results of two simultaneous senders in a LBA network with network coding turned on, with the ladder topology. These shows there is a slight deviation from the theoretical results.

performance described in Equation (5.8), however, with three senders, the number of dropped packets is much higher due to the high levels of noise through the network.

5.4.3 Four simultaneous senders with LBA and network coding

This scenario involves having four simultaneous senders sending packets to the nodes in a network of varied length. With four sending nodes, the network will be placed under significant load, as there will be four packets than need

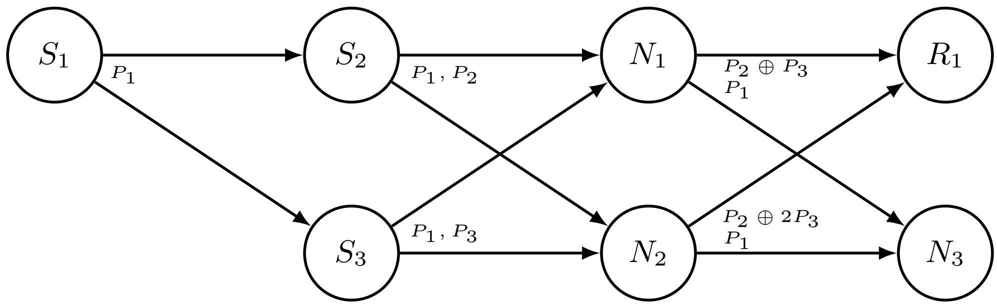

Figure 5.3: Example of three simultaneous senders using LBA and network coding. From this, the senders S_2 and S_3 encode their packets, P_2 and P_3, with the packet P_1. The intermediate nodes N_1 and N_2 are required to send at least two encoded, novel packets which follow a random encoding scheme. The node R_1 thus receives four novel packets, and only needs three to correctly decode, thus it can decode packets P_1, P_2 and P_3.

to move through the network. Figure 5.6 shows the simulation results for how the network performed. There is a drift from the theoretical results, and this maybe due to the load that the network is under. The corrected equation for four sending nodes can be seen in Equation (5.14).

$$A_{NC4} = \frac{1}{2}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N} + 2\right) + \frac{N}{4} - 0.5 \tag{5.14}$$

5.4.4 All the different sending scenarios with LBA and network coding

This is the comparison of the performances between adding the sending nodes, with the results being seen in Figure 5.7. What is interesting with these results, is it shows how with four senders, the performance is only slightly better than one sender, implying that network coding might not scale very well in very heavy loads.

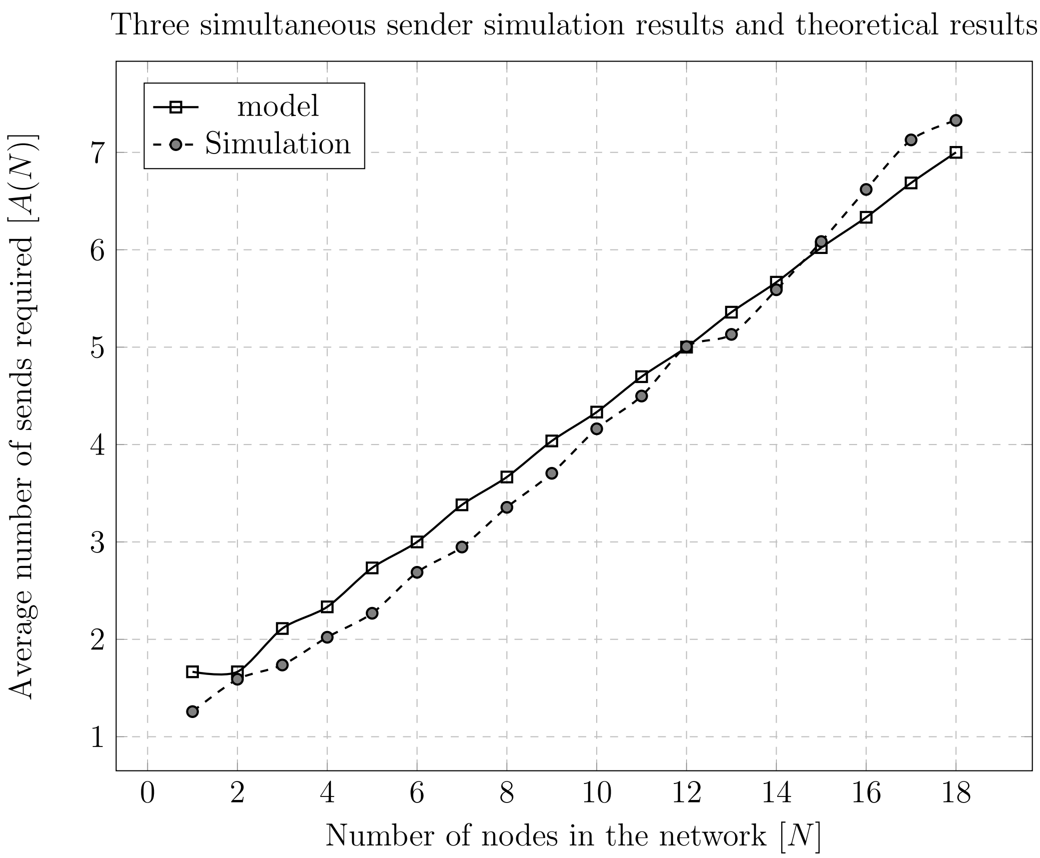

Figure 5.4: The simulation and theoretical results for the scenario where there are three simultaneous senders over LBA with network coding and using the ladder topology. The simulation results agree well with the theoretical expectations.

5.4.5 Two senders with and without network coding

The scenarios where two senders are sending messages to the nodes within a network are shown in the Figure 5.8. This scenario shows how the application of network coding allows for the number of sends required to be greatly reduced compared to the more naive approach of broadcasting with LBA. With the network having a maintained min-cut of two, the network coding clearly demonstrates that network coding only requires half the number of broadcasts, which is exactly as predicted.

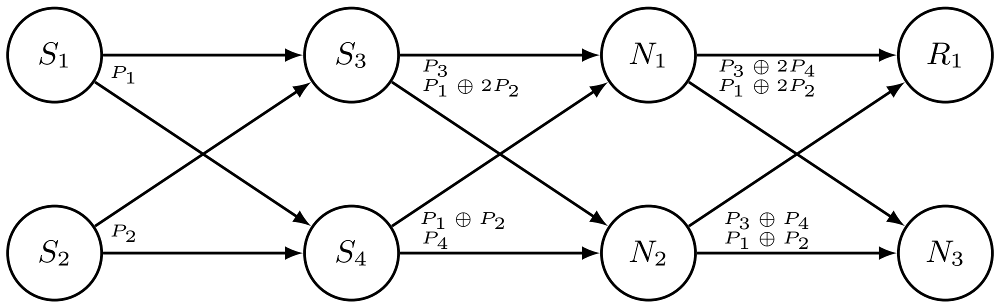

Figure 5.5: Example of four simultaneous senders using LBA and network coding down a ladder topology. From this example, nodes S_3 and S_4 encode packets P_1 and P_2 with their respecting packets. The intermediate nodes N_1 and N_2 each create two novel packets that allow node R_1 to receive four novel packets which allows it to decode and extract all four packets. With a min-cut of two, and four packets being sent, each node is required to send only two packets as seen in this example.

5.4.6 Three senders with and without network coding

In Figure 5.9, it is shown how network coding performs against the non-network coding based LBA algorithm. From this figure, it becomes clear that network coding allows for a more scalable system, as its performance does not deteriorate as quickly. Furthermore, it behaves in the way predicted by the model. This demonstrates that the addition of network coding improves the scalability of the network.

5.4.7 Four senders with and without network coding

In Figure 5.10, the simulation results for LBA-only and LBA with network coding are compared. In this example, the scenario with LBA-only could only run up to seven nodes, there after the simulation crashed. The network coding scenario ran successfully through all the nodes. The simulation software could not handle the number of events with the LBA-only case and as a result it was unable to run.

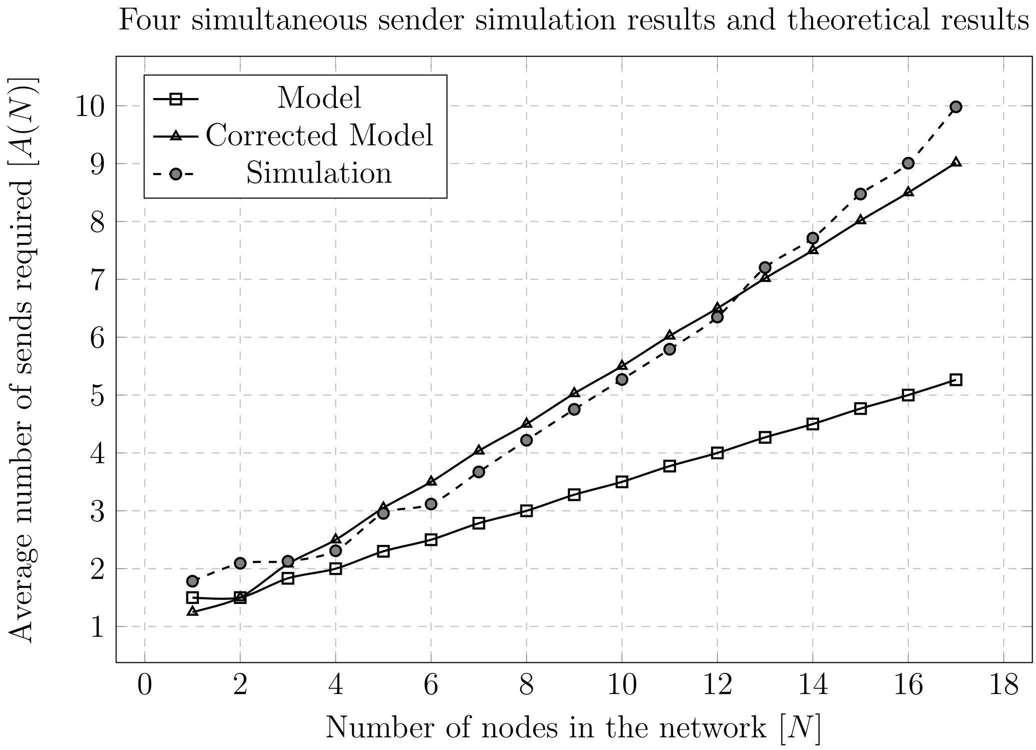

Figure 5.6: Simulation results for when there are four simultaneous senders. The results show that under the scenario where there are four sending nodes, there is a drift from the theoretical results.

5.4.8 LBA with Network Coding Interference patterns

This simulation tests the response of LBA with network coding when interference is introduced. With network coding multiple packets are encode together and this may allow for interference to impact more than just one packet. The same approach was performed for this test as the LBA-only scenario in the previous chapter, where by an interfering node is transitioned to and from the network, but the distance that interfering node travels is varied. The shorter the distance the interfering node travels, the closer it is to the network and the more it will interfere. The results for this simulation can be seen in Figure 5.12.

In Figure 5.12, it is clear that interference has a very strong impact on performance; with there being a sudden drop in throughput in the scenar-

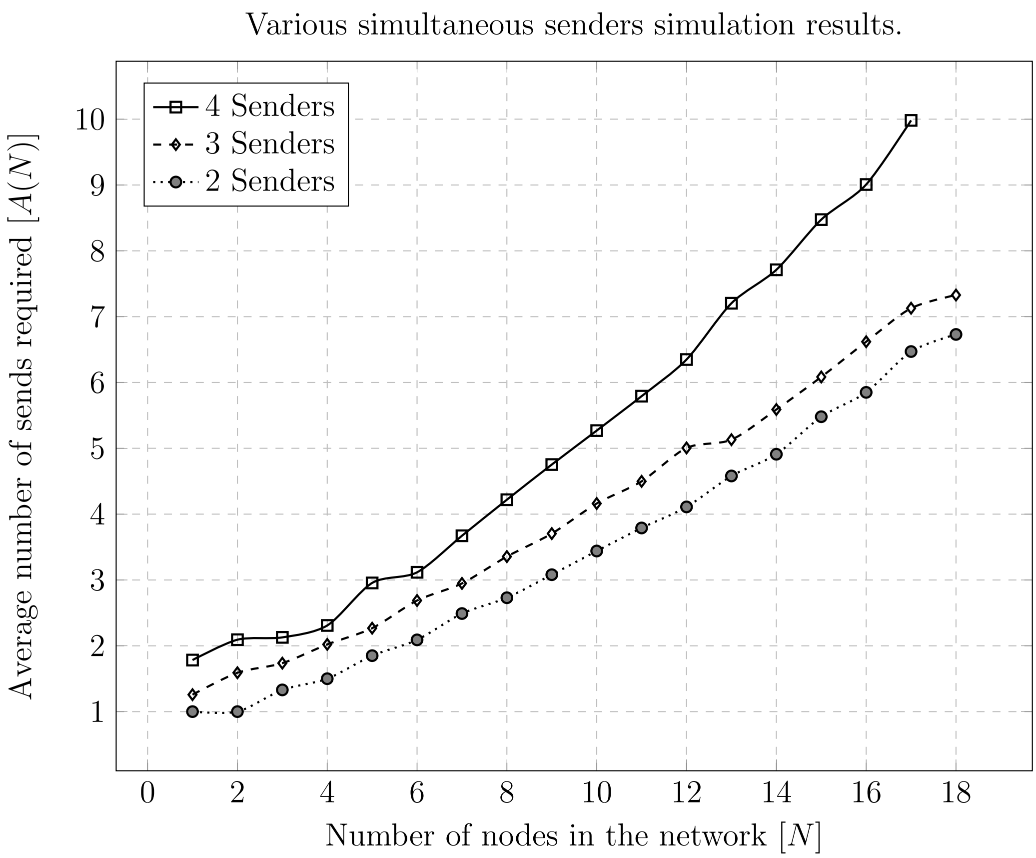

Figure 5.7: All three simultaneous sender scenarios for LBA with network coding enabled, showing that there is a significant drift for the four sending scenario compared to the three and two sending scenarios.

ios where the interfering node only travels 100 meters away from the other network. This sudden and rapid drop in performance many a result of the encoded packets being impacted and the nodes drop the network are not receiving enough packets to decode the messages. The sudden drop off persists and shows that when the interfering node is moving back and forth only 50 meters from the network, the scenario with two sending nodes only has 35% throughput, and the scenario with three sending nodes only has a throughput of 19%.

Since the interfering node is only moving towards and away from the network in simulations, and it has limited impact on the network for cases that

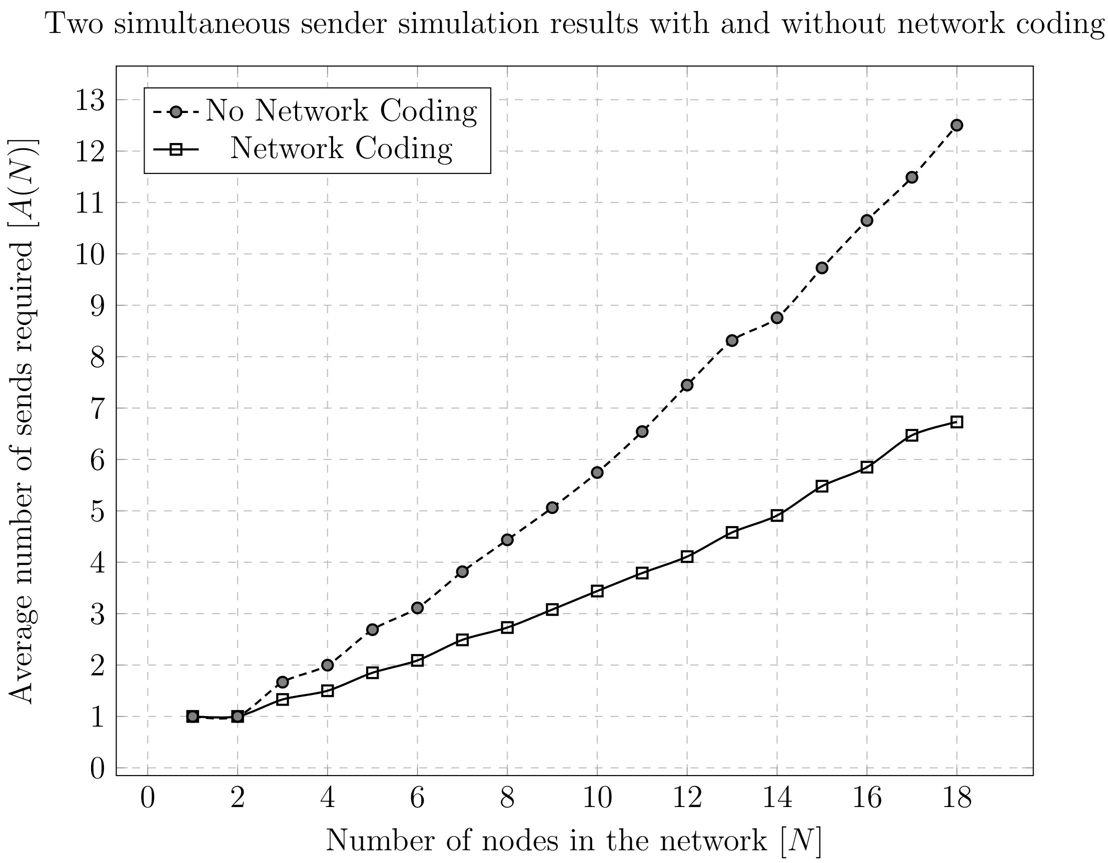

Figure 5.8: Two simultaneous senders with and without network coding enabled. These demonstrates the power and benefit of enabling network coding. Furthermore, network coding allows for the benefit of almost double the throughput of the non-encoded based scheme.

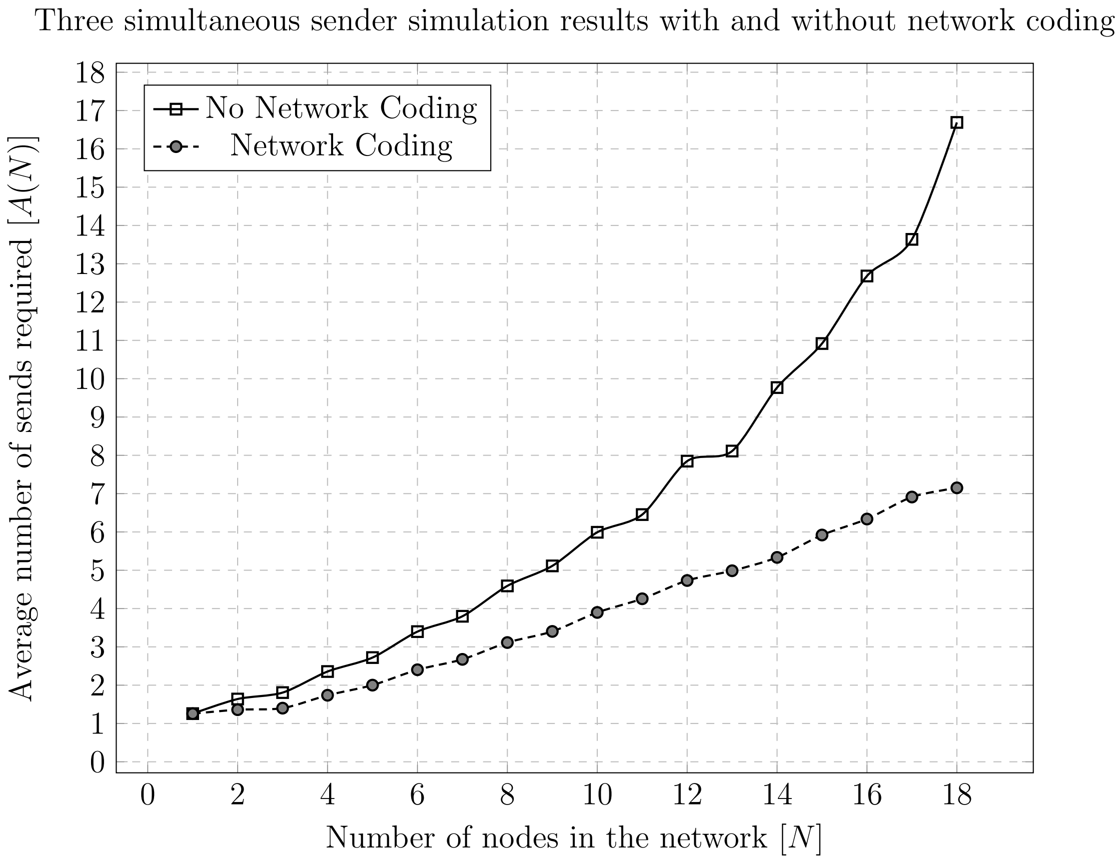

Figure 5.9: Three senders with and without network coding enabled. This shows the massive gains that network coding adds over a naive broadcasting technique. However, in this scenario, there is no gain of 2 as network coding cannot encode three packets as efficiently as just two packets. As a result the gains are closer to about 1.5 times better.

Four simultaneous sender simulation results with and without network coding

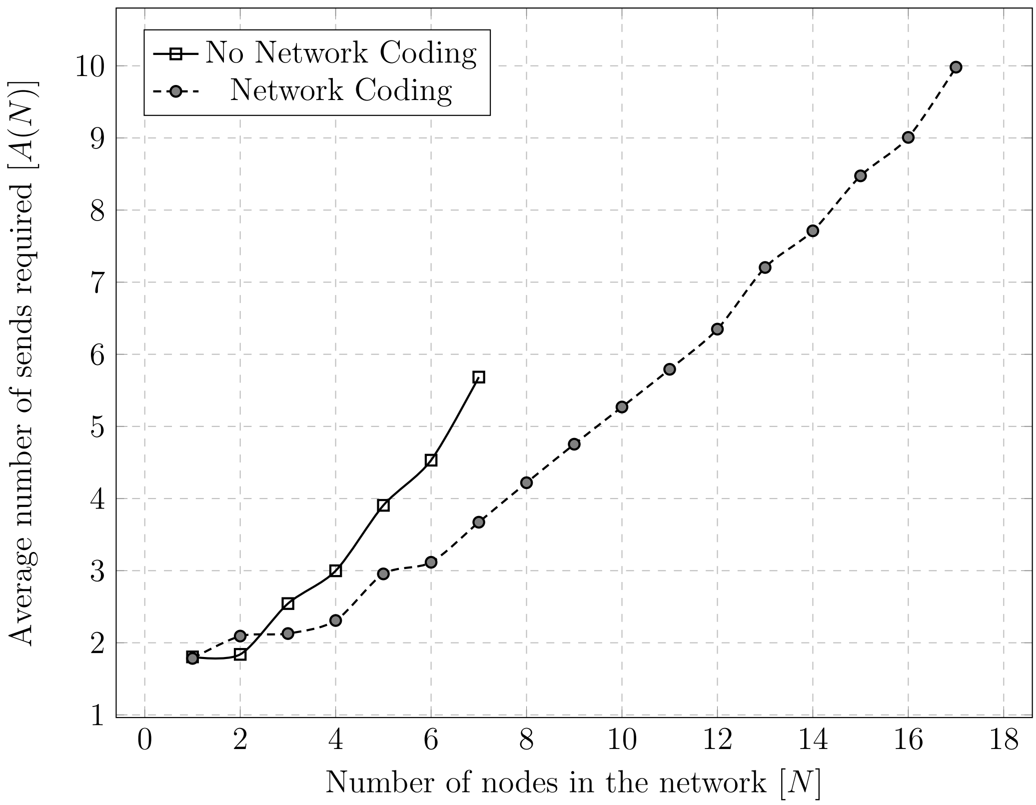

Figure 5.10: Four senders with and without network coding. In this scenario, the standard LBA algorithm is unable to be simulated past seven nodes as the network becomes too chaotic and the network crashed. Network coding on the other hand is much more stable and as a result is far better suited at handling a large number of packets moving through the network at the same time.

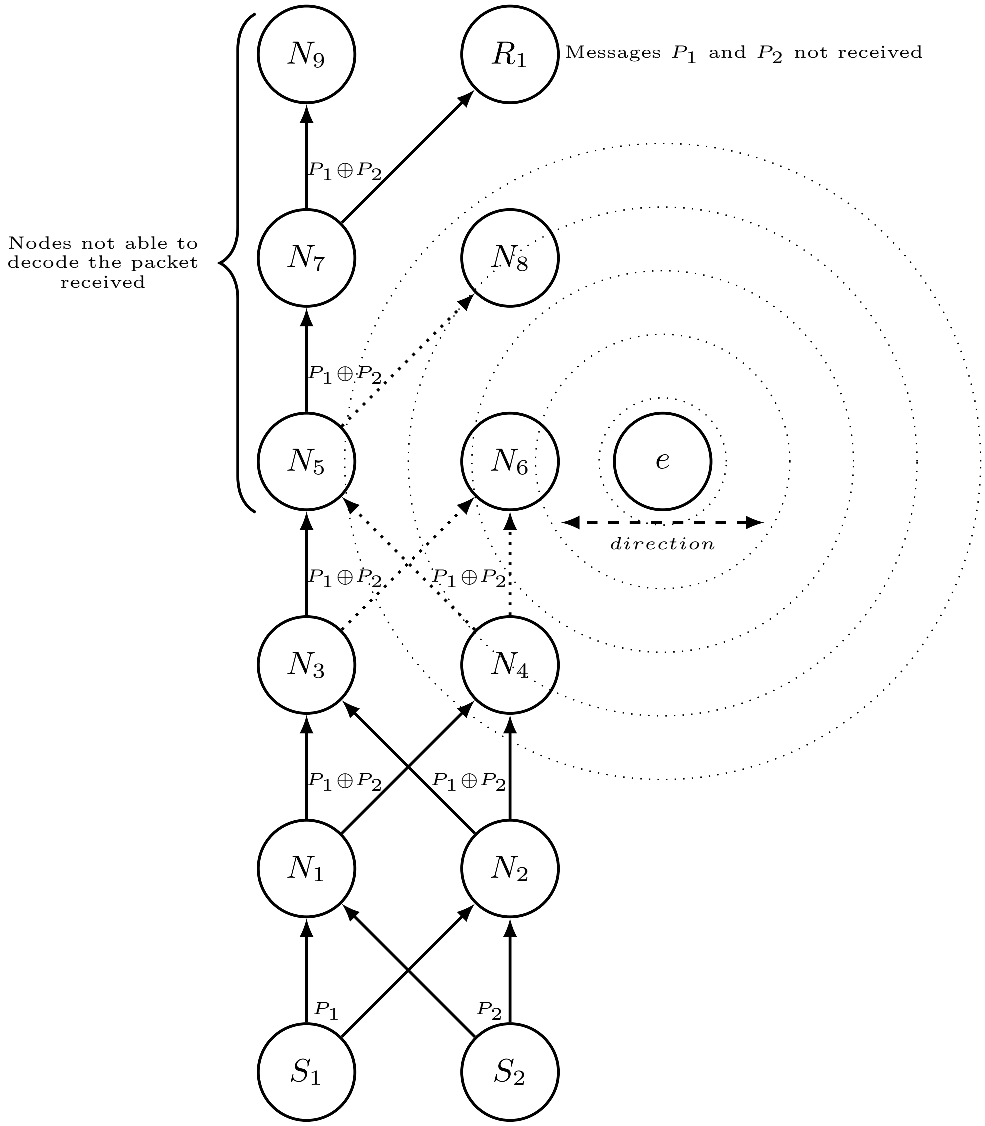

Figure 5.11: An example of the interference simulation scenario for network coding. Node e is the error inducing interference node that moves towards and away from the the network. In this example, the messages from node N_3 and N_4 are not received by node N_6, effectively disconnecting it. While node N_5 receives a novel packet from N_3, it only received one novel packet. Since N_6 receives no packets, node R_1 does not enough novel packets to correctly decode and extract P_1 and P_2. Because of this property, nodes N_7,N_8,N_9 and R_1 are effectively disconnected from the network.

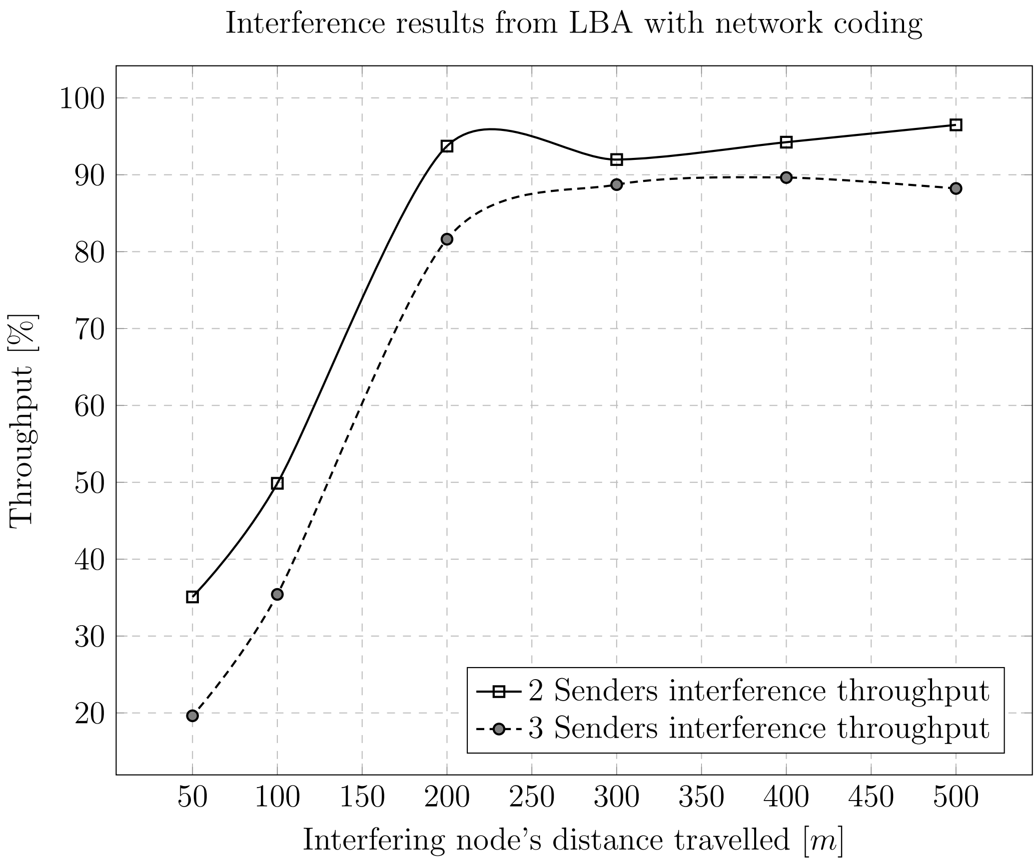

Figure 5.12: The network performance when there is an interference node moving towards and away from a network. With the scenarios where there are two and three sending nodes being shown in this graph.

it moves a greater distance than 200 meters, it can be postulated that network coding has some recovery, but it is not significant enough when there is a lot of interference. Furthermore, as more packets are encoded into the network, the impact on interference increases massively since throughput is almost half for the scenario where there are three sending nodes as opposed to the case where there are only two sending nodes.

Given the interfering node is moving back and forth perpendicular to the network at the middle point of the straight network, and the throughput is less than 50% for the 50 meter case, it can be postulated that more than just packets further down the network are being disrupted. Network coding does appear to have a major weakness to interference, and the more packets that are encoded together, the worse the impact of interference is.

5.4.9 LBA with Network Coding Interference versus LBA-only

This is section focuses on the comparison between the protocol that only uses LBA versus the protocol that uses both LBA and network coding. The scenarios where there are two and three senders are compared and these results can be seen in Figures 5.13 and 5.14. Ordinary LBA offers a case where by packets are just broadcast naively, but this case adds more redundancy in times when not many nodes are transmitting. Furthermore, since network coding encodes packets together, the impact of interference is substantially more severe since a dropped packet may mean two or three encoded packets cannot be solved. The relative throughput show here is how many of the sent packets are received correctly.

In Figure 5.13, the scenario of the throughput of two sending nodes for LBA-only as well LBA with network coding are shown. The relative throughput for both scenarios are similar for when the interfering node is moving back and forth 200 meters or more. Network coding does not experience substantially worse performance until the interference gets too great. For the

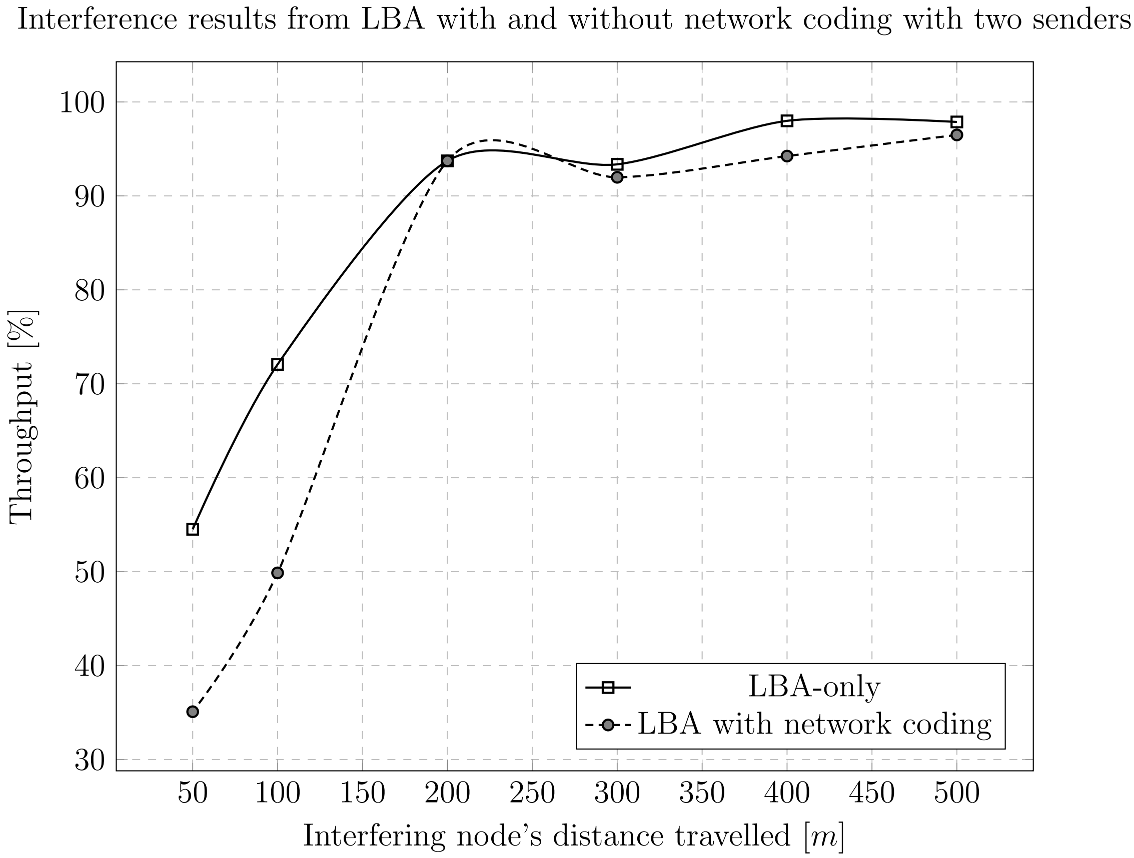

Figure 5.13: Network throughput for when there is an interfering node moving a certain distance from a network for LBA-only and LBA with network coding scenarios. This figure looks at the case when there are two sending nodes. Very similar performance can be seen between the standard LBA and the LBA with network coding, with the exception for when the interfering node is closer than 150 meters to the network. Network coding is unable to correctly decode packets when there are too many dropped packets.

scenarios where the interfering node moving back and forth only 50 and 100 meters; the application of network coding is severely detrimental to throughput when compared to just using LBA. This may imply that the network coding as a protocol needs to be implemented opportunistically, and not absolutely.

In Figure 5.14, a very similar situation is shown as seen in Figure 5.13, with the only exception being that network coding dominates in throughput for lower interference situations. However, as seen before, when the interfering node is moving to and from the network 50 and 100 meters away, network coding experiences a massive drop in throughput. Due to the higher number of encoded packets, the impact of interference is just as expected - substantially worse and detrimental.

5.5 Analysis

The results from the protocol that utilizes both network coding and LBA have been gathered and analysis can be made on the impact of these results. Given that an ideal model based on the Ladder topology was used, how close that model predicted behaviour is to what the simulations show gives insight into the applicability of the model, as well as provides whether the simulation's results are correct. Critical analysis on whether network coding has an application, as well as its drawbacks needs to be made and finally whether there are any applications that are suitable for this protocol in VANET.

5.5.1 Multiple senders over LBA with network coding

The three scenarios that were looked at with the multiple senders are when there are two, three and four senders. Unlike the case where LBA-only is being used, this case was able to complete the simulation with four simultaneously sending nodes. This is mostly attributed to the way that when network coding lowers the required number of packets to be sent, given the min-cut is at least two.

Interference results from LBA with network coding with three senders

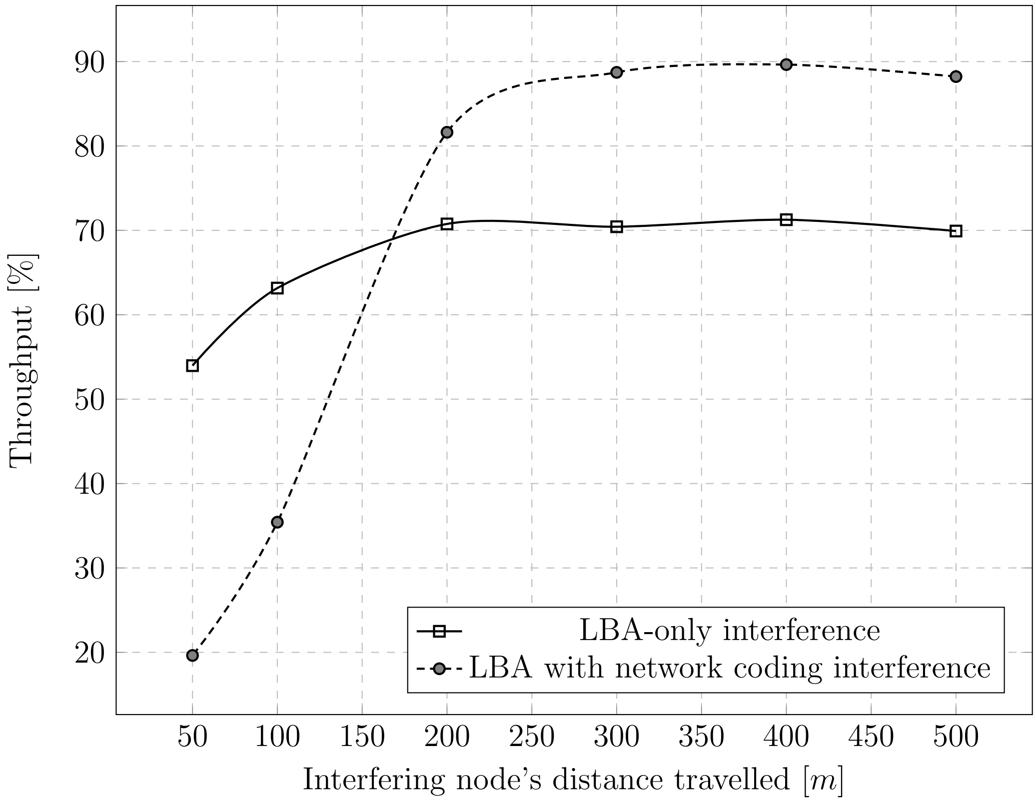

Figure 5.14: Network throughput for when there is an interfering node moving a certain distance from a network for LBA-only and LBA with network coding scenarios. This figure looks at the case when there are two sending nodes. Network coding shows substantial improvement in the scenarios where the interference is not substantial, such as when the interfering node is starting at 200 meters away. However, when the interfering node is closer, it ruins performance much faster and there are too many packets being dropped, disabling their correct reception of them.

From the results gathered from the two, three and four simultaneous senders shows how scalable the network coding scenario is. The behaviour from the simulations show a very close relationship to the predicted results. Furthermore, network coding shows promising results for larger networks because of its close coupling to the predicted performance.

Figure 5.15 shows the packet loss for the different scenarios that the LBA with network coding simulations generated. Curiously, there was a massively high number of packet losses for the situations where there are three and four sending nodes. The high packet losses stabilize as the network gets longer. The reason for the high number of packet drops might be due to the network coding not having time to correctly encode packets effectively when the network has very few nodes. This is a very different situation to the simple LBA algorithm which only experiences very high packets losses when the network get too long and there are more than two packets moving down it simultaneously. From analysing the results in Figure 5.15, network coding may not be an appropriate algorithm to use when the addressed packets are only a single hop away from the sending nodes. However, the algorithm works well for longer networks as it allows for the network to stabilise and transmit reliably. The ordinary LBA algorithm is much better at handling packets that are not destined to travel far from the transmitting node.

5.5.2 LBA with network coding's simulation performance against expected performance

LBA with network coding behaves very closely to how it is predicted to perform as seen in Figures 5.2, 5.4 and 5.6. The scenario with only two simultaneous senders is slightly off the expected results, but a minor correction to the formula was a very close fit 5.13. The parameters on the formula imply that there is not a perfect encoding rate as packets migrate through the network. This makes sense as the drift gets more pronounced down the network.

For the scenarios with three and four senders, they fit the models exception-

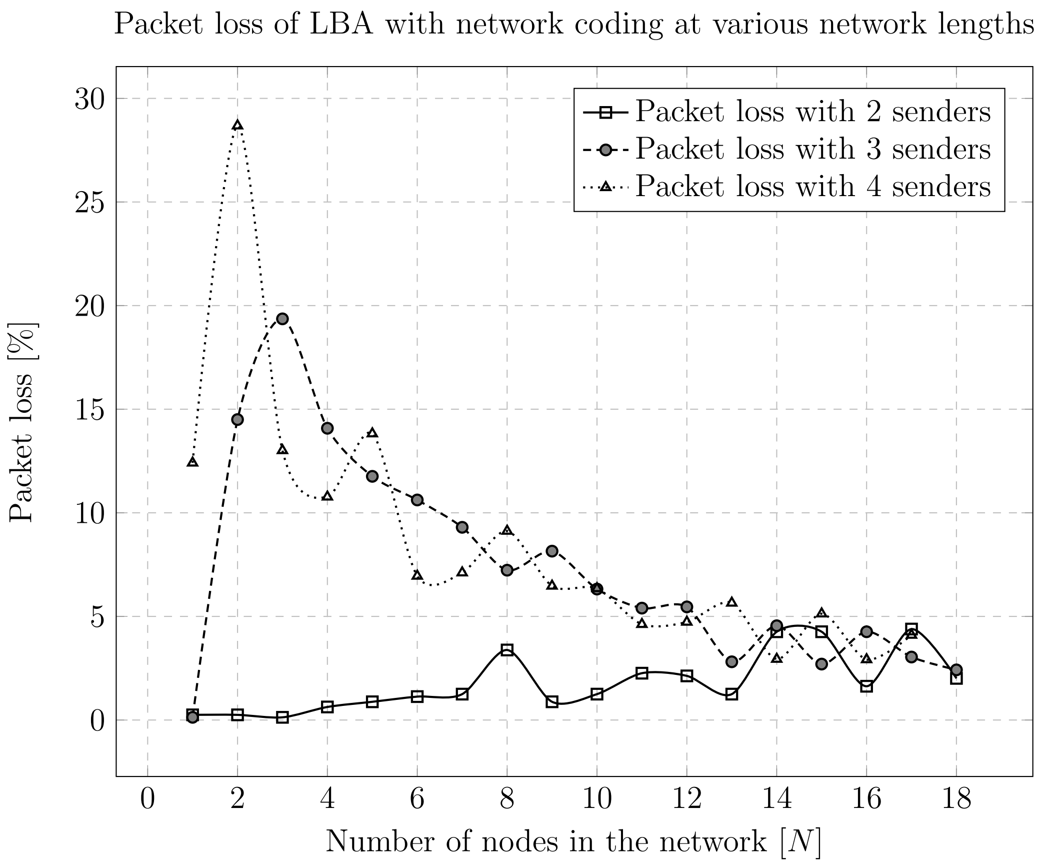

Figure 5.15: Packet loss for the multiple senders scenarios for the LBA with network coding simulations. Situations where there are a lot of sending nodes see high packet losses when the networks are too short. These high packet losses are attributed to the propagation time it takes for network coding to get enabled, as the packets need to migrate through intermediate nodes to allow for encoding to be implemented.

ally well. This would make sense since the probability of at least two packets being encoded together increases substantially because there are more packets moving down the network. With the min-cut being two, at the very least, only two packets need to be encoded in order to again the benefits. This means that when there are three simultaneous senders, then the chance for at least two packets being buffered and encoded at an intermediate node is much higher.

5.5.3 The theoretical gains vs simulation gains

Theorems (11) and (12) show that there are theoretical limits to how big the gains that network coding can promise. In Figure 5.16, the difference between the LBA and the LBA with network coding can clearly be seen to tend to the theoretical limit dictated by Theorem (11). Since both simulations are run in the same circumstances, it can clearly be seen that while the theoretical analysis might not be perfectly able to describe the systems, they still are forced to obey the laws that are present.

In Figure 5.17, the difference between the LBA with and without network coding and 3 simultaneous senders is being presented. This graph shows interesting results where by the theoretical limit is being exceeded at times. However, it can be seen that there is a trend which tends to the limit of 1.5, and the times where this limit is exceeded maybe times where the LBA algorithm is unable to keep up with the number of packets being sent. This means more packets are being sent that required due to the out of order nature of the packets. Network coding has a buffering system and congestion control built into it, were as these features need to be explicitly built into the LBA algorithm.

5.5.4 Scalability of LBA and network coding

From the aspect of scalability, network coding offers a substantially better mechanism for scalability than the algorithm with just LBA. This can be seen in the Figures 5.8, 5.9 and 5.10. The network coding algorithm behaved more as expected than the standard LBA algorithm. Furthermore, after four

The simulation results of the 2 senders tending to the theoretical limits.

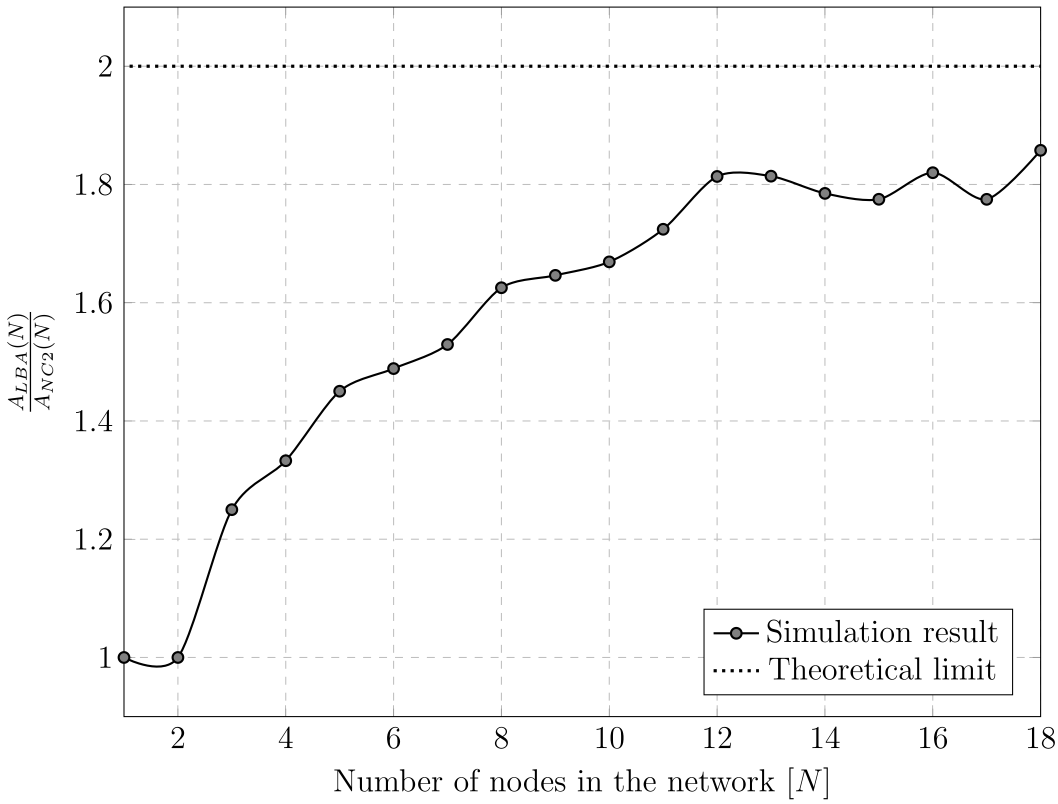

Figure 5.16: Graph showing how the simulation results tend to the theoretical limits that have been calculated in Theorem (11).

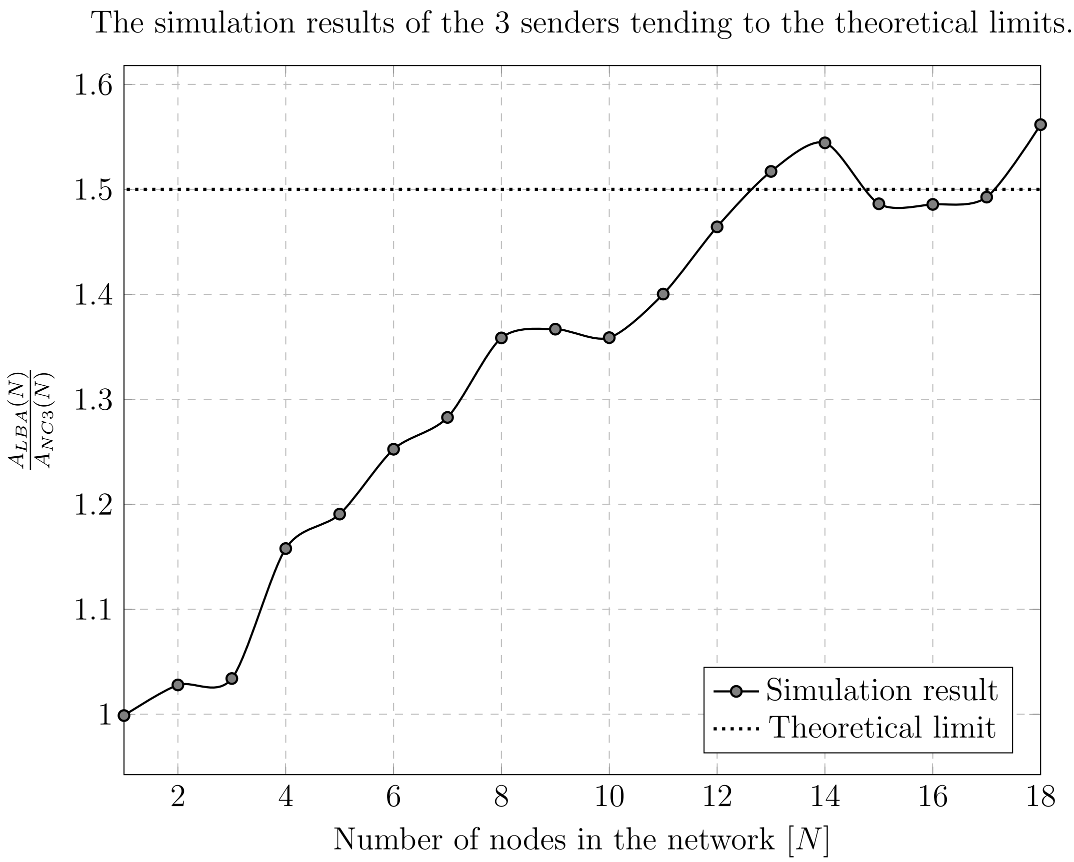

Figure 5.17: Graph showing how the simulation results tend to the theoretical limits that have been calculated in Theorem (12).

simultaneously sending nodes, the standard LBA protocol had so many packets being broadcast the simulation software would crash. While under the exact same scenario, the network coding based protocol was able to mitigate the barrage of packets moving through the network and encode them.

With network coding, the network does operate well even when the network is long. This proves to be a challenge with the standard LBA protocol as the simulation shows how it rapidly loses its linear growth in Figures 5.8, 5.9 and 5.10 when compared to the network coding protocol. From this, it is deduced that network coding adds stability and scalability to a well behaved network. With network coding, the network can maintain reliable transmissions, and with the benefits of reduced number of packets required to be sent.

While network coding is better suited to well behaved networks with low noise, the situation is different for when there is interference. Network coding adds improved immunity to low levels of noise given that there is some redundancy built into the encoded packets. This allows it to maintain a reliable network when the interfering node is only present in the network momentarily. Figure 5.12 shows how the network quickly loses reliability when the interference is too pronounced. This is a problem with having the packets encoded. If a node does not received enough encoded packets, then it cannot recover any of the packets encoded in them. This is why the 3 senders scenario does much worse because there are more packets encoded that are susceptible to being unable to decode. In Figure 5.13, the LBA has a worst case of 50% packet loss since the interfering packet is positions half way through the network. With network coding, it shows how even nodes near the interfering node are impacted.

From the analysis in Section 5.5.3, it can be seen that network coding does offer a systematic improvement over the standard LBA algorithm. Due to its better scalability, the network coding algorithm actually beats the theoretical limit for the case where there are 3 senders, but this is due to the superior congestion control.

5.5.5 Practical applications of LBA with network coding

Adding network coding to the LBA algorithm shows massive potential in improving the algorithms scalability and reliability in scenarios where the network has many senders, and when the network is very long. However, network coding is more susceptible to noise that the standard LBA algorithm due to the packets being encoded. Adding network coding creates an ideal scenario for sending messages reliably far back in a network. While standard LBA is better at transmitting information rapidly to nodes around the sending node, the addition of network coding improves the transmission of messages further down the network.

Network coding suits nodes informing other nodes further down the network. It is ideal for applications where a node needs to alert of events or road conditions to other nodes much further down the network. These scenarios require that there will be multiple sending nodes, since a lot of vehicles might be alerting of specific traffic or road conditions way back to the intersection of the road.

Furthermore, if vehicles were updating traffic lights or road side lights to turn on for the favour of the vehicle, network coding is definitely a more favoured approach. Since many vehicles will be transmitting at the same time, having the congestion management that network has natively is a massive benefit. Also, nodes will want to alert lights much further down the network.

Finally, network coding is exceptionally ill suited for applications where packets are destined to not travel far. The high packet losses seen in network coding is a serious concern for this. However, the standard LBA offers excellent response in this regard.

Chapter 6

CONCLUSION

6.1 Network Coding Potential

From the Chapter 5, the potential benefits and drawbacks of network coding become evident. From the analysis, it is seen that network coding improves the scalability of networking on an LBA, but at the cost of reliability for when there is high amount of interference. The theoretical gains were shown to be confirmed by the simulation results with the same general trends being followed.

With the addition of the ladder topology, modelling both the LBA and the LBA with network coding was achievable, but not without its limitations. The theoretical models that used the ladder topology were followed by the simulation results. The LBA models were described well by the models, with the only problems being that the theoretical models could not take packet loss due to too much interference into account.

From the results that have been gathered by the theoretical analysis and simulations, it is observed that network coding definitely adapts the way that networking over a VANET has to be perceived. With a system that has less overhead and offers more stability for longer networks, network coding does offer a lot of potential and it is an avenue for future research, if not development of a commercial product.

6.2 Future Work

As mentioned, network coding offers significant benefits over regular LBA, however, it is not without its own problems. The scenarios with interference and when the networks are very short, demonstrate that network coding algorithm could use more research to address these. While out of scope of this paper, the coupling of LBA and network coding can also be investigated in other networking fields, such as disaster management, or setting up temporary networks.

6.3 Recommendations

There were problems encountered while implementing network coding, LBA and running these over a simulator. The most notable problem faced, was a minor bug with the buffering system that did not empty the buffer correctly at a specific time. This problem resulted in packets not being encoded correctly and these errors propagated through to other packets and resulted in extremely high packets losses. Debugging these sort of problems is time consuming, and it is recommended to build a mechanism that allows for encoding problems to be detected and identified.

Furthermore, it is recommended to build the small parts separately after designing the entire system. The Gaussian Elimination algorithm is difficult to get right when directly interfacing with a finite field, thus, rather build an interface that allows the Gaussian Elimination algorithm to be run over normal integers, and then once that is proven to work over all cases then migrate over to finite fields.

6.4 Conclusion

The addition of network coding to LBA for VANET improves the performance of the network by reducing the number of redundant packets being sent, and this also improves the scalability for longer networks. This answered the main question proposed, however, in the process of answering this research question, further questions were answered. Network coding did provide on its promise of improved performance in the form of a reduced number of packets being sent as well as a more scalable network, it also showed that this came at the minor cost of susceptibility to interference.

The addition of the ladder topology made creating theoretical models possible. With both LBA and LBA with network coding algorithms behaving in ways that the models predicted, the ladder topology shows to be an appropriate topology to use. By using the theoretical models, theoretical performance maximums were also generated, showing the power of using the ladder topology for analysis.

With network coding, there are teething problems that need to be addressed. It is not as robust as just normal LBA, it requires a longer network for the packets to get encoded effectively. This process of it requiring a certain number of intermediate nodes to enable encoding made it worse than LBA for very short networks. However, this is a minor problem, and can be addressed by making a more intelligent network coding algorithm that is able to decide when it should encode packets.

Despite the one short fall of network coding, it has proven to be a massive improvement over just normal broadcasting. Observing the gains approach the theoretical limits of the network, network coding should match or out perform other traditional routing schemes, especially dynamic networks such as VANET.

With the benefits that network coding brings and the future work well known, the hope is that this research spurs on further interest and possibilities for coupling LBA with network coding to VANET and other networks. The results demonstrate that there is a lot of scope that can be pursued further.

REFERENCES

[1] J. A. Stankovic. "Wireless Sensor Networks." *Chapter in Handbook of Real-Time and Embedded Systems*, pp. 1–19, 2008. URL `http://www.cs.virginia.edu/~stankovic/psfiles/Kim_CL2004-0162.pdf`.

[2] F. L. LEWIS. "Wireless Sensor Networks." In *Smart Environments: Technology, Protocols, and Applications, Chapter 2*, pp. 1–18. 2005.

[3] G. M. A. C. Roberto Verdone, Davide Dardari. *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*. Academic Press, 2008.

[4] T. Arampatzis, J. Lygeros, and S. Manesis. "A survey of applications of wireless sensors and wireless sensor networks." In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pp. 719–724. IEEE, 2005.

[5] A. Z. Faza and S. Sedigh-Ali. "A general purpose framework for wireless sensor network applications." In *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International*, vol. 2, pp. 356–358. IEEE, 2006.

[6] L. M. Borges, F. J. Velez, and A. S. Lebres. "Survey on the characterization and classification of wireless sensor network applications." *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 4, pp. 1860–1890, 2014.

[7] X.-Y. Li, Y. Wang, and Y. Wang. "Complexity of data collection, aggregation, and selection for wireless sensor networks." *Computers, IEEE Transactions on*, vol. 60, no. 3, pp. 386–399, 2011.

[8] F. Wang and J. Liu. "Networked wireless sensor data collection: issues, challenges, and approaches." *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 4, pp. 673–687, 2011.

[9] Z. Ismail and R. Hassan. "A performance study of various mobility speed on AODV routing protocol in homogeneous and heterogeneous MANET." In *Communications (APCC), 2011 17th Asia-Pacific Conference on*, pp. 637–642. IEEE, 2011.

[10] S. A. Thorat and P. Kulkarni. "Design issues in trust based routing for MANET." In *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, pp. 1–7. IEEE, 2014.

[11] S. S. Anjum, R. M. Noor, and M. H. Anisi. "Survey on MANET based communication scenarios for search and rescue operations." In *IT Convergence and Security (ICITCS), 2015 5th International Conference on*, pp. 1–5. IEEE, 2015.

[12] M. Shimizu, H. Todoroki, and K. Takami. "MANET routing scheme for establishing a connection from an isolated LAN to the Internet using HTTP communication." In *Communications (APCC), 2013 19th Asia-Pacific Conference on*, pp. 751–756. IEEE, 2013.

[13] Ranjan, Prabhakar, Ahirwar, and K. Kant. "Comparative study of vanet and manet routing protocols." In *Proceedings of the International Conference on Advanced computing and communication Technologies (ACCT 2011)*. 2011.

[14] A. Rahman, W. Olesinski, and P. Gburzynski. "Controlled flooding in wireless ad-hoc networks." In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, pp. 73–78. IEEE, 2004.

[15] D. Sivakumar, B. Suseela, and R. Varadharajan. “A survey of routing algorithms for MANET.” In *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, pp. 625–640. IEEE, 2012.

[16] B. K. Chaurasia, R. S. Tomar, S. Verma, and G. S. Tomar. “Suitability of MANET routing protocols for vehicular ad hoc networks.” In *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*, pp. 334–338. IEEE, 2012.

[17] Z. J. Haas, J. Y. Halpern, and L. E. Li. “Gossip-Based Ad Hoc Routing.” *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 14, pp. 479–491, 2006.

[18] A. Bachir and A. Benslimane. “A Multicast Protocol in Ad hoc Networks Inter-Vehicle Geocast.” *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 4, pp. 2456 – 2460, 2012.

[19] M. Kihl, M. L. Sichitiu, and H. P. Joshi. “Design and Evaluation of two Geocast protocols for Vehicular Ad-hoc Networks.” *Journal of Internet Engineering*, vol. 4, pp. 2456 – 2460, 2008.

[20] D. K. Mishra. “Privacy Preservation in MANET: Issues and Challenges.” *Third International Conference on Intelligent Systems Modelling and Simulation*, 2012.

[21] M. Azarmi, M. Sabaei, and H. Pedram. “Adaptive Routing Protocols for Vehicular Ad hoc Networks.” *2008 Internatioal Symposium on Telecommunications*.

[22] R. Michoud, A. M. Orozco, and G. Llano. “Mobile ad-hoc routing protocols survey for the design of VANET applications.” In *Intelligent Transportation Systems Symposium (CITSS), 2012 IEEE Colombian*, pp. 1–6. IEEE, 2012.

[23] A. Dahiya and R. Chauhan. "A comparative study of MANET and VANET environment." *Journal of computing*, vol. 2, no. 7, pp. 87–92, 2010.

[24] S. B. Raut and L. Malik. "Survey on vehicle collision prediction in VANET." In *Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on*, pp. 1–5. IEEE, 2014.

[25] P. Nithya Darisini and N. S. Kumari. "A survey of routing protocols for VANET in urban scenarios." In *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, pp. 464–467. IEEE, 2013.

[26] S. K. Bhoi and P. M. Khilar. "Vehicular communication: a survey." *IET Networks, Volume 3, Issue 3,*, pp. 204–217, 2014.

[27] E. Schoch, F. Kargl, and M. Weber. "Communication Patterns in VANETs." *IEEE Communications Magazine, Volume 46 , Issue 11*, pp. 119–125, November 2008.

[28] Y.-B. Ko and N. H. Vaidya. "Geocasting in mobile ad hoc networks: Location-based multicast algorithms." In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pp. 101–110. IEEE, 1999.

[29] D. Tian, K. Shafiee, and V. Leung. "Position-based directional vehicular routing." In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6. IEEE, 2009.

[30] O. Kaiwartya and S. Kumar. "Geocast Routing: Recent Advances and Future Challenges in Vehicular Adhoc Networks." *International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 291–296, 2014.

[31] H. Ghafoor and K. Aziz. "Position-based and geocast routing protocols in VANETs." In *Emerging Technologies (ICET), 2011 7th International Conference on*, pp. 1–5. IEEE, 2011.

[32] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. "Network Information Flow." *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 46, pp. 1204–1216, 2000.

[33] C. Fragouli, J.-Y. L. Boudec, and J. Widmer. "Network Coding: An Instant Primer." *ACM SIGCOMM Computer Communication Review*, vol. 36, pp. 63–68, 2006.

[34] S. Katti, H. Rahul, W. Hu, D. Katabi, and M. Mdard. "XORs in the Air: Practical Wireless Network Coding." *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 16, pp. 497–510, 2008.

[35] R. W. Yeung. *Information Theory and Network Coding*. Springer US, 2007.

[36] C. Fragouli and E. Soljanin. "Network Coding Fundamentals." *Foundations and Trends in Networking*, vol. 2, pp. 1–133, 2007.

[37] L. R. Ford Jr and D. R. Fulkerson. *Flows in networks*. Princeton university press, 2015.

[38] C. Fragouli and E. Soljanin. "Network Coding Applications." *Foundations and Trends in Networking*, vol. 2, p. 135269, 2007.

[39] I.-H. Hou, Y.-E. Tsai, T. F. Abdelzaher, and I. Gupta. "AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks." *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 2189–2197, 2008.

[40] I. Salhi, Y. Ghamri-Doudane, S. Lohier, and E. Livolant. "CoZi: basic Coding for better Bandwidth Utilization in ZigBee Sensor Networks." *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1 – 6, 2010.

[41] Z. Yang, M. Li, and W. Lou. "R-Code: Network Coding based Reliable Broadcast in Wireless Mesh Networks with Unreliable Links." *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6, 2009.

[42] S. Ahmed and S. S. Kanhere. "VANETCODE: Network Coding to Enhance Cooperative Downloading in Vehicular Ad-Hoc Networks." *IWCMC '06*, pp. 527–532, 2006.

[43] M. H. Firooz. "Collaborative Downloading in VANET using Network Coding." *IEEE ICC 2012 - Wireless Communications Symposium*, pp. 4584–4588, 2012.

[44] M. Stehk. *Comparison of Simulators for Wireless Sensor Networks*. Ph.D. thesis, MASARYK UNIVERSITY FACULTY OF INFORMATICS, 2011.

[45] C. Fragouli and E. Soljanin. "Network Coding Fundamentals, chapter 2.1, Theorem 2.1." *Foundations and Trends in Networking*, vol. 2, p. 13, 2007.

[46] C. Fragouli and E. Soljanin. "Network Coding Fundamentals, chapter 2.1, Theorem 2.2." *Foundations and Trends in Networking*, vol. 2, pp. 1–133, 2007.

[47] S. Ebers, H. Hellbuck, D. Pfisterer, and S. Fischer. "Short paper: Collaboration between VANET applications based on open standards." In *Vehicular Networking Conference (VNC), 2013 IEEE*, pp. 174–177. IEEE, 2013.

[48] A. Boulis. *Castalia A simulator for Wireless Sensor Networks and Body Area Networks, version 3.2, User's Manual*. NICTA, 2011.

[49] J. S. Plank. "Galois.h, Galois.cc." URL `http://web.eecs.utk.edu/~plank/plank/papers/CS-07-593/`.

Appendix A

Simulation Software

This simulation software used is the Castalia framework built on top of OMNet++. Castalia was the most well documented simulation framework available, has an excellent layered approach to implementing new protocols, and the community is responsive.

A.1 Simulating with Castalia Framework

Implementing a new protocol with Castalia is not too difficult as all that is needed is creating a new folder inside the appropriate layer. So for routing, creating a new folder inside the communication and routing folders within the global folder called *'src'*. Then the software needs to implement the virtual functions and packets defined by that specific layer. With the aid of a script file that uses the *'.ned'* extension, which defines the inputs into this layer such as default variable values, or values that can be changed by the script that runs the simulator. The *'.ned'* file also defines the gates that this module will use, such as what other modules that it will be sending messages out to or receiving messages from.

The user manual[48] provides the fundamentals on how to implement the different layers, as well as how to implement certain operations. This manual is very comprehensive and provides, with great detail, everything that it will take to implement a successful layer into the simulator. However, some things are not covered in the user manual and a certain amount of tinkering is required in order to implement specific things. One such issue encountered is that the mobility module is not directly accessible to the routing layer; thus implementing location based addressing is impossible. This problem is mitigated by simply grabbing the module from the parent modules using the following line of code in the startup function:

```
mobilityModule = check_and_cast< VirtualMobilityManager*>
(getParentModule()->getParentModule()->getSubmodule“MobilityManager”));
```

With the routing layer set up, in order to run a simulation, a new folder under the directory ‘Simulations’ needs to be made, and a new file with the name *‘omnetpp.ini’* must be created. This file defines how the simulation will be carried out, and this file allows you to define how long the simulation must run for, the size of the field that the simulation will run on, the total number of nodes, the wireless channel specifics; what MAC, routing and application layer will be used, as well as set the variable for each of these layers.

A.2 Implementing Routing Layer into Castalia

Implementing the routing layer into Castalia involves writing the software that handles the messages to and from the application and MAC layers. In order to implement network coding into the routing layer, the messages from the application layer were needed to be unpacked, with the data and header information (such as destination and source location) being accessed so as to allow for all that information to be encoded into network coded packets. With network coding, there is an increased coupling between the application and routing layer since the routing layer needs to understand how to encode the packets.

Location based addressing adds more complexity to the implementation of the routing layer, as now decisions need to be made on whether or not to for-

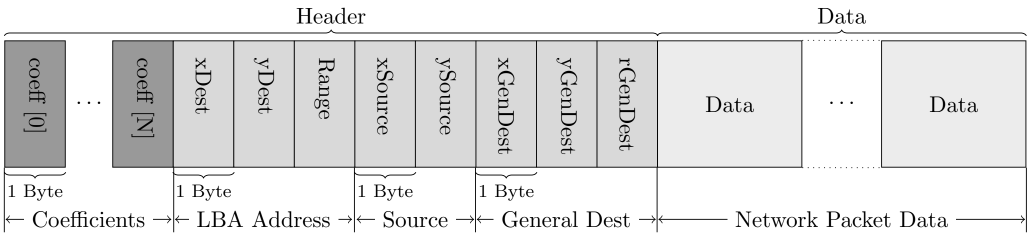

Figure A.1: Packet structure for a routing layer that uses both network coding and LBA

ward a packet, to encode a packet, to decode the received packets and move them to the application layer, or to simply drop a packet from the buffer if it is no longer valid in the network. This decision making coding that implements the location based addressing has to be integrated seamlessly into the network coding in order for the system to work.

The packets that were used in the routing layer are shown in Figure A.1. The packets design shows that the header has space for the coefficients, the x and y coordinates for the source and destination nodes, the range around the x and y coordinate for the destination node, as well as the *GenDest* x and y coordinates and the range around the *GenDest*. The *GenDest* (short for General Destination) is used to allow for packets that do not share the same destination, but can still be encoded with network coding. It is a simple scheme to improve the number of network coding opportunities.

A.3 Collecting Results from Castalia

Collection results from Castalia is a simple process if it is done correctly. The simulation software allows you to declare an output by calling the function *declareOutput ("Output Name")* in the startup function of the layer your are implementing. Once this function has been called, the simulator will create

a new label within the simulator's logging file which keeps track of the values stored. Once the output has been declared, the programmer just needs to call the function *collectOutput("Output Name")* with an output name that matches the declared output name. If the function *collectOutput("")* is called with only the output name, then it will increment that output by one every time the function is called. This is useful for keeping track of discrete events such as the number of packets sent or received. Users can also increase the output in different ways, by using the following formatting:
collectOutput("output name", index, "label", amountIncreased).

For this simulation, the number of packets sent, received and received correctly were mainly kept track of, thus the collected outputs are very simple integers. The application layer tracks how many packets it has sent to the routing layer and how many were correctly received, the routing layer detects how many packets it sends to the MAC layer and how many were received from the MAC layer. This allows for the number of successful packets to be tracked and the number of packets that have been generated, from this the comparison between what the models predict and what the simulations generate can be made. The simulation software also includes standard recorded data that is included with every simulation such as energy usage, but most of this is not used in this research.

There are a lot of different scenarios that are being investigated with this simulation software, and so it was decided to organise each different scenario into one separate file. This means that each and every simulation that is run has a different file detailing its results, and the naming scheme of the file identifies which simulation was run. This method also allows for the backing up of all the simulation results to be seamless and easy to recover, and finally it makes it simpler to move the data over to an online spreadsheet.

All the results are gathered into a spreadsheet so that they can easily be processed and the predicted performance from the model can be easily gathered. Since the simulations were done in Linux and the rest of the work was completed in Windows, online spreadsheets were used for smooth integration with the plotting being done by using the LaTeX package 'tikz'.

Appendix B

Tables of results

Table B.1: Results of the scenario where there is only one sending node and only LBA algorithm is being used.

Number of Nodes	Model prediction	Simulation	Corrected model
1	1.00	1.00	0.50
2	1.00	1.00	0.70
3	1.67	1.67	1.57
4	2.00	2.00	2.10
5	2.60	2.71	2.90
6	3.00	3.18	3.50
7	3.57	3.91	4.27
8	4.00	4.47	4.90
9	4.56	5.35	5.66
10	5.00	5.97	6.30
11	5.55	6.62	7.05
12	6.00	7.29	7.70
13	6.54	8.35	8.44
14	7.00	8.99	9.10
15	7.53	9.99	9.83
16	8.00	10.82	10.50
17	8.53	11.85	11.23
18	9.00	12.85	11.90

Table B.2: Results of the scenario where there are two sending nodes and only LBA algorithm is being used.

Number of Nodes	Model prediction	Simulation	Corrected model
1	1.00	1.00	0.48
2	1.00	1.00	0.66
3	1.67	1.67	1.51
4	2.00	2.00	2.03
5	2.60	2.69	2.81
6	3.00	3.11	3.39
7	3.57	3.82	4.14
8	4.00	4.44	4.75
9	4.56	5.06	5.49
10	5.00	5.75	6.12
11	5.55	6.54	6.85
12	6.00	7.45	7.48
13	6.54	8.31	8.20
14	7.00	8.76	8.85
15	7.53	9.73	9.56
16	8.00	10.65	10.21
17	8.53	11.49	10.92
18	9.00	12.50	11.57

Table B.3: Results of the scenario where there are three sending nodes and only LBA algorithm is being used.

Number of Nodes	Model prediction	Simulation	Corrected model
1	1.00	1.26	1.14
2	1.00	1.64	1.29
3	1.67	1.81	2.10
4	2.00	2.36	2.57
5	2.60	2.72	3.31
6	3.00	3.40	3.86
7	3.57	3.80	4.57
8	4.00	4.59	5.14
9	4.56	5.11	5.84
10	5.00	5.99	6.43
11	5.55	6.45	7.12
12	6.00	7.85	7.71
13	6.54	8.11	8.40
14	7.00	9.77	9.00
15	7.53	10.92	9.68
16	8.00	12.68	10.29
17	8.53	13.64	10.96
18	9.00	16.69	11.57

Table B.4: Results of the scenario where there are four sending nodes and only LBA algorithm is being used.

Number of Nodes	Model prediction	Simulation
1	1.00	1.80
2	1.00	1.84
3	1.67	2.54
4	2.00	3.00
5	2.60	3.91
6	3.00	4.53
7	3.57	5.68

Table B.5: Comparison of the results from the different scenarios where only LBA is being used.

Number of Nodes	One Sender	Two Senders	Three Senders	Four Senders
1	1.00	1.00	1.26	1.80
2	1.00	1.00	1.64	1.84
3	1.67	1.67	1.81	2.54
4	2.00	2.00	2.36	3.00
5	2.71	2.69	2.72	3.91
6	3.18	3.11	3.40	4.53
7	3.91	3.82	3.80	5.68
8	4.47	4.44	4.59	N/A
9	5.35	5.06	5.11	N/A
10	5.97	5.75	5.99	N/A
11	6.62	6.54	6.45	N/A
12	7.29	7.45	7.85	N/A
13	8.35	8.31	8.11	N/A
14	8.99	8.76	9.77	N/A
15	9.99	9.73	10.92	N/A
16	10.82	10.65	12.68	N/A
17	11.85	11.49	13.64	N/A
18	12.85	12.50	16.69	N/A

Table B.6: Results from the interference patterns with just LBA.

Distance travelled by interference node	2 Sending nodes	3 Sending Nodes
500	97.87	69.92
400	97.99	71.26
300	93.36	70.43
200	93.73	70.76
100	72.06	63.16
50	54.51	53.97

Table B.7: Packet loss percentage of the different scenarios that LBA was tested under in %. The percentage is for a given network length.

Number of Nodes	**One Sender**	**Two Senders**	**Three Senders**
1	0.00	0.25	0.13
2	0.00	0.25	0.21
3	0.25	0.00	0.60
4	0.00	0.00	0.68
5	0.25	0.25	0.75
6	0.25	0.25	0.92
7	0.25	0.00	0.46
8	0.00	0.00	0.72
9	0.00	0.25	0.98
10	0.00	0.25	5.61
11	0.25	0.00	2.39
12	0.00	0.13	6.61
13	0.00	0.25	4.04
14	0.00	0.00	11.64
15	0.00	0.13	17.18
16	0.00	0.13	22.45
17	0.00	0.00	21.98
18	0.00	0.00	31.43

Table B.8: Results of the scenario where there are two sending nodes and LBA algorithm with network coding is being used.

Number of Nodes	**Model prediction**	**Simulation**	**Corrected model**
1	1.00	1.00	0.79
2	1.00	1.00	0.88
3	1.33	1.33	1.31
4	1.50	1.50	1.56
5	1.80	1.85	1.95
6	2.00	2.09	2.25
7	2.29	2.49	2.62
8	2.50	2.73	2.93
9	2.78	3.08	3.30
10	3.00	3.44	3.61
11	3.27	3.79	3.97
12	3.50	4.11	4.29
13	3.77	4.58	4.65
14	4.00	4.91	4.97
15	4.27	5.48	5.33
16	4.50	5.85	5.65
17	4.76	6.47	6.01
18	5.00	6.73	6.34

Table B.9: Results of the scenario where there are three sending nodes and LBA algorithm with network coding is being used.

Number of Nodes	**Model prediction**	**Simulation**
1	1.67	1.26
2	1.67	1.59
3	2.11	1.74
4	2.33	2.02
5	2.73	2.27
6	3.00	2.69
7	3.38	2.95
8	3.67	3.36
9	4.04	3.70
10	4.33	4.16
11	4.70	4.50
12	5.00	5.01
13	5.36	5.13
14	5.67	5.59
15	6.02	6.08
16	6.33	6.62
17	6.69	7.13
18	7.00	7.33

Table B.10: Results of the scenario where there are four sending nodes and LBA algorithm with network coding is being used.

Number of Nodes	**Model prediction**	**Simulation**	**Corrected Model**
1	1.50	1.78	1.25
2	1.50	2.09	1.50
3	1.83	2.13	2.08
4	2.00	2.31	2.50
5	2.30	2.96	3.05
6	2.50	3.12	3.50
7	2.79	3.67	4.04
8	3.00	4.22	4.50
9	3.28	4.75	5.03
10	3.50	5.27	5.50
11	3.77	5.79	6.02
12	4.00	6.35	6.50
13	4.27	7.20	7.02
14	4.50	7.71	7.50
15	4.77	8.47	8.02
16	5.00	9.01	8.50
17	5.26	9.98	9.01

Table B.11: The comparison of all the results of the different scenarios where LBA and network coding are used.

Number of Nodes	2 Senders	3 Senders	4 Senders
1	1.00	1.26	1.78
2	1.00	1.59	2.09
3	1.33	1.74	2.13
4	1.50	2.02	2.31
5	1.85	2.27	2.96
6	2.09	2.69	3.12
7	2.49	2.95	3.67
8	2.73	3.36	4.22
9	3.08	3.70	4.75
10	3.44	4.16	5.27
11	3.79	4.50	5.79
12	4.11	5.01	6.35
13	4.58	5.13	7.20
14	4.91	5.59	7.71
15	5.48	6.08	8.47
16	5.85	6.62	9.01
17	6.47	7.13	9.98
18	6.73	7.33	N/A

Table B.12: Comparison of scenario with two sending nodes with and without network coding.

Number of Nodes	No Network Coding	With Network Coding
1	1.00	1.00
2	1.00	1.00
3	1.67	1.33
4	2.00	1.50
5	2.69	1.85
6	3.11	2.09
7	3.82	2.49
8	4.44	2.73
9	5.06	3.08
10	5.75	3.44
11	6.54	3.79
12	7.45	4.11
13	8.31	4.58
14	8.76	4.91
15	9.73	5.48
16	10.65	5.85
17	11.49	6.47
18	12.50	6.73

Table B.13: Comparison of scenario with three sending nodes with and without network coding.

Number of Nodes	No Network Coding	With Network Coding
1	1.26	1.26
2	1.64	1.36
3	1.81	1.40
4	2.36	1.74
5	2.72	2.00
6	3.40	2.40
7	3.80	2.67
8	4.59	3.11
9	5.11	3.40
10	5.99	3.90
11	6.45	4.26
12	7.85	4.73
13	8.11	4.99
14	9.77	5.34
15	10.92	5.92
16	12.68	6.34
17	13.64	6.91
18	16.69	7.15

Table B.14: Comparison of scenario with four sending nodes with and without network coding Note that the scenario with no network coding could not run for networks longer than 7 nodes.

Number of Nodes	**No Network Coding**	**With Network Coding**
1	1.80	1.78
2	1.84	2.09
3	2.54	2.13
4	3.00	2.31
5	3.91	2.96
6	4.53	3.12
7	5.68	3.67
8	N/A	4.22
9	N/A	4.75
10	N/A	5.27
11	N/A	5.79
12	N/A	6.35
13	N/A	7.20
14	N/A	7.71
15	N/A	8.47
16	N/A	9.01
17	N/A	9.98

Table B.15: Results from the interference patterns for LBA with network coding.

Distance travelled by interference node	**2 Sending nodes**	**3 Sending Nodes**
500	96.49	88.22
400	94.24	89.64
300	91.98	88.72
200	93.73	81.62
100	49.87	35.42
50	35.09	19.63

Table B.16: Results from the interference patterns for LBA with and without network coding for the scenario with only two senders

Distance travelled by interference node	**Without Network Coding**	**With Network Coding**
500	97.87	96.49
400	97.99	94.24
300	93.36	91.98
200	93.73	93.73
100	72.06	49.87
50	54.51	35.09

Table B.17: Results from the interference patterns for LBA with and without network coding for the scenario with only three senders

Distance travelled by interference node	**Without Network Coding**	**With Network Coding**
500	69.92	88.22
400	71.26	89.64
300	70.43	88.72
200	70.76	81.62
100	63.16	35.42
50	53.97	19.63

Table B.18: The comparison of all the different packet losses of different scenarios. The packet loss % is given for a specific network length.

Number of Nodes	**2 Senders %**	**3 Senders %**	**4 Senders%**
1	0.25	0.13	12.41
2	0.25	14.50	28.68
3	0.13	19.36	13.01
4	0.63	14.08	10.77
5	0.88	11.76	13.82
6	1.13	10.62	6.95
7	1.25	9.30	7.11
8	3.38	7.23	9.12
9	0.88	8.15	6.47
10	1.25	6.32	6.35
11	2.26	5.40	4.62
12	2.13	5.46	4.73
13	1.25	2.81	5.65
14	4.26	4.55	2.94
15	4.26	2.70	5.13
16	1.63	4.26	2.92
17	4.39	3.04	4.10
18	2.01	2.42	N/A

Table B.19: The theoretical limit of network coding's gains over the vanilla LBA algorithm with two sending nodes.

Number of Nodes	**Simulation results**	**Theoretical Gain**
1	1.0	2
2	1.0	2
3	1.25	2
4	1.33277592	2
5	1.45030426	2
6	1.488609113	2
7	1.52938222	2
8	1.625344353	2
9	1.646291769	2
10	1.669093557	2
11	1.724240423	2
12	1.813548978	2
13	1.814055237	2
14	1.784929757	2
15	1.774982849	2
16	1.819914347	2
17	1.774874177	2
18	1.857754608	2

Table B.20: The theoretical limit of network coding's gains over the vanilla LBA algorithm with three sending nodes.

Number of Nodes	Simulation result	Theoretical Gain
1	0.998745295	1.5
2	1.027884539	1.5
3	1.034018962	1.5
4	1.157857228	1.5
5	1.190649628	1.5
6	1.252438106	1.5
7	1.282753292	1.5
8	1.358484838	1.5
9	1.366802356	1.5
10	1.358732463	1.5
11	1.400248096	1.5
12	1.464265508	1.5
13	1.517061853	1.5
14	1.54430192	1.5
15	1.486255622	1.5
16	1.485667538	1.5
17	1.492598298	1.5
18	1.561637442	1.5

Appendix C

Proofs

Proof. To derive the formula seen in Theorem 7, the scenario for three senders is considered. Table C.1 contains the results generated from analysing Figures 5.3.

Table C.1: The total number of packets sent per sending node during to reach a node in a LBA network using network coding and the ladder topology for the scenario where there are two sending nodes.

Node	2 Senders
1	$1.\dot{6}$
2	$1.\dot{6}$
3	3
4	3
5	$4.\dot{3}$
6	$4.\dot{3}$
$\vdots$	$\vdots$
N	$\begin{cases} \frac{2(N+1)+1}{3} & \text{if } N \text{ is odd} \\ \frac{2N+1}{3} & \text{if } N \text{ is even} \end{cases}$

From Table C.1, the total sends required for a packet to reach a specific node in the network can be considered below.

The proposed formula that will satisfy this behaviour can be seen in Equation (5.7), to show this formula works for all scenarios, the odd and even value of N are considered differently.

When N is odd then:

If $S(N)$ is valid, then $S(N+1)$ must be equal to $S(N)$, given the ceiling function, and $S(N+2)$ must be equal to $S(N)+\frac{4}{3}$. The $\frac{4}{3}$ term can intuitively be seen in Table C.1, where in increments in steps of $1.33\dot{3}$. First, assuming that $S(N)$ is correct, then $S(N+1) = \frac{1}{3}\left(\lceil\frac{N+1}{2}\rceil \times 4 + 1\right)$

But N is odd, and the ceiling function will effectively add one onto the odd number before it is divided by two so as to create the correct value. Because of this, the $S(N)$ term must equal $S(N+1)$ for odd N values. For the $S(N+2)$ term, this can be written as: $S(N+2) = \frac{1}{3}\left(\lceil\frac{N+2}{2}\rceil \times 4 + 1\right)$

However, $N+2$ is odd because N is odd, and the ceiling function to round up, this becomes: $\frac{1}{3}\left(\lceil\frac{N+2}{2}\rceil \times 4 + 1\right) = \frac{1}{3}\left(2\left(N+3\right)+1\right)$

After some manipulation, $\frac{1}{3}\left(2\left(N+3\right)+1\right) = \frac{1}{3}\left(2\left(N+1\right)+1\right) + \frac{4}{3}$

Which breaks down to mean that $S(N+2) = S(N) + \frac{4}{3}$.

Thus, proven for odd N values.

For even values of N, $S(N+1)$ must equal $S(N) + \frac{4}{3}$. Starting with the basics, $S(N+1) = \frac{1}{3}\left(\lceil\frac{N+1}{2}\rceil \times 4 + 1\right)$. However, N is even, thus, $N+1$ is odd. This means that the term: $\lceil\frac{N+1}{2}\rceil = \left(\frac{N+2}{2}\right)$.

Thus, from this, $S(N+1) = \frac{1}{3}\left(\left(4\frac{N+2}{2}\right)+1\right)$ is derived. After some manipulation: $\frac{1}{3}\left(\left(\frac{N+2}{2}\right)+1\right) = \frac{1}{3}\left(4\left(\frac{N+1}{2}\right)+5\right)$. This breaks down into Equation (C.1).

$$S(N+1) = \frac{1}{3}\left(4\left(\frac{N+1}{2}\right)+1\right)+\frac{4}{3} \tag{C.1}$$

As expected.

□

Proof. Provided Theorem 8, the average number of sends required is simply the summation of the sends required for a packet to reach a node divided by the number of nodes that are in the network. This trend can be seen in Table C.2.

The formula for this behaviour seen in Equation (5.8) from Theorem 8 is derived below.

Table C.2: The average number of sends required to send packets to a network of length N, provided that all nodes are uniformly addressed

Number of nodes	Average number of sends required
1	$\frac{1.\dot{6}}{1} = 1.\dot{6}$
2	$\frac{1.\dot{6}+1.\dot{6}}{2} = 1.\dot{6}$
3	$\frac{1.\dot{6}+1.\dot{6}+3}{3} = 2.\dot{1}$
4	$\frac{1.\dot{6}+1.\dot{6}+3+3}{4} = 2.\dot{3}$
$\vdots$	$\vdots$
N	$\frac{S(1)+S(2)+S(3)+\cdots+S(N)}{N} = \sum_{i=1}^{N} \frac{S(i)}{N}$

With $S_{NC3}(N) = \frac{1}{3}\left(\left\lceil \frac{N}{2} \right\rceil 4 + 1\right)$, the average can be expressed as: $\frac{1}{3N}\sum_{i=1}^{N}\left(\left\lceil \frac{N}{2} \right\rceil 4 + 1\right)$

To find the closed form solution of this summation, the conditions for where N is even and odd need to be considered separately. Furthermore, the identity where $\sum_{i=1}^{N} i = \frac{N(N+1)}{2}$ will be used.

For the cases where N is even, then the summation will break down into two duplicated summations. Thus we can express it as:

$$\frac{1}{3N}\sum_{i=1}^{N}\left(\left\lceil \frac{N}{2} \right\rceil 4 + 1\right) = \frac{2}{3N}\sum_{i=1}^{\frac{N}{2}}(4i+1)$$

It is intuitive that Equation (C.2) is true.

$$\sum_{i=1}^{N}(4i+1) = (N+3)N \tag{C.2}$$

From Equation (C.2), the closed form solution for even N values can be manipulated to form: $\frac{2}{3N}\sum_{i=1}^{\frac{N}{2}}(4i+1) = \frac{(N+3)}{3}$

However, this can be written into a more general form and the final equation can be written as in Equation (C.3).

$$S_{NC3}(N_{even}) = \frac{1}{3}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{2}{N} + 3\right) \tag{C.3}$$

For the case where N is odd, there are effectively two summations, one that ends at $\frac{N-1}{2}$ and another that ends at $\frac{N+1}{2}$. Thus the expression can be written:

$$\frac{1}{3N}\sum_{i=1}^{N}\left(\left\lceil \frac{N}{2} \right\rceil 4 + 1\right) = \frac{1}{3N}\left(\sum_{i=1}^{\frac{N-1}{2}}(4i+1) + \sum_{i=1}^{\frac{N+1}{2}}(4i+1)\right)$$

Using the identity seen in Equation (C.2), but instead of N, the values $\frac{N-1}{2}$ and $\frac{N+1}{2}$ are used instead, this generates:

$$\frac{1}{3N}\left(\sum_{i=1}^{\frac{N-1}{2}}(4i+1) + \sum_{i=1}^{\frac{N+1}{2}}(4i+1)\right) = \frac{1}{3N}\left((N+2)\left(\frac{N-1}{2}\right) + (N+4)\left(\frac{N+1}{2}\right)\right)$$

After some manipulation, this simplifies to:

$$\frac{1}{3N}\left((N+2)\left(\frac{N-1}{2}\right) + (N+4)\left(\frac{N+1}{2}\right)\right) = \frac{1}{3N}\left(N^2 + 3N + 1\right)$$

This formula can be written into a form that is more general as Equation (C.4).

$$S_{NC3}(N_{odd}) = \frac{1}{3}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{2}{N} + 3\right) \tag{C.4}$$

Which is the same as the even case, Equation (C.3). Thus for all N, the formula holds as shown in Theorem 8.

□

Proof. To derive the formula seen in Theorem 9, the scenario for four senders is considered. Table C.3 contains the results generated from analysing Figure 5.5

Table C.3: The total number of packets sent per sending node during to reach a node in a LBA network using network coding and the ladder topology for the scenario where there are four sending nodes.

Node	2 Senders
1	1.5
2	1.5
3	2.5
4	2.5
5	3.5
6	3.5
$\vdots$	$\vdots$
N	$\begin{cases} \frac{N+1}{2} + 0.5 & \text{if } N \text{ is odd} \\ \frac{N}{2} + 0.5 & \text{if } N \text{ is even} \end{cases}$

From Table C.3, the total sends required for a packet to reach a specific node in the network can be considered below.

The proposed formula that will satisfy this behaviour can be seen in Equation(5.9), to show this formula works for all scenarios, the odd and even value of N are considered differently.

When N is odd then:

If $S(N)$ is valid, then $S(N+1)$ must be equal to $S(N)$, given the ceiling function, and $S(N+2)$ must be equal to $S(N)+1$. The 0.5 term can intuitively be seen in Table C.3, where in increments in steps of 1. First, assuming that $S(N)$ is correct, then the equation $S(N+1) = \lceil \frac{N+1}{2} \rceil + 0.5$ is proposed. But N is odd, and the ceiling function will effectively add one onto the odd number before it is divided by 2 so as to create the correct value. Because of this, the $S(N)$ term must equal $S(N+1)$ for odd N values. For the $S(N+2)$ function, $S(N+2) = \lceil \frac{N+2}{2} \rceil + 0.5$. However, $N+2$ is odd because N is odd, and the ceiling function to round up, this becomes:

$\lceil\frac{N+2}{2}\rceil + 0.5 = \frac{N+3}{2} + 0.5$. After some manipulation, it can be seen that: $\frac{N+3}{2} + 0.5 = \frac{N+1}{2} + 1.5$, which means that $S(N+2) = S(N) + 1$. Thus, proven for odd N values.

For even values of N, $S(N+1)$ must equal $S(N)+1$. Starting off, $S(N+1) = \lceil\frac{N+1}{2}\rceil + 0.5$. However, N is even, thus, $N+1$ is odd. This means that the term: $\lceil\frac{N+1}{2}\rceil = \left(\frac{N+2}{2}\right)$. Thus, this means that for an even N:

$$S(N+1) = \frac{N+2}{2} + 0.5$$

After some manipulation:

$$S(N+1) = \frac{N}{2} + 1.5$$

And this breaks down into:

$$S(N+1) = S(N) + 1$$

As expected.

□

Proof. Provided Theorem 10, the average number of sends required is simply the summation of the sends required for a packet to reach a node divided by the number of nodes that are in the network. This trend can be seen in Table C.4.

The general formula for this behaviour seen in Equation (5.10) from Theorem 10 can be easily derived.

With $S_{NC4}(N) = \lceil\frac{N}{2}\rceil + 0.5$, the average can be expressed as:

$$A_{NC4}(N) = \frac{1}{N}\sum_{i=1}^{N}\left(\left\lceil\frac{N}{2}\right\rceil + 0.5\right)$$

Table C.4: The average number of sends required to send packets to a network of length N, provided that all nodes are uniformly addressed

Number of nodes	Average number of sends required
1	$\frac{1.5}{1} = 1.5$
2	$\frac{1.5+1.5}{2} = 1.5$
3	$\frac{1.5+1.5+2.5}{3} = 1.8\dot{3}$
4	$\frac{1.5+1.5+2.5+2.5}{4} = 2$
$\vdots$	$\vdots$
N	$\frac{S(1)+S(2)+S(3)+\cdots+S(N)}{N} = \sum_{i=1}^{N} \frac{S(i)}{N}$

To find the closed form solution of this summation, the conditions for where N is even and odd need to be considered separately. Furthermore, the identity where $\sum_{i=1}^{N} i = \frac{N(N+1)}{2}$ will be used.

For the cases where N is even, then the summation will break down into two duplicated summations. Thus we can express it as:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\left\lceil\frac{N}{2}\right\rceil + 0.5\right) = \frac{2}{N}\sum_{i=1}^{\frac{N}{2}}(i + 0.5)$$

It can be shown that Equation (C.5) is valid.

$$\sum_{i=1}^{N}(i + 0.5) = \frac{1}{2}N\left(N + 2\right); \tag{C.5}$$

From Equation (C.5), the closed form solution for even N can be derived to to following:

$$\frac{2}{N}\sum_{i=1}^{\frac{N}{2}}(i + 0.5) = \frac{N}{4} + 1$$

However, this can be written into a form that is more general:

$$\frac{N}{4}+1=\frac{1}{2}\left(\left\lceil\frac{N^2}{2}\right\rceil\frac{1}{N}+2\right)$$

For the case where N is odd, there are effectively two summations, one that ends at $\frac{N-1}{2}$ and another that ends at $\frac{N+1}{2}$. Thus the expression can be written:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\left\lceil\frac{N}{2}\right\rceil+0.5\right)=\frac{1}{N}\left(\sum_{i=1}^{\frac{N-1}{2}}(i+0.5)+\sum_{i=1}^{\frac{N+1}{2}}(i+0.5)\right)$$

Using the identity seen in Equation (C.5), but instead of N, the values $\frac{N-1}{2}$ and $\frac{N+1}{2}$ are used instead, this generates the following:

$$\frac{1}{N}\left(\sum_{i=1}^{\frac{N-1}{2}}(i+0.5)+\sum_{i=1}^{\frac{N+1}{2}}(i+0.5)\right)=\frac{1}{2N}\left(\left(\frac{N-1}{2}+2\right)\left(\frac{N-1}{2}\right)+\left(\frac{N+1}{2}+2\right)\left(\frac{N+1}{2}\right)\right)$$

After some manipulation, this simplifies to:

$$\frac{1}{N}\left(\frac{1}{2}\left(\frac{N-1}{2}+2\right)\left(\frac{N-1}{2}\right)+\frac{1}{2}\left(\frac{N+1}{2}+2\right)\left(\frac{N+1}{2}\right)\right)=\frac{1}{N}\left(\frac{N^2+1}{4}+N\right)$$

This formula can be written into a form that is more general as:

$$\frac{1}{N}\left(\frac{N^2+1}{4}+N\right)=\frac{1}{2}\left(\left\lceil\frac{N^2}{2}\right\rceil\frac{1}{N}+2\right)$$

and following this, we can finally say that:

$$A_{NC4}(N)=\frac{1}{2}\left(\left\lceil\frac{N^2}{2}\right\rceil\frac{1}{N}+2\right)$$

Which is of the form expressed in Theorem 10, and thus the derivation is complete.

□

Proof. To prove the Theorem 11, we start with taking the limits of the left hand side

$$\lim_{N\to\infty} \frac{A_{LBA}(N)}{A_{NC2}(N)} = \lim_{N\to\infty} \frac{\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N}}{\frac{1}{2}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N} + 1\right)}$$

From this, the $\frac{1}{N}$ and $\frac{1}{2}$ can be pulled out of the equation giving:

$$= \lim_{N\to\infty} 2\left(\frac{\frac{1}{N}}{\frac{1}{N}}\right) \frac{\left\lceil \frac{N^2}{2} \right\rceil}{\left(\left\lceil \frac{N^2}{2} \right\rceil + N\right)}$$

with the $\frac{1}{N}$ terms cancelling, the $\left\lceil \frac{N^2}{2} \right\rceil$ can now be pulled out and cancelled, giving:

$$= \lim_{N\to\infty} \frac{2}{\left(1 + \frac{N}{\left\lceil \frac{N^2}{2} \right\rceil}\right)}$$

From this, we can clearly see that:

$$\lim_{N\to\infty} \frac{N}{\left\lceil \frac{N^2}{2} \right\rceil} = 0$$

And from this, we no longer have any N terms, allowing the limits to be dropped and leaving the answer being:

$$\lim_{N\to\infty} \frac{2}{1} = 2$$

□

Proof. To prove the Theorem 12, we start with taking the limits of the left hand side

$$\lim_{N\to\infty} \frac{A_{LBA}(N)}{A_{NC3}(N)} = \lim_{N\to\infty} \frac{\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N}}{\frac{1}{3}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{2}{N} + 3\right)}$$

The $\frac{1}{N}$, $\frac{1}{3}$ and 2 can term can be pulled out providing:

$$\lim_{N\to\infty} \frac{\left\lceil \frac{N^2}{2} \right\rceil \frac{1}{N}}{\frac{1}{3}\left(\left\lceil \frac{N^2}{2} \right\rceil \frac{2}{N} + 3\right)} = \lim_{N\to\infty} \frac{3}{2} \frac{\left\lceil \frac{N^2}{2} \right\rceil}{\left(\left\lceil \frac{N^2}{2} \right\rceil + \frac{3N}{2}\right)}$$

$$\lim_{N\to\infty} \frac{3}{2} \frac{\left\lceil \frac{N^2}{2} \right\rceil}{\left(\left\lceil \frac{N^2}{2} \right\rceil + \frac{3N}{2}\right)} = \lim_{N\to\infty} \frac{3}{2} \frac{1}{\left(1 + \frac{3N}{2\left\lceil \frac{N^2}{2} \right\rceil}\right)}$$

Given that

$$\lim_{N\to\infty} \frac{3N}{\left\lceil \frac{2N^2}{2} \right\rceil} = 0$$

It becomes clear that:

$$\lim_{N\to\infty} \frac{3}{2} \frac{1}{\left(1 + \frac{3N}{2\left\lceil \frac{N^2}{2} \right\rceil}\right)} = \lim_{N\to\infty} \frac{3}{2}$$

□

Appendix D

Test Bed

In order to test the theoretical work and run the simulations, a test bed is required to run the simulations. Because of the niche topic that is being investigated, a lot of the test bed was developed privately. This section serves as a description of the test bed developed and used to carry out the experiments.

D.1 Castalia Simulator

The simulation software used to build the test bed on is the Castalia framework, which itself is built on top of OMNet++. The Castalia framework offers significant advantages over other simulators in that it is an easy to use, flexible framework for WSN. Furthermore, the documentation and community support for the framework is excellent and this proved extremely helpful throughput development.

With location based addressing being a core component of the research, the Castalia framework did not explicitly have support for giving location data to the routing layer. Rather, a node's position was only accessible to the Application layer. However, this problem was circumvented by changing some of the source code within the framework to add the required modules to the routing layer. This flexibility saved a lot of time in development and the documentation make finding a solution to problems quick to find.

The Castalia framework uses layers in order to separate the implementation, with the development done in the routing Layer and the application layer. The routing layer was used to implement LBA and the network coding. The application layer was used to implement the packet sending and to track how many packets were sent and were successfully received.

Castalia also allows for information to be easily tracked and collected into a filtered text file. Castalia also has a simple to use interpretor for the data that was collected, allow for it to be processed before hand.

D.2 Network coding implementation

The most challenging component of this test bed was implementing a robust network coding algorithm. With there being three major components to this; an implemented finite field library, a matrix manager and a Gaussian Elimination algorithm.

To handle the finite fields, software by James S. Plank from the Department of Computer Science at the University of Tennessee was used. This library implements the finite field function of addition, subtraction, multiplication and division over an 8 bit finite field. While the field size is small, it was enough to build reliable software off, and its implementation was very fast.

The Gaussian Elimination software was just written from the well known algorithm, and partial pivoting is also included with the algorithm. For this algorithm, all 2 dimensional arrays were cast into a signal pointer, and then reverted back into a 2D array, with all matrices kept at a constant size of 10-by-10 elements for ease of implementation. The Gaussian Elimination was implemented to find if there are two linearly dependent functions included in the matrix - and it identifies which line needs to be replaced. All the operations were performed over the finite field. The only thing that Gaussian Elimination cannot do is always solve for when there are sparse matrices being solved.

Finally, a matrix manager was implemented. Due to Gaussian Elimination not being able to work with sparse matrices, a mechanism is needed to handle how matrices are being dealt with and finally a mechanism to manage how full a matrix is and if it is fully ranked. This is the role of the matrix manager. Firstly, it checks to see if there are enough equations that have been input into and the matrix is able to be solved. Once it is, it then runs through an algorithm that removes the zone columns - providing a square matrix that can be solved by Gaussian Elimination. After the matrix has been solved, it then reorders the elements to represent their correct positions.

The development of all this software allows for the successful implementation of network coding. All code was done in C++ as per Castalia's requirements, and simple data types were used for speed reasons. With matrices being kept at a fixed size of 10 for this simulation, this means all header's of the packets that carried the encoding coefficients where also 10 elements long. The data type used for all information were of type *unsigned char*, since they are 8-bits long.

D.3 A buffering for outgoing messages

Another software component is the outbuffer system. This system is used to build up the packets that need to be sent out - and keeps track of how many packets need to be sent. With the system having a min-cut of two, the outbuffer is alerted of this number and then it knows that for every two packets that come into its buffer, only one needs to be sent. Furthermore, the outbuffer also keeps all the packets that it receives and every time a call for one of those packets it made, it encodes all the packets inside it together and again multiplies each element with a randomly generated coefficient. This system ensures that there is a well distributed spread in the encoding and packets are unlikely to be linearly dependant.

The outbuffer also has a trigger that instantly clears it once a certain amount of time has elapsed. The purpose of this is to make sure no ‘old’ packets remain in circulation. By opting for this more dynamic system, it was found that there was a good throughput through the network.

D.4 Data collection and access to the code

The Castalia framework offers excellent data collection techniques, and all the results were recorded in a spreadsheet. With Castalia allowing for the code itself to define what needs to be collected and what the values mean, it is easy to collect the data.

All source code is currently on a private Bitbucket repository, with all results being stored too. A clone link can to the source code can be provided upon request.