An optimisation approach to improve the throughput in wireless mesh networks through network coding

Masters Dissertation

Dissertation submitted in fulfilment of the requirements for the degree

Master of Engineering (M.Eng)

in Computer and Electronic Engineering
at the Potchefstroom campus of the North-West University

C. van der Merwe

20047290

Supervisor: M.J. Grobler
18 November 2011

Declaration

I, Corna van der Merwe, hereby declare that the dissertation entitled "An optimisation approach to improve the throughput in wireless mesh networks through network coding" is my own original work and has not already been submitted to any other university or institution for examination.

C. Van der Merwe
Student number: 20047290
Signed on the 18th day of November 2011 at Potchefstroom.

Acknowledgements

I would like to thank Mrs. M.J. Grobler for her guidance and support throughout the completion of this research in this dissertation.

I would like to thank Telkom SA Limited (through Telkom Centre of Excellence at the NWU) as well as the National Research Foundation of South Africa for funding this research.

I would also like to thank my fellow research group members for their assistance and support over the past two years. The many laughs, coffee and jokes helped keep me sane.

I would like to thank Dr. F. Terblanche for his assistance and advice on the validation procedures of this study.

I would like to thank my parents, Klaas and Annemarie Wierenga, and all my friends and family for their unwavering support and encouragement throughout the years, leading up to this day. I would like to thank my loving husband, Cor, for his support, and never failing to lift my spirits.

Finally, I would like to thank my Heavenly Father for granting me with support and opportunities from His hand, every day.

Abstract

In this study, the effect of implementing Network Coding on the aggregated throughput in Wireless Mesh Networks, was examined. Wireless Mesh Networks (WMNs) are multiple hop wireless networks, where routing through any node is possible. The implication of this characteristic, is that messages flow across the points where it would have been terminated in conventional wireless networks. User nodes in conventional wireless networks only transmit and receive messages from an Access Point (AP), and discard any messages not intended for them.

The result is an increase in the volume of network traffic through the links of WMNs. Additionally, the dense collection of multiple RF signals propagating through a shared wireless medium, contributes to the situation where the links become saturated at levels below their capacity. The need exists to examine methods that will improve the utilisation of the shared wireless medium in WMNs.

Network Coding is a coding and decoding technique at the network level of the OSI stack, aimed to improve the boundaries of saturated links. The technique implies that the bandwidth is simultaneously shared amongst separate message flows, by combining these flows at common intermediate nodes. The number of transmissions needed to convey information through the network, is decreased by Network Coding. The result is in an improvement of the aggregated throughput.

The research approach followed in this dissertation, includes the development of a model that investigates the aggregated throughput performance of WMNs. The scenario of the model, followed a typical example of indoors WMN implementations. Therefore, the physical environment representation of the network elements, included an indoors log-distance path loss channel model, to account for the different effects such as: power absorption through walls; and shadowing.

Network functionality in the model was represented through a network flow programming problem. The problem was concerned with determining the optimal amount of flow represented through the links of the WMN, subject to constraints pertaining to the link capacities and mass balance at each node. The functional requirements of the model stated that multiple concurrent sessions were to be represented. This condition implied that the network flow problem had to be a multi-commodity network flow problem.

Additionally, the model requirements stated that each session of flow should remain on a single path. This condition implied that the network flow problem had to be an integer programming problem. Therefore, the network flow programming problem of the model was considered mathematically equivalent to a multi-commodity integer programming problem. The complexity of multi-commodity integer programming problems is NP-hard. A heuristic solving method, Simulated Annealing, was implemented to solve the goal function represented by the network flow programming problem of the model.

The findings from this research provide evidence that the implementation of Network Coding in WMNs, nearly doubles the level of the calculated aggregated throughput values. The magnitude of this throughput increase, can be further improved by additional manipulation of the network traffic dispersion. This is achieved by utilising link-state methods, rather than distance vector methods, to establish paths for the sessions of flow, present in the WMNs.

Keywords: *Aggregated throughput; integer programming; multi-commodity; network coding; network flow problem; network throughput; optimisation; simulated annealing; system throughput; throughput; wireless mesh networks.*

Opsomming

Die uitgevoerde navorsingsstudie van hierdie verhandeling, ondersoek die effek wat die uitvoering van Netwerkkodering het op die datadeurset van draadlose roosternetwerke ("Wireless Mesh Networks", WMNs). WMNs is draadlose netwerke waarin data oor veelvuldige skakels oorgedra word, met die bykomende eienskap dat elke node in die netwerk in staat is om hierdie oordrag uit te voer. In gevolge hiervan, kan die informasieboodskappe verder as sekere punte in die netwerk aangestuur word as wat toegelaat word in konvensionele een-skakel draadlose netwerke.

Die resultaat van hierdie eienskap, is dat die volume van netwerkverkeer deur die skakels van 'n WMN verhoog. 'n Bykomende faktor is die digte voorkoms van RF-golwe wat deur 'n gedeelde lugruimte moet voortplant, wat daartoe bydrae dat die netwerkskakels versadig word by 'n vlak laer as die kapasiteit van die skakels. Daar bestaan dus 'n behoefte om tegnieke te bestudeer wat die aanwending van hierdie gedeelde lugruimte in WMNs kan verbeter.

Netwerkkodering is 'n data kodering- en dekoderingstegniek by die netwerkvlak van die OSI stapel in netwerktoerusting, wat daartoe poog om die versadigingsgrens van netwerkskakels te verhoog. Dié tegniek lei daartoe dat die bandwydte gelyktydig deur verskillende vloeie van boodskappe gedeel kan word, deur die betrokke vloeie saam te voeg by tussenstaande nodes. 'n Enkele vloei word aan gestuur om sodoende die hoeveelheid transmissies wat benodig word, om al die boodskappe aan te stuur, te verminder. Die algehele datadeurset van die netwerk word hierdeur verbeter.

Die navorsingsbenadering, sluit die ontwerp van 'n wiskundige model in, wat die datadeurset vermoë van WMNs bestudeer. Die opstelling van die model stel gevalle voor vir tipiese inhuis toepassings van WMNs. Dus het die wiskundige voorstelling van die fisiese omgewing in die model, 'n inhuis logaritmiese-afstand padverlies kanaalmodel uitgevoer. Daarom is verskeie aspekte, soos die absorbering van drywing deur mure, asook gesamentlike refraksie en refleksie van die RF-golwe in ag geneem.

In die model is die netwerk bedrywighede voorgestel deur 'n netwerkvloei programmeringsprobleem. Die probleemstelling was daartoe gemik om die optimale hoeveelheid vloei, wat deur elke skakel in die netwerk voorgestel word, te bepaal. Elke skakel se vloei was beperk deur verskeie funksies wat verband hou met die skakelkapasiteite en die beginsel van ewewig in elke node.

Die gebruikstoepassing van die model het daartoe meegebring dat veelvoudige sessies van vloeie gelyktydig in die netwerkprobleem aanwesig is, wat elk op 'n enkele vloeibaan moet bly. Dus word die netwerkvloei programmeringsprobleem beskou as die wiskundige ekwivalent van 'n multi-kommiditeit heeltallige programmeringsprobleem. Die kompleksiteit wat geassosieer word met multi-kommiditeit heeltallige programmeringsprobleme is NP-hard. Daarom is die heuristiese metode, Gesimmuleerde Tempering, uitgevoer om die voorgestelde doelfunksie van die netwerkvloei programmeringsprobleem in die model op te los.

Die bevinding van hierdie navorsingsstudie lewer bewys dat die uitvoering van Netwerkkodering in WMNs die berekende algehele datadeurset naastenby verdubbel. Hierdie verbetering kan selfs verder verhoog word deur bykomende manipulasie van die verspreiding van die netwerkverkeer. Voorbeelde van sulke manipulasie kan bewerkstellig word deur die toepassing van vloei-oordragsmetodes wat gebaseer word op die vraag en fisiese toestande van die betrokke skakels, in stede van metodes wat uitsluitlik kortste roetes in ag neem.

Sleutelterme: *Algehele deurset; deurset; draadlose roosternetwerke; heeltallige programmering; multi-kommiditeite; netwerkdeurset; netwerkkodering; netwerkvloei probleem; optimering*

Contents

Appendices

List of Figures

List of Tables

List of Notations

List of Symbols

TP	Aggregated (system) throughput
X	Buffered symbol
C	Channel capacity
d	Distance
f	Frequency
Y	Incoming/receiving symbol
H	Neighbours
E	Network edges - links
G	Network graph
V	Network vertices - nodes
f	Node flow
P	Power
ξ	Sessions
λ	Session flow
s	Single session
λ	Wave length

List of Subscripts

c	Carrier wave
i	Current node
j	Upstream node
k	Downstream node
r	Receiving
t	Transmitting

List of Acronyms and Abbreviations

AM Amplitude Modulation

AP Access Point

CI Confidence Interval

FM Frequency Modulation

FSL Free Space Loss

IEEE Institute of Electrical and Electronics Engineers

ITU International Telecommunication Union

LNC Linear Network Coding

LOS Line of Sight

LP Linear Programming

MAC Medium Access Control

MIP Mixed Integer Programming

OFDM Orthogonal Frequency Division Modulation

OSI Open Systems Interconnection model

QP Quadratic Programming

RF Radio Frequency

RLNC Random Linear Network Coding

RSSI Received Signal Strength Indicator

TEM Transverse Electromagnetic

WLAN Wireless Local Area Network

WMN Wireless Mesh Network

1. Introduction

In this introductory chapter, a background description of issues pertaining to the research topic is given in section 1.1. The problem statement is defined in section 1.2, followed by a list of issues to be addressed in section 1.3. The methodology utilised to conduct the research is described in section 1.4. The chapter is concluded by providing a brief overview of the document structure in section 1.5.

1.1. Background

Wireless Mesh Networks (WMNs) are infrastructure-less wireless networks that exist when a collection of wireless nodes temporarily connect, in an ad hoc manner, to form a mesh. These networks are considered wireless multiple hop networks, in which each network node is able to perform routing tasks [1], [2], [3]. From this nature of WMNs, two areas of concern are identified: 1. The inadequacy of current routing methods for WMNs; and 2. The confined wireless environment of WMNs.

Conventional wireless networks do not need information routing schemes, since every user node is only one hop away from the wireless Access Point (AP). When the need for wireless routing protocols originated, the developers abstracted the protocols from methods originally intended for wired networks. The implication is that these protocols do not account for the intrinsic broadcasting nature of omnidirectional wireless nodes [4], [5]. Wireless messages can flow across certain points in the WMN where it would have been terminated in conventional wireless networks. User nodes in conventional wireless networks only transmit and receive messages from an AP, and discard any messages not intended for them. The result, effecting the network performance in terms of throughput, is an increase in network load without an increase in the original number of initiated sessions.

The wireless environment of WMNs entails harsh conditions for the propagation of Radio Frequency (RF) signals. The signals are subject to various degrading effects such as path loss, shadowing, interference, noise, disturbances and obstacles in the path of the signals [6], [7]. A valuable resource in WMNs is the RF bandwidth, which must be shared between the network users. High volumes of wireless traffic are utilising common links, leading to a decrease in the amount of bandwidth available to each user [1], [3]. In terms of throughput, the RF signals in WMNs are degraded by the effects from multiple wireless transmissions in a common broadcast range, together with the effects of space and matter on the quality of the signal propagation.

Network Coding is a data encoding and decoding process at the Network Layer of the Open Systems Interconnection model (OSI) stack. It is a means of sharing network resources between the commodities of a network. The technique offers relief to the wireless links burdened with multiple wireless transmissions, by improving the saturation threshold of the links. This aim is accomplished by reducing the number of flows present in the wireless medium of the link, in the following way: For two separate message streams, flowing in opposite directions through a common intermediate node, the messages are combined at the intermediate node to form a single broadcast message. The intended next hop nodes receive the combined broadcast message simultaneously, and resolve their intended messages through decoding procedures [5], [8], [9].

The physical environment of WMNs and the effect of Network Coding can be mathematically represented in a model of WMN systems. An established method to numerically analyse networks is to cast a network flow problem as a series of linear programming functions, and then solve the original problem [10]. Formulating and solving network flow problems as linear programming problems, is referred to as *network flow programming* [11], [12].

1.2. Problem statement

The shared wireless broadcast medium of WMNs, with multiple concurrent sessions, leads to the situation where the network links become saturated at a level below their capacity. This condition is due to a high volume of separate message flows, causing interference on each other as well as interference on arbitrary transmissions over adjacent links, subject to scheduling delay. This indicates a need to understand the effect that a Network Coding implementation will have, on the throughput performance of WMNs.

The performance of WMNs, in terms of aggregated throughput, is a combined effort of RF signal propagation through the environment of the network, together with the management of these message flows on definite paths through the network. This study sets out to develop a model of WMN systems, to assess the throughput performance of WMNs due to the implementation of Network Coding.

1.3. Issues to be addressed

The main objective of this research is to determine how the implementation of Network Coding improves the aggregated throughput of WMNs. In order to achieve this objective, the following issues will be addressed:

- The development and implementation of a model to represent WMN systems.
- The utilisation of Network Coding in the model.
- The method to calculate the optimal aggregated throughput of WMNs from the model.

1.4. Methodology

The steps taken in the research approach of this study was based on the scientific method. The classic linear scientific method is a process involving the following steps [13], [14]: 1. Description of real-world observations; 2. Formulating a hypothesis to explain (1); 3. Use the hypothesis to predict new observations; and 4. Evaluate the performance of the predictions.

The research process starts at the first step, and continually moves on to each consecutive step, once the tasks involved at the current step is completed. If a particular step cannot be completed satisfactory, the research process moves backwards to the previous step and make the necessary adjustments in order for the process to continue forward again [13], [14].

For this study, the method was expanded to the flow diagram presented in Figure 1.1. The first step was to formulate a clear statement of the the research problem, given in section 1.2. Section 1.3 stated the issues that will be addressed through this research. The next steps from the diagram in Figure 1.1, will now be discussed.

1.4.1. Literature study

A study of literature concerned with WMNs, Network Coding and the theory of network flow problems were conducted. The study of WMNs were aimed at gaining an understanding of the properties, type classification and operational conditions of these networks. This background knowledge aided in the development of the model by providing the framework of the WMN systems represented by the model. The background study of Network Coding techniques and implementations were utilised to determine a fitting approach for the development of a Network Coding principle in the model. The aspects of network flow theory studied gave the guidelines for the expression of network activities, in terms of message flows, as mathematical equations utilised in the model.

1.4.2. Model development and implementation

The numerical research in this study is concerned with the development of a mathematical model that represents WMN systems. This implies that the data necessary for the investigations in this study, is produced within an artificial environment. In order to produce results that accurately describe the behaviour of WMN systems, it is necessary to represent this controlled environment as realistic as possible. The topics discussed in sections 1.4.2.1 to 1.4.2.3, explain the different phases of the model development and implementation.

1.4.2.1. Network and data representation.

The number of nodes determined by the selected size of the network were evenly distributed in a mesh topology through a node placement routine. A maximum transmission distance were calculated, dependent on the maximum transmit power of the nodes. This distance indicated the maximum distance that the RF signals were able to propagate. In order to determine node adjacency, the

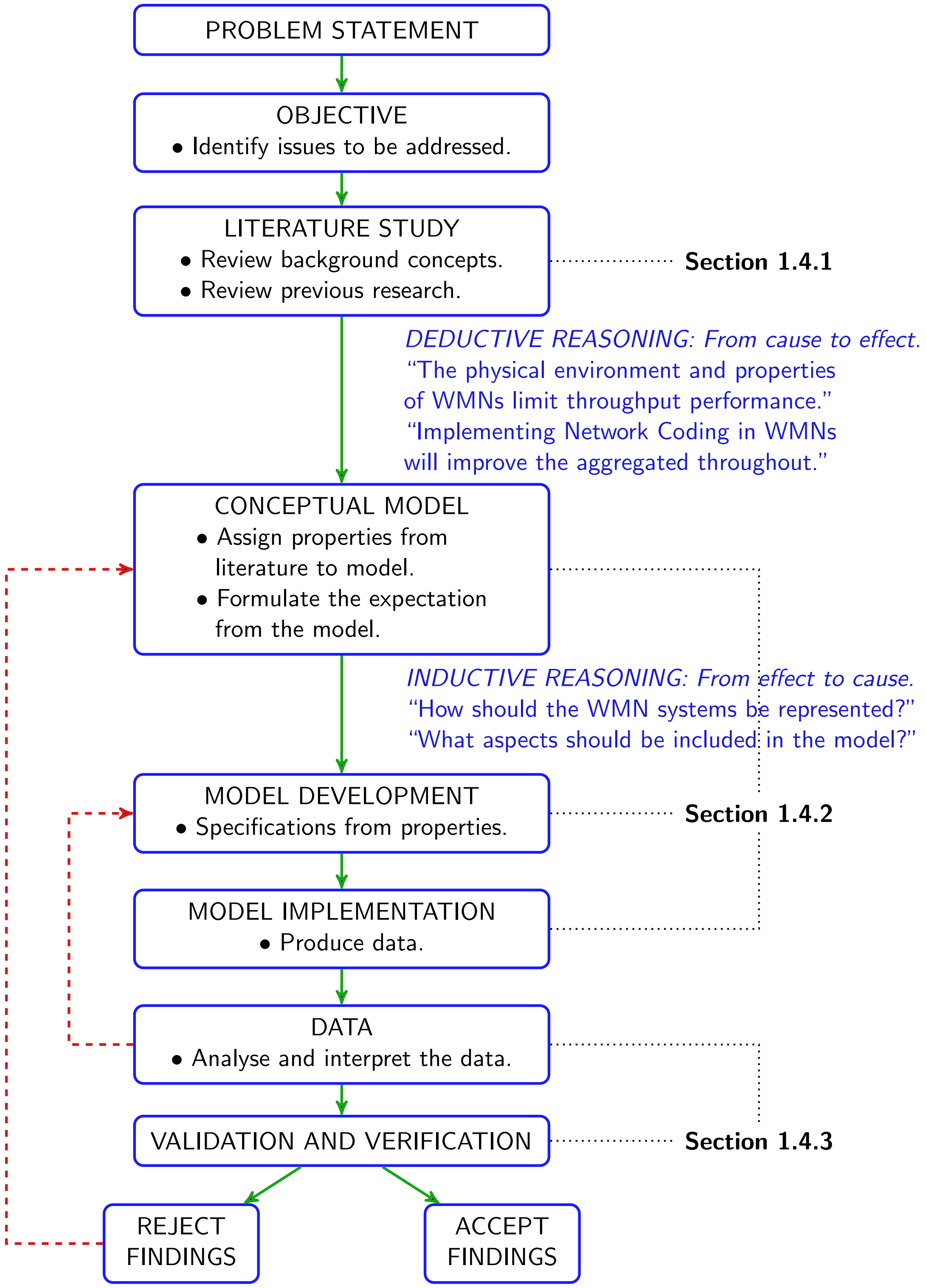


Figure 1.1.: Research methodology flow diagram

distance between all the combinations of paired nodes were calculated. Any distance less than the maximum transmission distance, yielded a valid link between the nodes, and the node pair were considered adjacent. The quality of the links were also represented by a Received Signal Strength Indicator for each valid link.

Additional to the representation of the nodes, links and link quality indicators, the data structures for each different level of network load were created. These data structures represented the information pertaining to the paths of the concurrent sessions present in the network, the occurrence of Network Coding opportunities, as well as the interference caused by active message flows on neighbouring links. The procedures utilised to create each network and to formulate the data structures, were developed and implemented in MATLAB®.

1.4.2.2. Network flow programming problem expression

The sessions presented in the model of WMN systems, were expressed in terms of flow and direction through a network flow problem. However, the functional requirements of the model stated that multiple concurrent sessions were to be represented. This condition implied that the network flow problem had to be a multi-commodity network flow problem.

Additionally, the model requirements stated that each session of flow should remain on a single path. This condition implied that the network flow problem had to be an integer programming problem. The network flow programming problem of the model were therefore considered mathematically equivalent to a multi-commodity integer programming problem.

Development of the equations expressing the network flow programming problem, were conducted following the approach of developing a linear programming problem. A goal function for the model were expressed, i.e. to calculate the maximum aggregated throughput of the network of the current investigated instance. This goal function were subject to the following: a vector of mass balance constraint functions, expressing the law of conservation of flow at each node on the session paths; as well as the capacity and Network Coding constraint functions, stating the boundaries of each link.

1.4.2.3. Optimal aggregated throughput calculation

A complexity analysis of the network flow programming problem for the model were conducted, in order to determine a fitting approach for the solving method of the model. The heuristic Simulated Annealing method was selected as an appropriate technique, and the algorithms utilised by this method were developed and implemented in C++.

1.4.3. Validation and verification of the model and results

In order to provide a comparative standard, to measure the performance of the model against, a control was developed in the form of a network flow problem. This control was specifically developed to be solved by a commercial CPLEX optimiser, a deterministic implementation of the simplex algorithm in a C-based language [15]. In order to consider the integer requirements of the control, an enumeration technique was implemented, which uses the simplex algorithm as solving method through the search space.

However, the CPLEX solver utilised a deterministic algorithm. Therefore the dimensionality of the network flow problem of the control had to be kept minimal. Benchmark properties and condition of the validation process were developed in order to keep the problem size within an acceptable range, which entailed that the Network Coding constraint and the effects of adjacent transmission interference were excluded from the model for the validation evaluation.

The conceptual logic of the model development was validated and verified through a process that listed the elements and properties of WMNs from the background literature, juxtaposed with the element and property developed and implemented in the model. This process was followed by a statistical validation of the model through several hypotheses tests.

In each hypothesis test, the behaviour of data sample parameters was determined by testing these parameters against certain assumptions stated in a null hypothesis, H_0, contradicted by the assumptions stated in an alternative hypothesis, H_A. The outcome of each hypothesis test either accepted, or rejected H_0. The model was proved to be a valid model that produces results with sufficient accuracy in efficient time.

1.5. Document layout

The overall structure of the dissertation takes on the form of six chapters, including this introductory chapter. The remainder of this document is structured as follows:

- **Chapter 2:** An in-depth study of literature pertaining to the aspects of WMNs, Network Coding and network flow theory are presented. The focus the each study was to gain an understanding in the different concepts needed to develop the model.
- **Chapter 3:** The model representing WMN systems are developed and implemented.
- **Chapter 4:** The data obtained from the model implementation are presented, together with an analysis of the observations. This chapter also provides a discussion regarding the interpretation of the results.
- **Chapter 5:** The developed model is proven to be a valid model, which yields results with sufficient accuracy in efficient time.
- **Chapter 6:** The conclusion gives a brief summary and critique of the findings. And finally, areas for future work are provided.

2. Literature study

This chapter provides a study of the literature relevant to the issues of the research topic. Section 2.1 describes the background of WMNs. A key issue of this section is the discussion on the challenges faced by WMNs. Section 2.2 describes the study of model development aspects. The important factors to include when developing the model, utilised in this research, are highlighted. The final study topic of the chapter, is the aspect of Network Coding, in section 2.3.

2.1. Wireless Mesh Networks (WMNs)

Current trends in next generation mobile and wireless networks show that the technology is shifting towards converged networks that suggest the integration of disparate technologies, interface platforms and services. One network configuration able to support this trend is a WMN. WMNs propose a foundation on which the integration of cellular networks, wireless sensor networks, Wi-Fi, WiMax and WiMedia networks, as well as the Internet is made possible.

2.1.1. Definitions and properties of WMNs

WMNs are infrastructure-less networks that exist when a collection of wireless users temporarily establish and manage connectivity amongst themselves in an ad hoc manner [1], [2], [3]. These networks have the following properties:

- *WMNs are self-forming and self-organised.* Each new node entering the network establishes its own identity and function in the mesh network. All nodes contribute to path discovery and maintenance [2], [3].
- *WMNs are self-healing.* The mesh configuration of WMNs often provides redundant paths between sources and sinks. Should an intermediate node fail or leave the network, connectivity between the source and sink is preserved (provided an alternative route is available) until the node is able to participate again, or a new node enters in its place [2].
- *WMNs are decentralised wireless networks.* There are no central managing nodes in WMNs. All network functions, such as routing, security, power management, network monitoring, etc., are carried out by each individual node that forms part of the WMN [2], [3].

- *WMNs are multi-hop networks.* Information is forwarded across the network by relaying traffic through intermediate nodes between sources and sinks [1], [2].

2.1.2. Classification of WMNs

WMNs consist of two types of network nodes [2], [3], namely wireless users and wireless routers.

- User nodes are any wireless device capable of establishing a network connection. For example: netbooks, 3G enabled cell phones, tablets, etc., are feasible network user nodes.
- A static backbone is a configuration of permanent routers and\or APs which may perform gateway and bridging functionalities. The purpose of this backbone is to offer a default route to the users in the WMN. This ensures a dedicated network coverage area, as well as reducing the number of hops between users communicating from the edge of the network. Backbone nodes usually deploy a different frequency band for a longer range to communicate with other backbone nodes [1].

The characteristics of WMNs are dependent on node type, node arrangement and node capabilities. Three classes of WMNs currently exist:

- **Pure or clients-only WMN:** This type of WMN only consist of user nodes forming a static ad hoc network. There are no backbone routers [1], [2], [3].

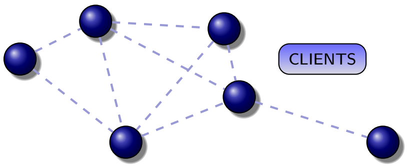


Figure 2.1.: Pure ad hoc WMN: all the nodes are users

- **Hybrid WMN:** User nodes connect to each other and to backbone router nodes. There is no connectivity between other networks external to the WMN [1], [3].

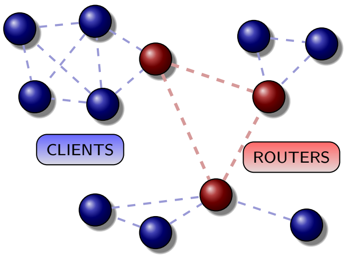


Figure 2.2.: Hybrid WMN: Acces routers connects clusters of users

- **Access or client-router WMN:** This is a hybrid WMN with added gateway functionality expanding connectivity to external networks. This can either be with other WMNs, the Internet, or even different network types such as wired, cellular or sensor networks [1], [3].

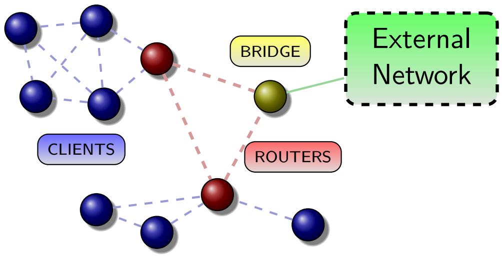


Figure 2.3.: Access WMN: Users connect to external networks, and interface with different network types

2.1.3. Current applications for WMNs

WMNs expand high-speed and robust broadband Wireless Local Area Network (WLAN) communication services at a low-cost. The target applications include city-wide broadband Internet services, neighbourhood community networks, All-office wireless services, rural networks and any other relevant application. Typical WMN applications are given by the following examples:

- military and border security
- video surveillance
- visitor usage
- emergency or rapid response services
- remote monitoring and control
- home media networking

The products offered by Motorola and MeshDynamics have already achieved what many conventional commercial wireless networks could not have - such as support for a high demand of data, as well as real-time voice and video streaming, critical to the applications listed above [17], [18].

2.1.4. The benefits offered by WMNs

The appeal of WMNs over wired network solutions or even conventional centralised wireless networks is expressed by the following advantages:

- *Low initial cost.* WMNs do not require high cost infrastructure such as base stations. Each node in the network is a connection point for new nodes that expand the network. Centralised wireless networks demand a high amount of APs to assure larger coverage areas with expensive wired connections to the Internet. WMNs can cover the same area with less APs. Expanding WMNs can also be done incrementally - on demand as the network users and network coverage requirements grow [2], [3], [19].
- *Robustness.* The mesh arrangement of nodes in a WMN provides path redundancy available between sources and sinks. Essentially this quality increases reliability of the network as single point failures and potential bottlenecks are minimised. [2], [3], [19].
- *Efficient coverage performance.* Signal power significantly drops when the transmission distance becomes too great. Users in conventional wireless networks are subjected to limitation on their distance from the AP to sustain their connection. Whereas users in a WMN utilise multiple

hops and channels that enable longer distance wireless communication without degrading the network performance [2], [3].

- *Easy to set-up.* Nodes automatically establish and maintain network connectivity to enable a interconnection service. New nodes that comply to the necessary standard can join the network seamlessly. Backbone routers can be common of-the-shelf equipment [2], [19].
- *Distribution of services.* Gateway or bridging functionalities in the backbone routers enable the interconnection of different services such as cellular, sensor networks, Wi-Fi, WiMax and WiMedia Networks. Through this integration users of existing networks can get access to services that are normally unavailable to them [3], [19].
- *Community-owned.* WMNs have a distributed network ownership, meaning the burden of network maintenance and support is shared by all the users, and does not rely on a single person or administrative body [19].

2.1.5. Challenges faced by WMNs

The study of challenges faced by WMNs, were divided into two separate categories. The first study examined the physical challenges existing in the environment of WMNs, and is referred to the *environmental* challenges. The second study considered the difficulties in network operations evolving from the properties of WMNs. These are the *operational* challenges. To implement a successful and effective model to represent WMNs, all these factors should be considered.

2.1.5.1. Environmental challenges

The properties of WMNs entail the operation of a dense collection of wireless nodes in a confined area. The space surrounding the nodes becomes occupied with RF signals, each of which is challenged to find it's designated destination over a course of multiple transmission hops. In order to understand the exact challenge cast onto each signal, it is necessary to know the basic principles of RF signals. Some of the fundamental definitions and properties of RF communication signals are explained as follows:

- **Radio:** Radio is a system where information is imposed onto an electromagnetic wave that is passed between a transmitter (source) and a receiver (sink) [6], [7], [20].
- **Information:** Information is *not* the RF *signal*. It is the essential content that must be transmitted.
- **Encoding:** Information needs to be prepared for a wireless broadcast. *Source encoding* converts the information into a sequence of bits. *Channel encoding* adds redundant bits to this sequence in a controlled manner. The purpose of channel encoding is to increase the signal's reliability against all the effects encountered en route to the receiver [6], [7].
- **RF signal:** A RF signal is an *energy* signal. The signal is defined by a moving field of electric and magnetic forces that propagate through space in the form of Transverse Electromagnetic (TEM) waves [6], [7].
- **Modulation:** An oscillator (or frequency synthesiser) provides a carrier frequency that will carry the channel encoded information sequence. The modulator impresses the information

sequence onto that carrier wave. This process is called *modulation*, and typical examples are Amplitude Modulation (AM), Frequency Modulation (FM) and Orthogonal Frequency Division Modulation (OFDM) [6], [7].

- **Bandwidth:** Bandwidth is the range (difference between highest and lowest) of frequencies that is contained in a composite signal [20]. *Broadband* refers to a signal with a very large bandwidth that is shared amongst many users.

The receiver must be able to recognise signals containing higher power levels at the tuned in frequency range[1] than surrounding noise values. From there, it demodulates the wave and interprets (decode) the imposed information sequence [6], [7]. But the transmitted RF signal will encounter many impediments on its way to the designated receiver. The power of a transmitted RF signal is effected by the following factors:

- **Distance:** RF signals attenuate over an unobstructed path as it travels away from its source (Free Space Loss (FSL)). FSL is frequency dependant: for higher frequencies the amount of attenuation is more than for lower frequencies [6], [7], [21].
- **Terrestrial influences:** Huygens principle states: "*At each point of a wave front new circular waves start*". Together with the fact that microwave beams widen, an area of concentric ellipsoids exists called the *Fresnel Zone*. Even with a clear Line of Sight (LOS), an obstructed Fresnel Zone hinders the propagation of RF signal since these obstructions cause many reflections (constructive and destructive) in the waves[2]. The effects of reflections are multipath distortions, fading and shadowing [6], [7], [21].

These properties of the RF environment pose several limitations on practical wireless networks in terms of the reliability of the signals and bandwidth aspects (such as network speed and network scalability). The most common limitations are:

- The distance from APs limits the communication link strength;
- Physical obstructions interfere with the communication link;
- Interference and noise caused by other WLANs, nearby busy nodes, electronic devices (like microwave ovens), fluorescent lights, and other relative originators of interference degrade the network performance;
- The shared medium limits the network efficiency;
- A large scale network carries a heavy network load;
- Every transmission is a broadcast, causing a higher payload of flooded data and session overheads; and
- The communication link strength is limited by power constraints.

2.1.5.2. Operational challenges

Practical WMNs have to overcome several constraints imposed by the natural properties of these networks. These constraints are divided into three categories in terms of interfacing aspects, protocol aspects and scalability issues.

---

[1] See appendix A.1 for a summary on current frequency spectrum allocations

[2] See appendix A.3 for explanations on the *wave front* and the *Fresnel Zone*.

- **Interface challenges for WMNs:** The first group of operational challenges resides in the interface layer operations of WMN devices. The two most prominent challenges are *link adjustments* and *heterogeneous interoperability* [3], [22], [23].
  - WMNs are dynamically formed. Therefore, the conditions under which the network operates vary in an unpredictable manner. Effective ways are needed to establish and manage link adjustments. It is also important to provide the upper layers in the OSI stack with sufficient link quality feedback. This information is essential in tasks such as detecting handover imminence and routing decisions.
  - A challenge exists for WMNs to provide for interoperability between devices from different manufacturers, as well as integrating multiple wireless interfaces. A multiple radio, multiple channel solution can become very expensive, and complicated to manage.
- **Protocol challenges for WMNs:** Protocol suites are the rules that govern specific actions in different layers of the OSI stack of the network devices. The challenges with regards to protocols are grouped into challenges pertaining the Medium Access Control (MAC) layer protocols, and challenges pertaining the Network layer protocols.
  - Knowledge of the network topology is needed for MAC protocols to aid the cooperation between neighbouring nodes, and nodes at multiple hop distances away. The dynamics of a WMN, with regards to topology changes, introduce difficulty in obtaining this knowledge. Another emerging concern is that the hidden-node problem[3] can surface easily if care is not taken in MAC protocol design [2], [22].
  - The routing protocols currently in use for wireless networks are hardly optimal and suitable for WMNs [1], [2], [22]. These protocols were derived from protocols used in wired networks, and do not account for the intrinsic broadcasting nature of omnidirectional wireless nodes [4], [5]. Another shortcoming is that these protocols were developed for *bursty* traffic and *voice* traffic. The trend in network traffic is shifting towards *data*, such as media streaming [24], which requires the routing of high volumes of traffic over longer periods of time [1]. Current routing methods simply cannot handle the data traffic optimally.
- **Scalability issues for WMNs:** WMNs must aim to support large network topologies without exponentially increasing the amount of network operations needed. The reality is that the performance of a network degrades as the number of hops between source and sink nodes increases [2], [3], [22], [23]. Addressing scalability issues remains an open research topic for all wireless networks.

2.2. WMN system modelling

A model is a logical abstraction or mathematical representation of a real-world system. Models are used to predict or analyse certain aspects of the real-world system in a conceptual manner. A well founded method to numerically analyse *networks* is done by casting a *network flow problem* as *linear programming functions*, in order to solve the original problem [10]. Formulating and solving network problems as linear programming is referred to as *network flow programming* [11], [12].

---

[3] See appendix A.4 for an explanation on the *hidden node problem*.

In communication networks the continued transfer of data between nodes over the underlying network elements and resources is present. This *flow* of bits is the *commodity* of the network, and transferring the commodity is at some *cost* to the network [10]. It is this perspective of networks that leads to a special grouping of problems referred to as **network flow problems** [10], [11], [12]. Examples of practical network flow problems are: the *transportation problem*, the *transshipment problem*, the *assignment problem*, the *maximum flow and minimum cut problem*, the *minimum spanning tree problem* and the *shortest route problem*.

In order to formulate a network flow programming problem representing the message flows of a WMN system, some issues are identified that will constitute to the model development. These issues are the following:

- Representing the environmental aspects.
- Representing the operational aspects.
- Representing the structure of the network.
- Methods to analyse the complexity of the problem represented by the model.
- Methods to solve the problem represented by the model.

2.2.1. Environmental aspects

The environmental aspects in the model, represents the physical environment of the real-world system. The purpose of aspects is to express the degrading effects on the quality of RF signals caused by space, matter and other phenomena. *RF channel modelling* techniques were studied in order to achieve this purpose. The wireless links between the nodes of the WMN are the *communication channels* of the system. It is possible to theoretically represent these channels by functions and statistical equations [6], [7], [25]. Typical aspects of the channel that are modelled by these equations are:

- The *capacity* of the channel.
- The *propagation* of the RF signal.
- *Influencing factors* effecting the RF signal.
- *Noise* introduced to the channel.

Some of the properties relevant to RF channel modelling are defined as follows:

- **Path loss:** Path loss attenuation is a deterministic expression for the loss in power in a transmitted RF signal. It is depended on the distance between the transmitter and receiver, and the frequency of the transmitted signal [6], [7], [25].
- **Shadowing:** RF signals arriving at a receiver are affected by the objects surrounding the transmission path, as well as the terrain between the transmitter and receiver. Shadowing is a stochastic *abstraction* that reflects the *result* of the sum of several phenomena affecting the propagation of RF signals [6], [7], [25]. Examples of these shadowing effects are:
  - *Reflections* from buildings and the ground.
  - *Diffraction* around buildings.
  - *Refraction* through walls or windows.
  - *Scattering* on buildings or the ground.
  - *Absorption* into trees and buildings.

- **Fading:** Fading is the interference of many scattered signals arriving at the receiving antenna. It is responsible for the most volatile changes of the signal strength, as well as changes in the signal phase. In a multipath propagation environment, the transmitted RF signal is reflected in such way that multiple copies of this signal cause interference at the receiving antenna. Fading is also a stochastic effect [6], [7], [25].
- **Radio Channel Attenuation:** Radio Channel Attenuation, $(a(t))$, is the total attenuation that the RF channel experience, composite of path loss $(a_{PL}(t))$, shadowing $(a_{SH}(t))$ and fading $(a_F(t))$ [6], [7], [25]:
$$a(t) = a_{PL}(t) \cdot a_{SH}(t) \cdot a_F(t) \tag{2.1}$$
- **Noise:** Noise is a degrading element that effects the *modulation* of RF signals. It is categorised as *thermal* and *artificial* noise [7], [25], [26]:
  - *Thermal noise*: Thermal noise, or Gaussian noise, is caused by the movement of charged particles inside electronic components.
  - *Artificial noise*: Equipment, like machines, produce radiating energy. This energy is referred to as artificial noise.
- **Interference:** Interference is caused by other RF transmitting electronic devices, producing electromagnetic disturbance patterns [6], [7], [25].
  - *Co-channel interference* occurs if two transmitters actively operate within the same radio frequency band. A receiver tries to receive the intended signal from the first transmitter, but also receives a (weak) signal from a second transmitter.
  - *Adjacent channel interference* is encountered when transmissions are conveyed on different (but close) frequency bands. It produces a significant interference on the power in a receiver. This effect is mainly due to imperfect pass band filters.

The following symbols are used during the discussion of the the environmental aspects of RF signal propagation:

- P_r at d : Receiving power in the propagated signal.
- d : Distance between the transmitter an receiver.
- P_t : Transmitted power in the propagated signal.
- λ : The wavelength of the propagated signal.

In order to develop the mathematical expressions representing the RF channel and environmental aspects in the model, the following functions and power calculations were studied:

- **Friis free space equation:** This received power equation shows how transmission power decreases as the distance of the link between the transmitter and receiver increases. The magnitude of the decrease is related to the square of the link distance d, and results in a 20dB/decade decay:
$$P_r(\text{at } d) = \frac{P_t \lambda^2}{(4\pi)^2 d^2} \tag{2.2}$$
where there is unity gains at the transmitting and receiving antennas, and no losses in the system [6], [7], [25].

- **Free space path loss:** With the received power calculated by Equation 2.2, the free space path loss, PL, is given by [6], [7], [25]:

$$\begin{aligned} PL(\text{in dB}) &= 10\log\frac{P_t}{P_r} \\ &= -10\log\frac{\lambda^2}{(4\pi)^2\,d^2} \end{aligned} \tag{2.3}$$

- **Far-field (Fraunhofer) region:** For the Friis model to be a *valid* predictor, the value of distance d must be inside the Fraunhofer region of the transmitting antenna. This region is any distance beyond the far-field distance, d_f, given by:

  $$d_f = \frac{2\,D^2}{\lambda}$$

  where D is the largest physical linear dimension of the antenna. It is also subjected to $d_f \gg D$ and $d_f \gg \lambda$ [6], [7], [25].
- **Closed-in reference point:** It is clear that Equation 2.2 does not hold if $d = 0$. In propagation models, a received power reference point can be obtained by using a *closed-in distance*, d_0. The value of d_0 must lie inside the Fraunhofer region, but be smaller than any practical distance used in wireless communication systems [6], [7], [25]. By using the closed-in distance in Equation 2.2, the received power at any distance greater than the reference point is given by:

$$P_r(\text{at } d) = P_r(\text{at } d_0)\left(\frac{d_0}{d}\right)^2 \qquad \text{where } d_f \leq d_0 \leq d \tag{2.4}$$

- **The log-distance path loss function:** A widely accepted model for indoors path loss is the log-distance path loss function [25]. This model conforms to the *distance power law*:

  $$PL(\text{in dB at } d) = PL(\text{in dB at } d_0) + 10\alpha\log\left(\frac{d}{d_0}\right)$$

  When $PL(\text{in dB at } d)$ and $PL(\text{in dB at } d_0)$ are replaced by the right hand sight values of Equation 2.3, this model transforms to:

  $$P_r(\text{at } d) = P_{r0}(\text{at } d_0) - 10\alpha\log(d)$$

  Further, an accepted value for the shadowing effect is a random value from a normal (Gaussian) distribution with a zero mean and standard deviation σ, given as X_σ. Adding this factor to the path loss model leads to the log-distance path loss model with added shadowing effect:

$$P_r(\text{at } d) = P_{r0}(\text{at } d_0) - 10\alpha\log(d) + X_\sigma \tag{2.5}$$

2.2.2. Network operation aspects

The aim of including network operation aspects in the model, is to represent the practical processes that are involved in a functioning network in a theoretical manner. The network operations were divided into two parts: 1. The issues concerned with path establishment in multiple hop networks; and 2. Augmenting these paths with flow.

2.2.2.1. Network paths

The task of routers in practical multiple hop networks, is to determine the next hop address of queued packets and to forward these packets accordingly. The metric utilised in order to make this decision, is dependent on the routing protocol implemented by the network devices. However, most routing protocols are based on two fundamentally comparing methods: *distance vector* algorithms and *link-state* algorithms [27], [28].

In terms of model development, the methods utilised by real-life routers are adopted to establish the methods for selecting paths in the operational aspects of the model. These methods are explained as follows:

- *Distance vector algorithms:* Every node creates a table of distances $D_T(i,m)$ from the current node *i* over **all** neighbours *m* towards **all** destinations *T*. Node *k* is selected as the next hop node for a path if $D_T(i,k) = \min D_T(i,m)$. Examples of distance vector algorithms are Dijkstra's shortest path algorithm and the Bellman-Ford shortest path algorithm [27], [28].
- *Link-state algorithms:* Every node calculates the link-state cost between itself and adjacent nodes (neighbours). A next hop decision is based on a comparison of the destination node of the flow, to the neighbour with the least cost en route to that destination [27], [28].

2.2.2.2. Network flow

The main concern of the model is to calculate the optimal aggregated throughput for WMNs. If the value of flow present on each link of all the paths present in the network is known, it is possible to conduct this calculation. A study on network flow problems were conducted in order to gain an understanding of developing the network flow programming problem for the model that will achieve this aim.

DEFINITIONS AND PROPERTIES USED IN NETWORK FLOW PROBLEMS

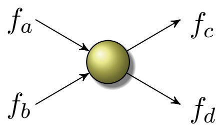
The definitions and properties utilised in network flow problems are given as follows [10], [11], [12]:

- **The network, G**: A mathematical representation for a network is a system of linear features connected at intersections (nodes) and the edges connecting any given pair of nodes. From Graph Theory abstraction, a network G is presented as a directed graph:

$$G = (V, E) \tag{2.6}$$

- **Vertices, V**: V in Equation 2.6 is the indexed set of nodes (vertices).
- **Edges, E**: E in Equation 2.6 is a spanning set of directed links (edges). (Also referred to as *arcs.*)
- **Variables:** The unknown flows carried by the edges $f_{(i,j)}$ from al nodes *i* to *j*, are the variables.

- **Notation:** For a node *i* ∈ *V*, $f^{+}(i)$ denotes the net flow leaving *i*, and $f^{-}(i)$ denotes the net flow entering *i*.
- **Capacity:** Every edge *e*(*i*, *j*) ∈ *E* is assigned with a upper bound rational number called the *capacity* of the link, $C_{(i,j)} \geq 0$.
- **Flow f:** A real function, $f = V \times V \rightarrow \mathbb{R}$, assigning values to an edge connecting each pair of vertices (*i*, *j*) ∈ *E*. A feasible flow are subjected to the following:
    - *Capacity constraint:* $0 \leq f_{(i,j)} \leq c_{(i,j)}$, ∀ (*i*, *j*) ∈ *E*.
    - *Flow conservation:* $f^{+}(i) = f^{-}(i)$, ∀ *i* ∈ *V* $-\{s,t\}$. Also referred to as the *mass balance* constraint [10].



$$0 = f_c + f_d - f_a - f_b$$

Figure 2.4.: Flow conservation (mass balance) in an intermediate node

- **Source s and sink t nodes:** Special entryway nodes which forms the interface with the environmental border of the network. At *s* there is a net gain of flow into the network and at *t* there is a net loss of flow out of the network.

$$f_s = f_c + f_d - f_a - f_b$$

(a) Source node

$$-f_t = f_c + f_d - f_a - f_b$$

(b) Sink node

Figure 2.5.: Net flows in source and sink nodes

- **Network flow:** The **value** of λ, $|\lambda|$, is equal to any of the following:
    - The net flow out of the source (*s*): $f^{+}(s)$ - $f^{-}(s)$
    - The net flow into the sink (*t*): $f^{-}(t)$ - $f^{+}(t)$
- **A Cut:** For X a non-empty subset of V and $\bar{X} := V - X$, $(X, \bar{X})$ is a *cut* of network G. A *source/sink cut* of G is cut (S, T) with $s \in S$ and $t \in T$. Implicitly $T = \bar{S}$
- **Cost per unit flow $c_{(i,j)}$:** For each edge *e(i,j)* ∈ *E* there is an associated cost per unit flow $c_{(i,j)}$.

BASIC CASE FOR NETWORK FLOW PROBLEMS

The *minimum cost flow model* is the most fundamental of all network flow problems. A large array of network problems can be cast as minimum cost network flow problems, and solved as such [10], [11].

A network $G = (V, E)$, with the following properties is defined:

- Each edge $e(i,j) \in E$ has an associated cost $c_{(i,j)}$ denoting the cost per unit flow on that edge. Assume the flow cost varies linearly with the amount of flow.

- A capacity $C_{(i,j)} \geq 0$ associated with each edge $e(i,j) \in E$ denoting the maximum amount that can flow on the edge.
- The decision variables, edge flows, represented as $f_{(i,j)}$.

The minimum cost flow problem as an optimisation model is formulated as follows:

$$\text{minimise} \sum_{edges} c_{(i,j)}\, f_{(i,j)} \tag{2.7}$$

subject to:

$$0 < f_{(i,j)} \leq C_{(i,j)} \qquad \forall\, e(i,j) \in E \tag{2.8a}$$

$$0 = f^{-}(i) - f^{+}(i) \qquad \forall\, i \in V - \{s,t\} \tag{2.8b}$$

The basic case of the network flow problem has the following two properties:

- *Duality Property:* A maximisation and a minimisation problem have the property that any feasible solution of the minimum problem is greater than or equal any feasible solution of the maximum problem. Both problems have the same optimum [10], [11], [12]. This property entails that the maximum flow will occur at a minimum cost to the network.
- *Integrality Property*: if the finite costs are all integers, and the maximum flow is bounded, then there is an optimal solution to the linear programming relaxation which will be an integer [10], [11], [12].

ADDING COMPLEXITY TO THE BASIC CASE FOR NETWORK FLOW PROBLEMS

The minimum cost flow problem from Equation 2.7 is categorised as a *single commodity* problem, since it solves the problem of finding the path with minimum cost to the network for a single commodity through the underlying network. In practice, the situation where multiple commodities must share the underlying network is usually the case. A flow exist for each pair of source and sink nodes. These flows are uniquely defined, *separate* commodities. Networks with this property are called *multi-commodity* networks. Solving multi-commodity network flow problems proves to be somewhat more complex than the original basic case problem.

- **Multi-commodity flow problems:** The fundamental distinguishing factor between single commodity and multi-commodity flow problems, is the fact that multiple commodities must *share* the underlying network. This property introduces additional parameters to the constraints on the equations describing the network [10]. It is important to differentiate between the multiple flows of each link in order to maintain each flow [11], [12]. The implication is that a flow vector and separate set of conservation functions must be kept for each individual commodity. Equation 2.7 is adapted to account for multi-commodity flow problems:

$$\text{minimise} \sum_{\substack{e \in E \\ k \in K}} c_{e,k}\, f_{e,k} \tag{2.9}$$

subject to:

$$\sum_{1 \leq k \leq K} f_{i,j,k} \leq u_{i,j,k} \qquad \forall\,(i,j) \in E \text{ and } k = 1, 2, \ldots, K \tag{2.10a}$$

$$0 \leq f_{i,j,k} \leq u_{i,j,k} \qquad \forall\,(i,j) \in E \text{ and } k = 1, 2, \ldots, K \tag{2.10b}$$

Here, *k* represents the different commodities in the set of all flow pairs, bound to a set of K mass balance constraints, which models the flow of each commodity $1, 2, \ldots K$. $f_{e,k}$ is the vector of flows for each commodity *k*, and $c_{e,k}$ is the vector of costs. The total flow of commodities on each link (i, j) is restricted to $u_{i,j}$.

For some cases, where the capacities on the links are not shared by the multiple commodities, solving the multi-commodity flow problem are not complex. The procedure is simply to create an individual network for each commodity and then solve the larger network with techniques developed for the basic case. However, for most cases with shared capacities, a multi-commodity modelling method is needed [10], [11], [12]. These techniques are not as well developed as for single commodity network flow problems [11]. Additional effort is needed to express multi-commodity problems in order to formulate an appropriate solution. Another discouraging characteristic of multi-commodity problems is that the integrality property *does not apply*, which add to the complexity of multi-commodity integer problems [11], [12].

- **Integer flow problems:** When a commodity flow $|\lambda|_k$ are subjected to remain integer, it means that in the effort of seeking the maximal possible flow (or minimal cost to network), $|\lambda|_k$ cannot be divided to follow multiple augmenting paths through the network [11], [12]. If this is the case, additional constraints and rules are imposed onto the main algorithm, which is different for each commodity. Solving all these commodities simultaneously leads to a very large search domain in finding the optimal solution, and can quickly become computational intensive.

2.2.3. Representing the network structures of the model

The means by which a network is presented, and how program variables are stored and updated, have an effect on the performance of the solving algorithm. Therefore, a suitable network representation and improved data structures are needed to ensure better running times for the algorithm. **Network representation** consists the following information types [10]:

- The network topology.
- The data associated with the network's nodes and edges (e.g. costs, capacities, etc).

Table 2.1 summarises some of the basic network representation techniques and the characteristics associated with each [10].

2.2.4. Complexity aspects of the model

The measurement of a problem's complexity is a rather abstract matter. Computer science and applied mathematics communities devised a set of analysis tools in a framework referred to as **computational complexity theory**, [10], [29]. Some of the properties of computational complexity

Table 2.1.: Popular network representations

Network representations:	Characteristics:
Node-edge incidence matrix	Space inefficient Hard to manipulate Represents constraint matrix
Node-node adjacency matrix	Used for dense networks Easy to implement
Adjacency list	Space efficient Easy to manipulate Used for dense and sparse networks
Forward and reverse star	Space efficient Easy to manipulate Used for dense and sparse networks

theory are discussed next, followed by the guideline utilised to classify problems according to its complexity.

2.2.4.1. Definitions and properties of computational complexity theory

Computational complexity theory utilises different notations to quantify the complexity of a problem. Some of the definitions and properties included by this process are explained as follows:

- **Problem size:** A quantified measurement of the size of a problem is calculated by creating a function of the elements in the problem representation. A minimum cost network flow problem, represented as an adjacency list, stores one pointer for each node and edge, and one data element for each edge cost coefficient and each edge capacity. The problem size is approximately $n \log n + m \log m + m \log C + m \log U$ bits, with n number of nodes and m number of edges in the underlying graph. C is the largest edge cost and U is the largest edge capacity [10], [29].
- **Time complexity (Worst-case complexity):** Time complexity is the number of steps or iterations taken by an algorithm, as a function of the size of its input, $T : N \rightarrow N$. One iteration is an operation that takes constant time, such as a variable assignment, a comparison, an array access or an arithmetic function. Time complexity measurement specifies the largest amount of time needed by the algorithm to solve any problem instance of a given size. For some constant $c \geq 0$, the running time needed to solve any network problem with n nodes and m edges is at most cnm. The time complexity measurement is referred to as a *worst-case analysis*, for it expresses the upper bound on the time taken by the algorithm [10], [29].
- $\mathcal{O}$ **("big-oh") notation:** A transformation that captures the dominant term, and ignores all constants in the problem. The transformation of a time complexity measurement is given as $\mathcal{O}(mn)$ time. Formally, the $\mathcal{O}$ notation is defined as follows:

  An algorithm is run in $\mathcal{O}(f(n))$ time if, for some numbers c and n_0, the time taken by the algorithm is at most $c\,f(n)\;\forall\; n \geq n_0$.

  The $\mathcal{O}$ notation is used to state the complexity of a problem as the asymptotic (bounded) growth rate of the running time of the problem. It assumes that each elementary mathematical operation, such as addition, subtraction, multiplication, division, assignment, and logical operations, etc. requires an equal amount of time to execute [10], [29].

- **Polynomial- and exponential-time algorithms:** A widely accepted definition for a "good" network algorithm is if it's worst-case (time) complexity is bounded by a polynomial function of the problem's parameters (a polynomial function of n, m, $\log\, C$ and $\log\, U$). The classification of such algorithms is *polynomial-time algorithms*. Examples are $\mathcal{O}(n^2)$, $\mathcal{O}(nm)$, $\mathcal{O}(n+m \log C)$ and $\mathcal{O}(nm\, \log\, \frac{n^2}{m})$.

  For *Exponential-time algorithms*, the worst-case running time grows as a function that cannot be polynomially bounded by the input length. Examples are $\mathcal{O}(nC)$, $\mathcal{O}(2^n)$ and $\mathcal{O}(n!)$.

  Polynomial-time algorithms are asymptotically higher ranked than exponential-time algorithms, and perform better. Therefore, the ideal is to obtain polynomial-time algorithms with the smallest possible growth rate, which would likely solve larger problems in the same amount of computing time [10], [29].

2.2.4.2. Classification of problems in terms of complexity

In computational complexity theory, problems are divided into different classes, in accordance with [29], [30]:

- How "quickly" the problem can be solved; and
- How "quickly" the solution can be verified.

Within this context, the reference to "quickly" basically means in how many steps the action involved can be executed. If an algorithm can yield the right solution (or set of solutions) for a problem with an input string of length *n*, the problem is considered to be a class P problem if the following condition holds [29], [30]:

The solution is achieved within $c \cdot n^k + c$ steps,
with constants *c* and *k* independent from the input string.

Class P problems are solved in polynomial time, and are termed as *deterministic polynomial-time* problems. For some cases the number of steps needed to solve a problem surpass the condition for P problems by far, however, these problems can be verified to be correct fairly "quickly". These problems are considered to be *non-deterministic polynomial-time* problems, alternatively class NP [29], [30].

- Class P problems are *solved* in polynomial time.
- Class NP problems are *verified* in polynomial time.

The set of problems in NP with the highest computational complexity rating is considered to be NP-complete [29], [30].

For a problem to be NP-hard, it must be " *at least as hard as the hardest problem in NP*". If an optimisation problem H has an NP-complete decision version L, then H is NP-hard [29], [30].

2.2.5. Solving method for the model

Knowing the complexity of a problem, ease the task of developing a solving algorithm that is fitting and optimal for the problem at hand. Solving a network flow problem typically requires the solution of an optimisation model with hundreds or thousands of variables, equations and inequalities. Computer programs to execute the calculations and procedures are implemented utilising several existing algorithms[4]. A study were conducted to investigate the procedures and evaluation metrics necessary when developing solving algorithms.

There are three basic elements to computational problem solving [10], [29]. These are the following:

- The algorithm for solving a specific class of problems;
- The computer program design and implementation for this procedure; and
- Applying the method to the data of a specific problem.

The important factors of good algorithms are given as [10], [29]:

- The **performance** of the algorithm - is it processing strenuous?
- The **effectiveness** of the algorithm - how fast is the algorithm (i.e how many iterations are needed)?
- **Accuracy** - are the results trustworthy?

Ideally, algorithms should be devised that have a small or reasonable computation time for the problems met in practice, and lead to results that are accurate. Accuracy can easily be measured by the correct validation and verification procedures.

There are many approaches to follow when developing an algorithm. The two most common approaches in network flow programming problems are listed as follows:

- **Deterministic Algorithms:** Deterministic methods aim for an exact solution for the underlying problem. An optimum solution is guaranteed with each execution of the algorithm. However, it is computational intensive and will take excessive time when the problem size becomes too large.
- **Heuristic Algorithms:** Heuristic methods search for solutions based on some type of decision criterion that can range from probabilistic to stochastic parameters. An optimum solution is not guaranteed, however an approximated solution is offered which usually falls close enough to the exact solution. Heuristic methods aim to be fast and reliable and offers an alternative to deterministic methods when the problem size is very large.

The deciding factor in selecting a fitting approach is the *size of a problem*. As a rule of thumb, the measurement of size is the amount of variables to solve for simultaneously, x, and is stated as: 1. *Small:* $x \leq 40$; 2. *Medium:* $41 \leq x \leq 200$; and 3. *Large:* $x > 200$. From this size categories, the appropriate approach for developing a solving algorithm is summarised by Table 2.2.

---

[4] See appendix B for the descriptions on these solving techniques.

Table 2.2.: Solving methods associated with problem size

Problem size:	Solving method:
Small	Enumeration Matrix algebra
Medium	Branch and bound Simplex Dynamic programming Other deterministic methods
Large	Branch and bound variants Controlled random search Network Simplex Pure random search Other heuristic methods

2.3. Network Coding

Network Coding is a data encoding and decoding process at the Network Layer of the OSI stack. It is a means of sharing network resources, such as bandwidth, to decrease the amount of transactions needed to convey multiple data flows over intermediate nodes, thereby increasing the data transfer rate. As information packets are passed on and relayed through the intermediate nodes, data from separate flows are combined and forwarded as a single flow. This process is successful if the designated receiver can restore the intended messages (decode the coded flow). The contributions made to Network Coding by [5], [31], [32] and [33] have proved successful Network Coding procedures for the cases of multiple unicast and multicast sessions through a network.

2.3.1. Network Coding: Example for the wireless environment

Wireless nodes broadcast their messages by default [5], [8], and any node within reception range of a transmitting node can receive this messages, regardless of the transmitter's intend. Network Coding offers a way to utilise this natural occurrence. Consider the following example: Two nodes, *A* and *B* exchange data with each other through an intermediate node *C*. Node *A* will send message *a* to node *B*, while node *B* will send message *b* to node *A*. See Figure 2.6:

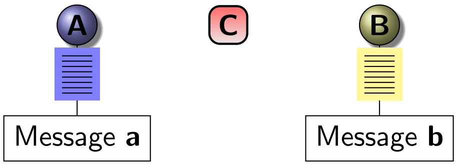


Figure 2.6.: Wireless multiple unicast example

Conventionally, nodes *A* and *B* each takes a transmission to send their respective messages to node *C*. Node *C* can only receive one message per transmission. After receiving both messages in it's queue, node *C* process these messages and relay message *a* with one transmission to node *B*, and uses another transmission to relay message *b* to node *A*. The total number of transmissions needed for nodes *A* and *B* to send each other a message through node *C*, adds up to **four** transmissions.

Figure 2.7 shows the abstracted view of this sequence of transmissions.

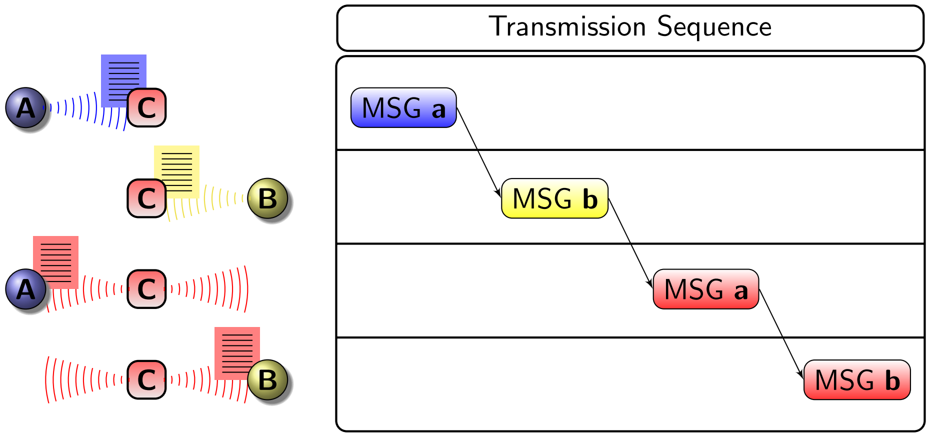


Figure 2.7.: Abstracted view of conventional message forwarding in a multiple wireless unicast

A Network Coding opportunity exists at node *C*: Message *a* and message *b* can be combined, after both messages are queued in node *C*, to form a single, combined message ($a \oplus b$). Nodes *A* and *B* will receive the same combined message in one transmission. To retrieve message *b* at node *A*, it uses the cached message *a* in its memory to reverse the Network Coding procedure and extract message *b* from the combined message ($\mathbf{a} \oplus \mathbf{b}$). Likewise, B uses message *b* in its memory to resolve message *a*. The total number of transmissions needed for nodes *A* and *B* to send each other a message through node *C*, adds up to **three** transmissions.

Figure 2.8 shows the abstracted view of this sequence of transmissions.

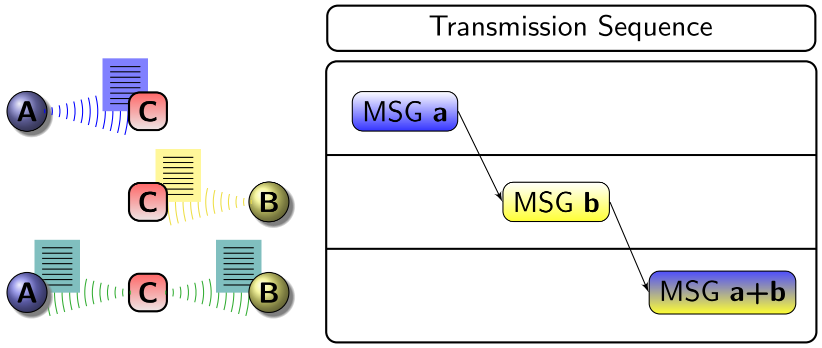


Figure 2.8.: Abstracted view of network coded message forwarding in a multiple wireless unicast

2.3.2. Types of Network Coding

A study of the literature pertaining to Network Coding, revealed three types of Network Coding that can be identified. These are Linear Network Coding (LNC), Random Linear Network Coding (RLNC) and Logical Network Coding.

2.3.2.1. Linear Network Coding

LNC is an algebraic approach in which linear combinations, of the network flow symbols and structured Network Coding coefficients, are formed as the flows pass through the network [31], [32], [34]. The notations followed by LNC are given by:

- Data units are declared as the symbols X and Y, from a defined finite field $\mathbb{F}_q : X , Y \in \mathbb{F}_q$
- $X_1 ... X_i$ denotes the symbols currently in node *i*'s buffer.
- $Y_1 ... Y_j$ denotes the symbols node *i* is receiving from its upstream links.
- $i , j \in V$, where V is the set of nodes in network $G = (V, E)$.
- Network Coding coefficients for the respective symbols are (α_i , β_j).

The network coded symbol, Y_k, is the linear combination

$$Y_k = \sum_i \alpha_i X_i + \sum_j \beta_j Y_j \qquad \forall (\alpha_i , \beta_j) \in \mathbb{F}_q \tag{2.11}$$

and is always on an outgoing edge towards a downstream node.

The properties of LNC include the following:

- $\mathbb{F}_q$ is defined by $q = 2^m , m \in \mathbb{R}$. This symbol field is larger than the binary field $\mathbb{GF}(2) =$ (0,1), which is the conventional composition of digital streams.
- Combining packets does not increase the packet size, since the Network Coding operation is element-wise.
- Transmitted symbols are recovered by destination nodes via Gaussian elimination (solving the linear system). This is possible if sufficient independent linear combinations are received, and this property is referred to as the *degrees of freedom*.
- This method of Network Coding is highly effective in static networks with pre-calculated node placements. This means that the flow patterns must preferably stay constant and not be volatile.

2.3.2.2. Random Linear Network Coding

Data flows through networks are highly volatile and hard to calculate in advance. Determining network codes in a centralised manner is therefore not feasible. RLNC is an algebraic approach to form linear combinations of the network data symbols and Network Coding coefficients. However, these Network Coding coefficients are calculated randomly.

The properties of RLNC include the following:

- RLNC is a distributed method [35], [36].
- Linear combinations of received symbols is performed in the same fashion defined by Equation 2.11. The difference is that nodes select independent and random Network Coding coefficients from a finite field [35], [36], [37].
- Destination nodes perform Gaussian elimination to solve the linear system and retrieve intended messages [35], [36].

A sufficient degree of freedom entails that the intended message can be recompiled from the information spread over multiple received instances.

2.3.2.3. Logical Network Coding

Logical Network Coding is a binary logic approach to Network Coding. All coding and decoding calculations are performed in the binary field, $X\,,\,Y\ \in\ \mathbb{GF}(2) = \{0\,,\,1\}$. The Network Coding calculation is a logical exclusive OR (XOR) arithmetic, [5], [9].

The Network Coding arithmetic is subject to the existence of Network Coding opportunities, which implicates that incoming flows are not necessarily combined. A Network Coding opportunity exists if the downstream nodes are able to retrieve their intended messages. S. Katti et al. [5], describe this opportunistic Network Coding principle as follows: *"...it should aim to maximize the number of native packets delivered in a single transmission, while ensuring that each intended nexthop has enough information to decode its native packet."* A *native packet* is an original, non-encoded data packet. The product of [5] is the COPE opportunistic Network Coding algorithm for wireless networks.

2.3.3. The advantages of Network Coding

Networks in practice have many adjacent nodes common to different flow paths that are not link disjoint. These links must be shared between the multiple flows, and the amount of bandwidth available for each separate flow is reduced. The consequences are:

- Slower transfer of data; and
- an increased probability of retransmission due to back-off procedures on the MAC level of the OSI.

Conservation of information flow [38] indicates that the accumulated information entering a node must equal the accumulated information that exit that node. This applies to every individual flow through the network. For a multiple hop path (information forwarded through intermediate nodes) the transfer rate of the flow as a whole, is only **as fast as the slowest link on the route** between source and destination.

Theoretically, the upper bound of flow through a network is the minimum of the maximum flows between the source and destination (Min-Cut Max-Flow Theorem) [38]. However, the fact that flows are forced to share common resources and are limited by the conservation of flow law, the aggregated throughput in practice is less than the flow calculated by the Min-Cut Max-Flow Theorem. Ahlswede et al. [33], established the benchmark for inter-flowing networks, proving that the multicast capacity of a network can reach this upper theoretical bound when LNC is implemented.

This inspired many research to aim for a gain in throughput for various network types, including wireless networks (where shared bandwidth becomes heavily burdened very quickly). Combining independent flows in the broadcast nature of wireless networks increase the forwarding capacity of shared links by reducing the number transmissions needed to forward all incoming packets [5], [8]. The significance of Network Coding procedures in networks, is that the aggregated throughput can be increased.

This chapter provided a study of the literature relevant to the issues of the research topic. Section 2.1 described the background of WMNs, the temporary collection of wireless nodes connecting, in an ad hoc way, to form a mesh. There are three types of WMNs, namely pure, hybrid and access WMNs. The classification of each WMN type depends on the functionality and application of the network.

Pure WMNs consist only of user nodes and do not offer connection to external networks. Hybrid WMNs have the property of a router backbone present in the network, that connects smaller groups of user nodes to form a larger WMN. Access WMNs contains APs that allow for the WMN to connect to external networks.

WMNs are subject to the harsh conditions of the wireless broadcast medium. The environmental challenges faced by these networks, include the degrading effects that space and matter have on the propagation of RF signals, as well as interference caused by multiple RF signals present in a shared wireless medium.

Section 2.2 discussed the aspects needed to develop a model representing WMN systems. The environmental aspects concerned in the model development, were addressed by a discussion on RF channel modelling. An important aspect was the deduction of the indoors log-distance path loss model.

The theory of network flow programming problems, showed how the operational aspects of WMNs can be expressed in a mathematical format. The issue of multiple concurrent sessions were also discussed and it was shown, that by representing multiple sessions in a network flow problem, together with the stipulation that these flows should remain on a single paths, the multi-commodity and integer properties are assigned to the network flow problem.

Section 2.3 discussed Network Coding, a coding and decoding process at the network level of the OSI stack. The technique implied that seperate flows, from opposite directions through a common intermediate node, can be combined in order to share the available bandwidth simultaneously. The result is in an improvement of the aggregated throughput.

3. Model development

In this chapter, a model is developed that is utilised to investigate the aggregated throughput of WMNs. Section 3.1 describes the scenario of the model in the form of typical indoors implementations of WMNs. The assumptions made for the developed model is given in section 3.2. The procedures contributing to network and data representation are described in section 3.3. Section 3.4 explains the network flow programming problem developed for the model, followed by a complexity analysis of the problem in section 3.5. The method utilised to solve the problem are presented in section 3.6. The chapter is finalised by giving a description of the different network instances investigated through the model implementation, in section 3.7.

3.1. Scenario description

The representation of the WMN system was developed in the form of typical indoors WMN implementations. An example of such a typical scenario is a university residence where students connect to the WMN with their WMN devices, from their respective rooms. A series of file sharing sessions are conducted between the students. Figure 3.1[1] gives a graphical view of the scenario example.

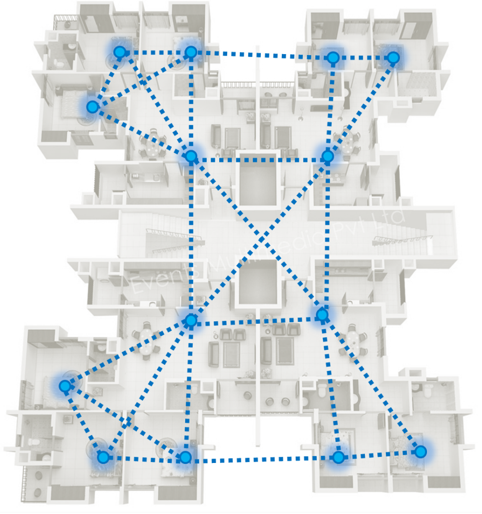


Figure 3.1.: Typical scenario example: students forming a WMN in a residence

---

[1] Background floor layout sourced from http://3ddubai.blogspot.com/2011/05/3d-floor-plans.html

Bearing in mind the scenario described above, the model will have to consider the following aspects:

- The physical environment implies indoors wireless WMNs, including obstructions such as the walls of the dormitory rooms between the nodes.
- The file sharing sessions between the students imply that a multitude of concurrent flows are present.

3.2. Model assumptions

During model development, some aspects were assumed to be true in order to simplify the model. These assumptions are listed below.

- Assumptions regarding the nodes of the network:
  - All the nodes are considered to be generic, therefore the type of device used by each student does not have an influence on the characteristics of the node and all nodes are equal.
  - There are no stand-alone nodes in the network. All the nodes are connected to the network and participate in the network activities.
  - No malicious nodes are present in the network.
- Assumptions regarding network activity:
  - All nodes can be sources or sinks of flow, and have an equal chance to be either, but not both for the same session.
  - Their may be multiple instances of sessions with identical source and sink nodes.
  - No specific routing method is implemented. Paths are selected on a lowest hop count, and a rudimentary link-state selection criteria.
  - All paths selected for the network is single-path. No multi-paths exist for a single session of flow.
- Assumptions regarding the environment:
  - The environment is *closed*. The only interference introduced into the network are co-channel interference.

3.3. Network and data representation

3.3.1. Network setup procedures

The setup procedures of the networks that were investigated by the model, focussed on the physical environment considerations of the scenario description. Table 3.1 gives the parameter settings utilised in these procedures.

Table 3.1.: Parameters and values for the network setup procedures

	Parameter:	Value:	Description:
1.	c	$3 \times 10^{8}\,m.s^{-1}$	Speed of light
2.	f_c	$2.4\,GHz$	Carrier frequency
3.	λ	$\frac{c}{f_c}$	Wavelength
4.	R_{tr}	$11\,Mbps$	Transmission rate
5.	$P_{max}(mW)$	$2.5\,mW$	Maximum transmit power[1]
6.	$P_{max}(dBm)$	$10\log(P_{TX\,max}(mW))\,dBm$	Maximum transmit power
7.	$P_{r\,TH}(dBm)$	$-85\,dBm$	Attenuation sensitivity threshold at receiver[1]
8.	$P_{r\,TH}(mW)$	$10^{(\frac{P_{r\,TH}(dBm)}{10})}\,mW$	Attenuation sensitivity threshold at receiver
9.	$P_{r0}(mW)$	$\frac{P_{max}(mW)\,\lambda^{2}}{(4\pi)^{2}}\,mW$	Closed-in reference point receiving power
10.	$P_{r0}(dB)$	$10\log\left(P_{r0}(mW)\right)\,dBm$	Closed-in reference point receiving power
11.	α_{r0}	3	Closed-in reference path loss coefficient ($\alpha = 2$ in free-space)
12.	α	3.25	Indoors path loss coefficient[2]
13.	σ	5.1	Indoors shadowing deviation[2]

[1] Characteristic value for IEEE802.11 [39] network devices.

[2] See [25] for a study on the parameter values of the indoors log-distance path loss model.

3.3.1.1. Placing the nodes

The node placement routine was developed to place the nodes of the investigated WMN in an evenly distributed mesh topology, with no overlapping nodes. The size of the network specified the size of an underlying grid used in the placement routine. A placement radius were declared for each intersection of the grid, and nodes were placed one by one at random (x, y) coordinates within this radius. The selected (x, y) coordinates were stored along with the identification number of the specific node placed at these coordinates, as the *coordinates-nodes* vector.

3.3.1.2. Calculating the maximum transmission distance, d_{max}

From aspects of the the physical environment for the described model scenario, the signals between nodes were affected by the obstructions implicated by the indoors setup, such as refraction through the walls, reflection from surrounding walls and the floor in each room, absorption, etc,. In practice, these degrading effects restrict the propagation of the RF signals to a maximum physical distance, after which the power level of the information contained in the signal cannot be distinguished from surrounding noise.

A calculated maximum transmission distance, d_{max}, was utilised by the model in order to represent this physical barrier in the network setup procedures. The calculation considers the maximum transmit power from the setup parameter values in Table 3.1, and relate this power level to the attenuation sensitivity threshold at the receiver, subject to indoors log-distance path loss. Through algebraic manipulation, the distance are calculated as follows:

$$d_{max} = \left(\frac{\lambda^2}{(4\pi)^2}\right)\left(\frac{P_{max}(mW)}{P_{TH}(mW)}\right)^{\dfrac{1}{\alpha_{r0}}} \tag{3.1}$$

3.3.1.3. Calculating the receiving power strength

The IEEE 802.11 standard [39], defines a rating mechanism by which RF energy is to be measured by the wireless device, called the Received Signal Strength Indicator (RSSI). Vendors of IEEE802.11 network devices, rate the RSSI value between 0 and a specific maximum RSSI value, the RSSI$_{max}$ [40]. Section 14.2.3.2 of the IEEE802.11 standard defines the RSSI metric, RXVECTOR RSSI. The significant statements made by the standard are quoted as follows: "*The RSSI is intended to be used in a relative manner. Absolute accuracy of the RSSI reading is not specified.*" [39]. This entails that there is no stipulation with regards to the relationship between the RSSI value and any particular energy measurement.

Therefore, the model design had the freedom to calculate the RSSI in a manner regarded as appropriate for the model. A RSSI$_{max}$ value of 1 was implemented, indicating 100% strength. A second RSSI reference point, RSSI$_{min}$, was assigned to the model indicating the strength of a signal at the maximum transmission distance, d_{max}. The attenuation sensitivity threshold at this distance from the transmitter was -85 dBm (see Table 3.1), and from the granular RSSI values table used by CISCO devices this power level yields a RSSI of 27%. Therefore, the value assigned to RSSI$_{min}$, was RSSI$_{min} = 0.27$.

The power level at the receiving nodes of the links, were calculated in accordance with the indoors log-distance path loss model with an added shadowing deviation [25]:

$$P_r(\text{at } d) = P_{r0}(\text{at } d_0) - 10\alpha \log(d) + X_\sigma \tag{3.2}$$

The RSSI at the receiving nodes of the links was calculated by utilising linear interpolation between RSSI$_{min}$ and RSSI$_{max}$:

$$\text{RSSI} = \text{RSSI}_{min} + (P_r - P_{rTH})\frac{(\text{RSSI}_{max} - \text{RSSI}_{min})}{(P_{max} - P_{rTH})} \tag{3.3}$$

3.3.1.4. Creating data sets for the elements

The steps necessary to represent the networks investigated by the model are given by Algorithm 1. The data sets for the elements represented by the network setup procedures are:

- The set of nodes, V.
- The set of links, E for every link $e \in E$.
- The fractional quality of each link indicated by the set of RSSI factors, E_{RSSI}

- The node-node adjacency matrix, N. By using either the top or bottom triangle of N, a sparse matrix was created that represent the graph of the network, $G = (V, E)$, in the form of an adjacency list.

Algorithm 1 Steps in network representation

```
Place the nodes in $V$
Calculate $P_{max}$ from Table 3.1
Calculate $d_{max}$ from Equation 3.1
Create $(n \times n)$ node-node adjacency matrix, $N$, with zeros
loop $\forall \ i \in V$
    loop $\forall \ j \in V$
        if $i \neq j$ do
        Calculate distance between $i$ and $j$, $d_{(i,j)}$ using the coordinates-nodes vector
        Calculate received power at $j$, $P_r$(at $d_{(i,j)}$) from Equation 3.2
            if $d_{(i,j)} \leq d_{max}$ do
            $N(i,j) = 1$
            Calculate $E_{RSSI}(i,j)$ from Equation 3.3
            else do
            $N(i,j) = 0$
            endif
        endif
    end loop
end loop
```

3.3.2. Representing the network load

The procedures involved in representing the load of the networks that were investigated by the model, focussed on the aspect of multiple sessions presented by the scenario, as well as the impact that these sessions have on each other.

3.3.2.1. Path calculations

Each investigated network was assigned with a network load, i.e. the number of concurrent sessions, s, in the network. A vector containing the concurrent sessions is defined as ξ and $s \in \xi$. Each session, s, was created by the following steps:

1. Create an array of possible sources, S. The *possible sources* differ for each type of WMN selected in the running configuration:
    - *Pure WMNs*: All the nodes may be selected as sources, therefore $S = V$
    - *Hybrid WMNs*: One fifth of the nodes in V are reserved for the router backbone R, therefore $S = V - R$. The nodes in R are selected on a random basis beforehand.
    - *Access WMNs*: One fifth of the nodes in V are reserved as APs A, therefore $S = V - A$. The nodes in A are selected on a random basis beforehand.
2. Create an array of possible sinks, T. The *possible sinks* differ for each type of WMN selected in the running configuration:
    - *Pure WMNs*: All the nodes may be selected as sinks, therefore $T = V$

   - *Hybrid WMNs*: One fifth of the nodes in V are reserved for the router backbone R, therefore $T = V - R$. The nodes in R are selected on a random basis beforehand.
   - *Access WMNs*: One fifth of the nodes in V are reserved as APs A. The nodes in A are the ultimate destinations in the access WMN, therefore $T = A$. The nodes in A are selected on a random basis beforehand.
3. For each session s, select a random source-sink pair from S and T respectively. The following condition must hold: source $\neq$ sink.

In accordance with the definition of hybrid WMNs, the network representation, G, is updated with direct links between the nodes in the router backbone R. These links are assigned with a RSSI of 1.

With the source and sink nodes known for each session, the paths were calculated as the consecutive set of links between each source and sink node. The built-in Bellman-Ford algorithm from MATLAB®, together with additional weight vectors, was utilised in the model to calculate the paths for the network. For every session s, two distinct paths were calculated, namely the distance vector path, PDV_s, and the link-state path, PLS_s.

- **Distance vector paths, PDV_s:** The distance vector paths, are selected by using the Bellman-Ford algorithm to determine the shortest paths between the source and sink node. The main metric used is *hop-count*. If two paths with equal hop distances exist, the path with the highest sum of RSSIs are selected.
- **Link-state paths, PLS_s:** The link-state path of a session, is calculated by implementing a link-budget scheme. Again the Bellman-Ford algorithm searches for the path, with the *lowest total cost* as metric. Once a link in the network is utilised in *any path* calculated thus far, the link is penalised with additional cost, forcing the path search algorithm to look at other link options in the next path calculation.

3.3.2.2. Link interference

The paths of multiple sessions present in the network, may cause co-channel interference on each other. Consider the example given by Figure 3.2.

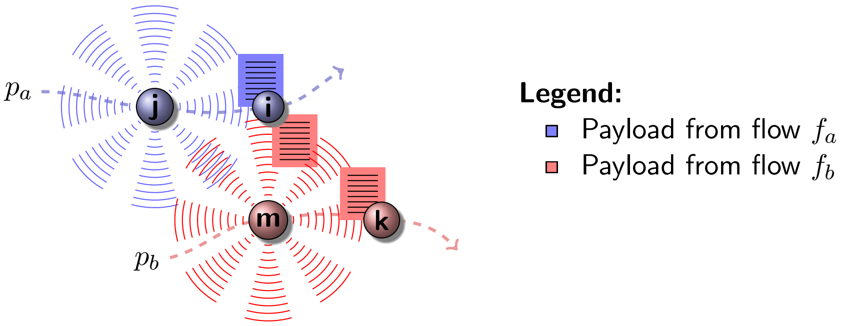


Figure 3.2.: Co-channel interference from multiple concurrent sessions

A certain link, $e(j,i)$, exists on path p_a carrying a load of flow, f_a. A neighbouring link, $e(m,k)$, on path p_b exists with an arbitrary flow f_b. Node i is a neighbour of node m, and will therefore be able to receive the broadcast of flow f_b. This will cause interference in the transmission of f_a over $e(j,i)$. Algorithm 2 was used to determine a vector $M_{i,s}$, the collection of nodes that will cause co-channel interference on the receiving node i for session s.

Algorithm 2 Steps in determining interfering neighbouring nodes

```
1. Calculate transmitting nodes on all links, $V_{TX}$
2. loop $\forall\ i \in\ V$
3. Calculate all neighbours for $i$, $H_i$
4. end loop
5. loop $\forall\ s \in\ \xi$
6.     loop $\forall\ i \in\ e(j,i)$ on $p_s$
7.      Get $H_i$
8.      Find all $m$ where $H_i\ \cap\ V_{TX}$
9.          if $m\ \neq\ j$ do
10.          Put $m$ in $M_{i,s}$
11.         endif
12.     end loop
13. end loop
```

3.3.2.3. Finding Network Coding opportunities

The procedure to find Network Coding opportunities within the paths of the multiple concurrent sessions, are concerned with finding link patterns that represent two separate flows from opposite directions through a common intermediate node. Consider two flows, f_a on path p_a and f_b on path p_b. A Network Coding opportunity exists, to combine these flows, at a common node i if the following link pattern occur:

$$p_a = \{\ldots,(j,i),(i,k),\ldots\}$$
$$p_b = \{\ldots,(k,i),(i,j),\ldots\}$$

The procedure examines the paths of all the concurrent sessions in order to find j,i,k and k,i,j matches. The Network Coding principle utilised in the model is defined as follows:

> *Two flows, f_a and f_b, entering a node i from opposite directions, are combined to form a single flow, f_{ab}. The process of combining flows f_a and f_b at node i, is done at a constant Network Coding rate, R_i, in order to combine an equal amount of bits from f_a and f_b per time interval. The rate of f_{ab} leaving node i, must adhere to the law of conservation of flow. Therefore, the respective rates of f_a, $R_{j,i}$, and f_b, $R_{k,i}$, are exactly one half of the rate of f_{ab}. The combined flow can also be expressed as f_{ba}*

In equation form:

$$\begin{aligned} f_a(R_{j,i}) + f_b(R_{k,i}) &= f_{ab}(R_i) \\ &= f_{ba}(R_i) \end{aligned} \tag{3.4}$$

therefore,

$$\begin{aligned} f_a(R_{j,i}) &= 0.5\, f_{ab}(R_i) \\ &= 0.5\, f_{ba}(R_i) \end{aligned}$$

and

$$\begin{aligned} f_b(R_{k,i}) &= 0.5\, f_{ba}(R_i) \\ &= 0.5\, f_{ab}(R_i) \end{aligned}$$

Not all the j, i, k and k, i, j matches on the session paths are *valid* Network Coding opportunities. In order to distinguish between valid and invalid Network Coding opportunities, the following explanations are provided.

VALID NETWORK CODING OPPORTUNITIES

The first case to consider, is the case where more than one Network Coding opportunity exist at a single intermediate node i. The deciding factor for a Network Coding opportunity is whether two flows from opposite directions pass through a common intermediate node. As long as this condition holds, multiple Network Coding opportunities are possible. Figure 3.3 gives an example of such a case:

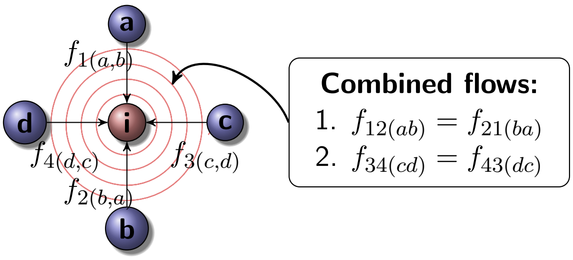


Figure 3.3.: Four incoming flows, and two combined outgoing flows

In this case four separate flows enter node i. Two Network Coding opportunities exist: Flows 1 and 2 are in opposite directions between nodes a and b, and flows 3 and 4 are in opposite directions between nodes c and d. Node i can combine both these cases, with a cost to scheduling since it cannot combine al four streams simultaneously. Flows 1 and 2 are combined separately from flows 3 and 4, and two combined flows are broadcast by node i.

INVALID NETWORK CODING OPPORTUNITIES

A combined flow is dependent on the ability of the receiving nodes to decode the Network Coded flow. Since each node only buffers the content of its own sent message, it can only decode a message with *one* other variable message. Combining more than two flows into one single flow is an invalid Network Coding opportunity, and Figure 3.4 gives an example of such a case.

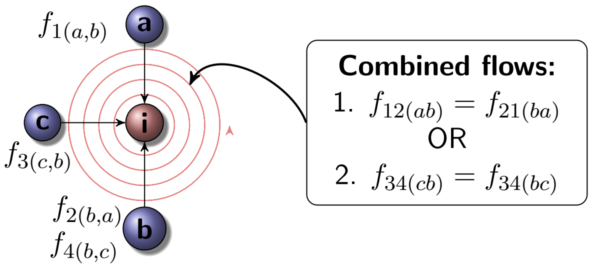


Figure 3.4.: Case for three incoming flows, and only one valid combined flow

3.3.2.4. Creating data structures representing the network load elements

The information in terms of the paths for all the concurrent sessions present in the investigated network, where Network Coding opportunities exist along these paths, as well as co-channel interference, are captured in the form of flow coefficients contained in the following data structures:

- PEM-DV and PEM-LS: The $(s \times e)$ paths-links matrix data structures, for distance vector paths and link-state paths respectively. The rows in each matrix represent the concurrent sessions, ξ present in the network. The columns represent the indices from the links vector, E. For each row, s, a value of 1 is assigned to each of the column indices representing the consecutive links of path p_s. The column indices that represents links *not* present in p_s, remains 0.
- XORed-PEM-DV and XORed-PEM-LS: The coded variation $(s \times e)$ paths-links matrix data structures, for distance vector paths and link-state paths respectively. These structures are identical to the PEM-DV and PEM-LS structures, with additional information included for the Network Coding opportunities. For each j, i, k and k, i, j match found for two separate paths, p_a and p_b, the column indices for the (j, i) and (i, k) links are updated to 0.5. This is done for both rows a and b.
- MEM-DV and MEM-LS: The $(s \times e)$ interfering nodes-links matrix data structures, for distance vector paths and link-state paths respectively. The rows in each matrix represent the concurrent sessions, ξ present in the network. The columns represent the indices from the links vector, E. For each column e representing $e(j, i)$, the rows of s in $M_{i,s}$ are assigned the value 1. The remaining indices are 0.

Additional to the data structures listed above, the procedures involved to represent the network load yielded the following data sets:

- The set of concurrent sessions, ξ.
- The set of source nodes for all the sessions, S.
- The set of sink nodes for all the sessions, T.
- The set of distance vector paths, PDV_s for each session $s \in \xi$.
- The set of link-state paths, PLS_s for each session $s \in \xi$.
- The co-channel interfering nodes, $M_{i,s}$ containing m nodes causing interference on each $i \in V$ for each session $s \in \xi$.

3.4. The network flow programming problem

The objective of the model is to achieve the maximum aggregated throughput for accumulated flows of concurrent sessions present in the investigated network. The *network flow programming problem*, is the network flow problem representation of the investigated network, formulated as a linear programming problem. Table 3.2 summarises the data sets and parameters utilised by the procedures involved in the network flow programming problem development.

Table 3.2.: Network data elements in the model

	Symbol:	Description:
1.	$G(V,E)$	Network graph adjacency list
2.	V	Set of nodes
3.	E	Set of directed links
4.	E_{RSSI}	Set of RSSIs for each link
5.	H_i	Set of neighbouring nodes for node i
6.	$M_{i,s}$	Set of m number of interfering nodes on node i for session s
7.	ξ	Set of sessions
8.	s_i	Session i contained in ξ
9.	S_i	Source node for session i
10.	T_i	Sink node for session i
11.	λ_i	Flow of session i
12.	$f_{e_{ji}}$	Flow on link from node j to node i
13.	f_i	Flow at node i
14.	C	Channel Capacity
15.	TP	Aggregated (system) throughput

3.4.1. Goal function

For all the session flows, $\lambda_{1\ldots s}$, the maximum aggregated throughput is reached at:

$$TP = \max \sum_{i=1\ldots s} \lambda_i \qquad \forall\, \lambda_i \,\in\, \xi \tag{3.5}$$

3.4.2. Constraint functions

The constraint functions of the network flow programming problem are the mass balance constraint, the capacity constraint and the Network Coding constraint.

3.4.2.1. Mass balance constraint

The law of conservation of flow [38], implies that all the accumulated flow entering a node, must equal the accumulated flow that exit the node. For notational purposes, the model adopts a **receiver-based approach**. At every intermediate node on the path of a session, the resultant flow value is zero. The source node on the path, S_i for $s = i$, is the originator of the data transfer and will not receive any flow. But flow will leave S_i, so the resultant flow for S_i will be *negative*. The sink node, T_i for $s = i$, is the ultimate receiver on the path and the message is not conveyed any further. There will only be flow entering T_i, and no flow leaving giving it a *positive* resultant flow.

The mass-balance constraint requires flow conservation at the intermediate nodes for each session, except the corresponding source and sink nodes, and is formulated as follows:

$$\begin{aligned}\delta f_i(s) &= \sum_{\forall\,\text{incoming}\,j} f_{e_{ji}}(s) + \sum_{\forall\,\text{outgoing}\,k} f_{e_{ik}}(s) \\ &= \begin{cases} -\lambda_s & \text{if } i = S_s, \\ \lambda_s & \text{if } i = T_s, \\ 0 & \text{otherwise} \end{cases}\end{aligned} \tag{3.6}$$

where $\delta f_i(s)$ is the *resultant* flow at node i for each session $s \in \xi$.

3.4.2.2. Capacity constraint

When multiple flows arrive at a router, it does not process these streams simultaneously. The packets enter a buffer and are processed and forwarded on a scheduled basis. To capture this delay in packet transfer, a commonly accepted model for scheduling is given by Kumar et. al. [41]:

$$f_{ij} + \sum_{\substack{\forall\, k \\ \forall\, l}} f_{kl} \;\leq C \tag{3.7}$$

where f_{ij} is the flow rate on link (i, j) and (k, l) an arbitrary link interfering with (i, j). C is the capacity of the link. Note - the value of the capacity of each link, $C_{(i,j)}\ \forall\ e(i, j) \in E$, is equal to the Channel Capacity, C, since the model is based on a single channel utilisation.

However, the network flow programming problem for the model, has to account for multiple commodities. Equation 3.7 is therefore modified to the following capacity constraint:

$$f_i(s) = \sum_{s\,\in\,\xi} \left(\sum_{\forall\,\text{incoming}\,j} f_{e_{(j,i)}}(s) \right) + \sum_{\substack{\forall\, m \\ \forall\, k}} f_{e_{(m,k)}} \;\leq C \tag{3.8}$$

Equation 3.8 entails that the flow on a link, $e_{(j,i)}$, of a single session, s, must share the capacity of that link with **all** the other concurrent sessions, as well as all other flows from arbitrary links, $e_{(m,k)}$, that interfere with that link.

3.4.2.3. Network Coding constraint

From the Network Coding principle defined in section 3.3.2.3, the Network Coding constraint for the network flow programming problem is stated as follows:

For two separate session flows, λ_a and λ_b, moving through an intermediate node, i, in opposite directions, $p_a = \{\ldots, (j,i), (i,k), \ldots\}$ and $p_b = \{\ldots, (k,i), (i,j), \ldots\}$, there exists a Network Coding rate, ϕ, at node i:

$$\begin{aligned} \phi_{j,i} &= 0.5\lambda_a \\ &= 0.5\lambda_b \\ &= \phi_{i,k} \end{aligned} \tag{3.9}$$

3.4.3. Expressing the network flow programming problem

With the system goal function defined, together with all the relevant constraint functions, the network flow programming problem can be formally stated by the equations below.

Solve:

$$TP = \max \sum_{i=1\ldots s} \lambda_i \qquad \forall\ \lambda_i\ \in\ \xi \quad \text{(Equation 3.5)}$$

Subject to:

$$\sum_{\forall\ \text{incoming}\ j} f_{e_{ji}}(s) + \sum_{\forall\ \text{outgoing}\ k} f_{e_{ik}}(s) \quad \text{(Equation 3.6)}$$

and

$$\phi_{j,i} = 0.5\lambda_a = 0.5\lambda_b \quad \text{(Equation 3.9)}$$

Bounded by:

$$f_i(s) = \sum_{s\,\in\,\xi} \left(\sum_{\forall\ \text{incoming}\ j} f_{e_{(j,i)}}(s) \right) + \sum_{\substack{\forall\, m \\ \forall\, k}} f_{e_{(m,k)}} \ \leq C \quad \text{(Equation 3.8)}$$

The network flow programming problem of the model, is classified as *mathematically equivalent to multi-commodity integer problems*. These type of problems are usually dimensionally large, and therefore difficult to solve due to its complexity [10], [11], [12].

3.5. Complexity analysis

The measure of complexity for the network flow programming problem were determined by a time complexity analysis. An estimate of the upper bound for the number of iterations needed, to obtain an acceptable solution for the problem, is provided by the analysis [10], [29]. This estimate is dependent on the size of the problem, in terms of the number of variables to solve. The symbols utilised in the complexity analysis follow the general notation for computational complexity theory, and are given as follows:

- The number of nodes in the network, n.
- The number of links in the network, m.
- The number of constraints k.

The procedures involved in performing the complexity analysis, are divided into three stages:

1. Analyse the complexity of determining a single flow value.
2. Analyse the complexity of executing multiple iterations.
3. Analyse the complexity of obtaining a sub-optimal solution.

The complexity analysis in stage (3) are subjected to the sub-optimal solution, since the optimal solution is achieved by executing multiple iterations, which is accounted for by stage (2).

3.5.1. Single flow value complexity estimate

Problems concerned in finding the maximum flow through a network, are based on the principle of iterative increments of an augmenting path method[2]. At the start of the problem, the maximum value for the flow is unknown. The goal is to select a proper augmenting path for each session, and to allow for opportunities to increment the flow at each iteration with the maximum number of units possible.

For n nodes in the network, the time needed for each augmenting path is $\mathcal{O}(n^2)$. In the worst-case analysis, the flow value is not bounded by n and m. The complexity is (Cm) for some set of capacity values for m, which is a pseudo-polynomial time. The iterations are not bounded and an optimum solution is never obtained. Therefore, the augment paths is restricted to find the *shortest* path. The complexity becomes $\mathcal{O}(nm(n+m))$. When flow is incremented simultaneously in more than one path, the time estimate is $\mathcal{O}(n^2m)$ for dense networks, and $\mathcal{O}(nm\log n)$ for sparse networks.

3.5.2. Multiple iterations complexity estimate

The multi-commodity property of the problem, implies that each commodity is limited by a set of constraints. This results in a vector of constraints present in the problem. Let the number of constraints be no more than $m-1$. Each link is examined at least once, but assumed to be examined several times. The number of examinations is bounded by k. Therefore, the complexity does not exceed $k(n+m)$, or $\mathcal{O}(n+m)$.

3.5.3. Sub-optimal solution complexity estimate

The sub-optimal solution is obtained in one iteration. Each link is examined for at least one constraint, and flow is updated. The complexity is $\mathcal{O}(n+m)$. For a worst-case analysis, all the links are bounded by the constraints of the problem, and the complexity is $\mathcal{O}(n+m+k)$.

---

[2] See appendix B.1 for a description of the general augmenting path method.

3.5.4. Final remarks on the complexity

Let the condition of finding the optimal solution be valid. The time needed to compute the maximum flow value over the shortest paths, for one iteration is $\mathcal{O}(nm(n+m))$. The time needed to find the optimal solution, subjected to multiple simultaneous paths, is $\mathcal{O}(n^2m)$for dense networks, and $\mathcal{O}(nm\log n)$ for sparse networks. The conclusion of the complexity analysis is that the network flow programming problem is a non-deterministic polynomial-time problem, since the time needed to solve the problem is bounded by a polynomial function of n and m, and therefore NP-hard[3].

3.6. Solving the optimal aggregated throughput

Since the network flow programming problem of the model is NP-hard, an appropriate solving method for the model was a heuristic[4] approach. The selected technique for the model was the Simulated Annealing method[5]. The reader is encouraged to refer to the detailed description of the method, found in appendix B.3.1.

3.6.1. Aspects of Simulated Annealing

Simulated Annealing is a heuristic, probabilistic, solving method, independently developed by Kirkpatrick, Gelett and Vecchi (1983) [44], and Cerny (1985) [45]. In short, its principle is to create random configurations of possible variable values, and move through the search domain to find "better" values, by comparing the goal function output of one configuration to the next. To prevent the solution to get fixated at a local minima or maxima, it also implements a *probability function* to enforce a transition.

This procedure is very flexible, since a custom probability function may be implemented, the maximum number of iterations for the algorithm may also be selected. The algorithm as such is easy to implement, with efficient documentation of previous implementations for support. Although it is not a deterministic method which will produce absolute results, the parameters utilised in the method can be fine tuned to offer solutions close enough to the absolute answers (in a fraction of the number of iterations). These parameters are:

- The **probability function**, p, used to establish a threshold where a configuration are *accepted* as the new "best" match, even though its goal function output is not better than the current configuration. This procedure allows the method to move towards a global optimum and not get stuck at a local optimum.
- The **temperature** parameter. This parameter decrease with every iteration until the terminating value is reached. The decreasing factor is determined by the *control parameter*.

Table 3.3 lists the basic elements and their description as it is used during the Simulated Annealing method.

---

[3] The work presented by [29], [42] and [43] prove the NP-hardness of multi-commodity integer programming problems.

[4] See Chapter 2, Section 2.2.5, for the explanations on deterministic and heuristic methods.

[5] See appendix B.3.5 for a comparison between different heuristic methods.

Table 3.3.: Defining the elements used for Simulated Annealing in the model

	SA element:	Defined for this model:
1.	Search space	$0 \leq \beta \leq C$ $\beta \in \Re$
2.	The configuration: a finite set of candidate values for λ_i	$\mathbb{C}$
3.	Initial configuration	$\mathbb{C}(0) \in \mathbb{C}$
4.	Neighbouring configuration	$\mathbb{C}'$ where $\mathbb{C}' \subset \mathbb{C} - \{i\}$ $\forall\, i \in \mathbb{C}$
5.	Acceptance probability function	$p = e^{\frac{\lambda(\mathbb{C}') - \lambda(\mathbb{C})}{\tau}}$
6.	Aggregated throughput (Equation 3.5) under configuration *c*	$\lambda(c)$
7.	Temperature parameter	τ
8.	Update factor for τ	θ
9.	Starting temperature	$\tau(0)$
10.	Terminating temperature	T_{min}

Settings for some these parameters are:

- θ is the multiply-decrease update factor for τ, and empirically valued between 0.85 and 1. For the model the value $\theta = 0.86$ was used.
- The initial temperature is selected as $\tau(0) = 10C$, where *C* is the channel capacity of each link.
- The terminating or minimum temperature value was set empirically to $T_{min} = 10^{-6}$.

3.6.2. The acceptance probability, p

An important situation to *avoid* during the solving routine, is for the algorithm to fixate at a configuration for a local optimum - this is a configuration which results in outputs less than the global maximum, yet it is still higher for any of its neighbouring configurations. To prevent this, an *acceptance threshold* is included in the solving procedure that will cause a transition from the current configuration, $\mathbb{C}$, to a neighbouring configuration, $\mathbb{C}'$, even though the aggregated throughput at $\mathbb{C}'$ is less than at $\mathbb{C}$. By enforcing this transition, continued iterations will lead to the global optimum.

This acceptance threshold is a probability function, p, and is dependent on the values for the aggregated throughput values for both the configurations, $\mathbb{C}$ and $\mathbb{C}'$. The requirements of the p in terms of $\mathbb{C}$ and $\mathbb{C}'$ are given below:

- The value for p must be > 0 (non-negative) when $\lambda(\mathbb{C}') > \lambda(\mathbb{C})$. This will ensure a transition to a lower output configuration when the algorithm approach a *local* optimum.
- When $\tau \Rightarrow 0$, the following must be true for p:
  - $p \to 0$ if $\lambda(\mathbb{C}') > \lambda(\mathbb{C})$
  - $p > 0$ if $\lambda(\mathbb{C}') < \lambda(\mathbb{C})$

This will ensure an increased favour for “downhill” transitions to better solutions as τ approaches zero.

From this requirements, the acceptance probability function for this experiment is defined as:

$$p = e^{\frac{\lambda(\mathbb{C}') - \lambda(\mathbb{C})}{\tau}} \tag{3.10}$$

3.6.3. The Simulated Annealing iteration

In order to find the maximum aggregated flow of the network investigated by the model, the following three algorithms are executed:

- Algorithm 3: Calculating the initial configuration, $\mathbb{C}(0)$.
- Algorithm 4: Selecting a neighbouring configuration, $\mathbb{C}'$.
- Algorithm 5: Determine the goal function output for the current configuration.

Algorithm 3 Calculating the initial configuration $\mathbb{C}(0)$

```
loop $\forall\ s \in\ \xi$
    Calculate paths from $S_s$ to $T_s$
end loop
loop $\forall$ receiving nodes $\in\ V$
    Find all Network Coding opportunities and update path-links matrix
end loop
repeat
    Increase flow with $\triangle\ \forall\ i\ \in \lambda_i$
until Links are saturated $\forall\ e \in\ E$
```

Algorithm 4 Selecting neighbouring configuration $\mathbb{C}'$

```
repeat
    loop $\forall\ \lambda_i\ \in\ \mathbb{C}$
        Select random values for each flow from $\beta$
    end loop
until Constraints are met.
```

Algorithm 5 Simulated Annealing algorithm for the model

```
Determine the initial configuration $\mathbb{C}(0)$ from algorithm 3
$\mathbb{C}\ \leftarrow\ \mathbb{C}(0)$
repeat
    Generate neighbour configuration $\mathbb{C}'$ from algorithm 4
    if $\lambda(\mathbb{C}') > \lambda(\mathbb{C})$ then
        $\mathbb{C} = \mathbb{C}'$
    else if $p(\mathbb{C}') > random[0,\ 1)$ then
        $\mathbb{C} = \mathbb{C}'$
    endif
    $\tau = \theta\ \centerdot\ \tau$
until $\tau\ <\ T_{min}$
```

3.7. Model implementation

The procedures involved in the model development, namely the representation of network elements, network load and the network flow programming problem all contributed to the aggregated throughput calculation of a single network under investigation. The model investigated multiple instances of WMNs variations in order to produce generalised results for the WMN system. These variations included the following:

- **Network size:** Network sizes were from the range (10, 15, 20, 25, 30, 35), indicating the number of nodes in the network.
- **Network load:** The load for each network varied from 5 to 100 concurrent sessions, incremented with 5 sessions at a time.
- **Network type:** Three types of WMNs were investigated, namely pure WMNs, hybrid WMNs and access WMNs.
- **Network Coding:** For each investigated network, the aggregated throughput was calculated for the case without implementing Network Coding, as well as the aggregated throughput for same network with a Network Coding principle implemented.
- **Forwarding path method:** There were two variations for each unique network pertaining to the method utilised to determine the forwarding paths of the message flows. The respective methods were the distance vector and link-state method.

The implementation of the model constituted of executing the following events:

1. Create a mesh network, G, with n number of nodes, V.
2. Utilise an indoors RF channel model in order to establish wireless links, E, between the nodes in V, together with the quality of these links, E_{RSSI}.
3. Select a random set of source nodes, S, and sink nodes, T, for s number of concurrent sessions in ξ.
4. For each session s, calculate the distance vector and link-state paths between S and T.
5. Compute the maximum value of the aggregated throughput for network G.

In this chapter, a model was developed to investigate the aggregated throughput of WMNs. This model was developed from the description of the model scenario, in section 3.1, and the assumptions, in section 3.2.

Section 3.3 described the network setup procedures of the model. These procedures were utilised to create the data elements, such as the graph representation of the network, the nodes, links and link qualities of the investigated network instance. A maximum transmission distance, together with the log-distance path loss model were implemented in order to create the links of adjacent nodes. The quality indicators were derived from the different power calculations at the transmitting and receiving nodes of each link.

Two paths for each concurrent session present in the investigated network were determined, i.e. the distance vector and link-state path. The concern for multiple paths, due to multiple concurrent sessions, were co-channel interference caused by flow existing in adjacent links. The broadcast nature of the nodes, implied that every node will receive transmissions from neighbouring nodes, causing a delay in the scheduling effect at each node. Additionally, the multiple paths also yielded valid Network Coding opportunities at certain nodes.

The information regarding the paths of the concurrent sessions, the interference caused by neighbouring nodes and the existence of Network Coding Opportunities, were contained in the following network data structures: The path-links matrix for distance vector paths and link-state paths, for the case where Network Coding is implemented and where it was not implemented; and the links-interference matrix. Each matrix contains the multiplication coefficients related to every link in every session. These coefficients were utilised during the flow calculations of the solving algorithm.

The network programming problem of the model, developed in section 3.4, represented the network operations of the investigated network. The goal function was developed to calculate the aggregated throughput for multiple concurrent sessions. Each session path was subject to the mass balance constraint, entailing that the conservation of flow should hold at each node on the path. The flows on the link were augmented with the maximum amount of flow permitted by the capacity and coding constraints of the network flow programming problem.

The developed network flow programming problem was considered as the mathematical equivalent of a multi-commodity integer problem. A complexity analysis in section 3.5, revealed that the problem was NP-hard. Therefore, the problem were solved by a heuristic algorithm. The selected heuristic solving method was Simulated Annealing. Section 3.6 showed how the acceptance probability of this method were selected, and gave the algorithms needed in order to solve the network flow programming problem.

4. Results

In this chapter the results obtained from the developed model implementation in the previous chapter are presented. Section 4.1 describes the procedures involved in processing the raw data into unique data sets for each investigated network instance, as well as a description of the different tests involved in the analysis of the data. The graphical presentation of the data is given in section 4.2, followed by an interpretation of the presented results in section 4.3.

4.1. Preparation

The goal of the model developed in the previous chapter, is to investigate the effect in the aggregated throughput of WMNs due to Network Coding. In order to achieve this goal, the results from the model implementation are analysed in terms of:

- Throughput in relation to network size; and
- Throughput in relation to network load.

4.1.1. Data sets

The output results from the model implementation, were processed and organised into different data sets for each unique problem instance investigated by the model. In total, a number of 288000 unique problem instances were investigated. These problem instances were setup in terms of the following parameters:

- The size of the network, ranging from (10 15 20 25 30 35) number of nodes;
- The WMN type, namely pure WMNs, hybrid WMNs and access WMNs;
- Whether the network implements Network Coding, or not;
- The method utilised to determine the paths of the sessions present in the network; and
- The number of concurrent sessions in the network, ranging from 5 to 100 concurrent sessions and increased in incremental steps of 5 sessions.

The model yielded 100 problem instances for each combination of the setup parameters listed above, resulting in 100 values of calculated aggregated throughput. These values are the content of the data sets presented in this chapter. For graphical presentation of the results, the following values are calculated:

- Aggregated throughput (the mean of each data set); and
- Single session average throughput (The mean of each data set divided by the number of loads constituting the data set).

From the central limit theorem [46], a normal distribution for a sample is assumed if the sample size is greater than 30. Since 100$\gg$30, it is assumed that the 100 values of each data set yielded by the model is normally distributed. The mean values utilised in the analysis of the results, are assumed to be acceptable approximations of the unknown population mean values for WMN throughput calculations.

4.1.2. Data analysis

In order to analyse the data obtained through the model implementation, two tests were carried out:

- **Test 1: network level performance.** The aggregated throughput values of all the network sizes, at all the different loads are examined. The purpose of this is to compare the performance of networks implementing Network Coding in relation to the performance of networks that do not implement Network Coding.
- **Test 2: session level performance.** In this test the average throughput of single sessions in networks implementing Network Coding are examined, compared to the single sessions of networks that do not implement Network Coding.

4.2. Results and observations

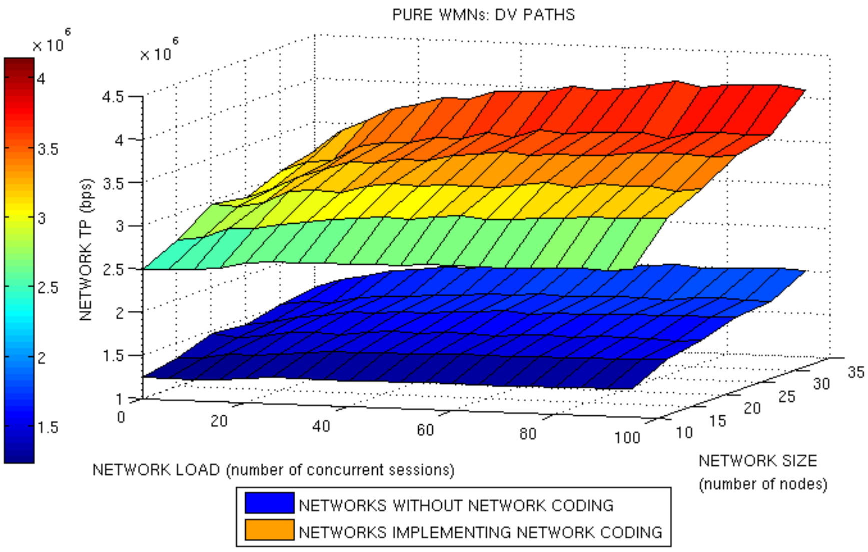
The results of Test 1 and Test 2 described in the previous section are presented, as well as a description of additional relevant observations.

4.2.1. Results of test 1

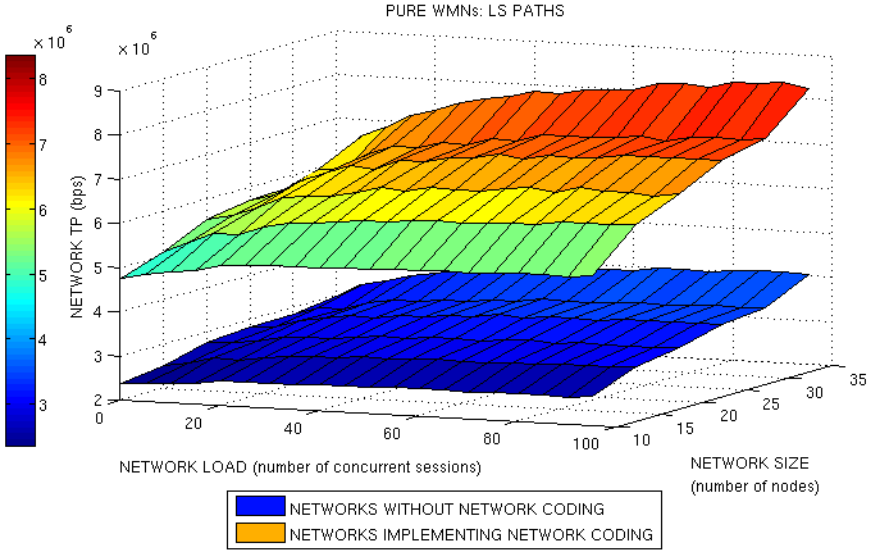
In this section, the aggregated throughput data sets of the investigated network instances are presented. Each graph represents a *surface plot* of the different network variations, in terms of Network Coding implementation and the path determination method utilised. This type of plot was selected, since all the information pertaining to the test can be captured by a single graph. The x-axis represents the different network loads, the y-axis represents the different network sizes and the z-axis represents the aggregated throughput value. A color bar is utilised, in order to get a visual perspective of the throughput values.

4.2.1.1. Pure WMNs

The mean aggregated throughput values of the investigated pure WMNs instances are presented in Figure 4.1 (a) and (b). The presented graphs represent networks that implemented Network Coding related to the same networks without a Network Coding implementation. Figure 4.1(a) presents the results for the case where paths were determined through a distance vector method, and Figure 4.1(b) presents the results where paths were determined through a link-state method.



(a) Pure WMNs with distance vector paths



(b) Pure WMNs with link-state paths

Figure 4.1.: Aggregated throughput for pure WMNs

The aggregated throughput values for networks implementing Network Coding are higher than the aggregated throughput values for networks where Network Coding is not implemented. Additionally, the aggregated throughput values for networks in which a link-state path determination method was utilised, are higher than for networks in which a distance vector path determination method was used. Table 4.1 presents a summary of critical aggregated throughput values from the graphs in Figures 4.1(a) and 4.1(b).

Table 4.1.: Data ranges for different network sizes - pure WMNs

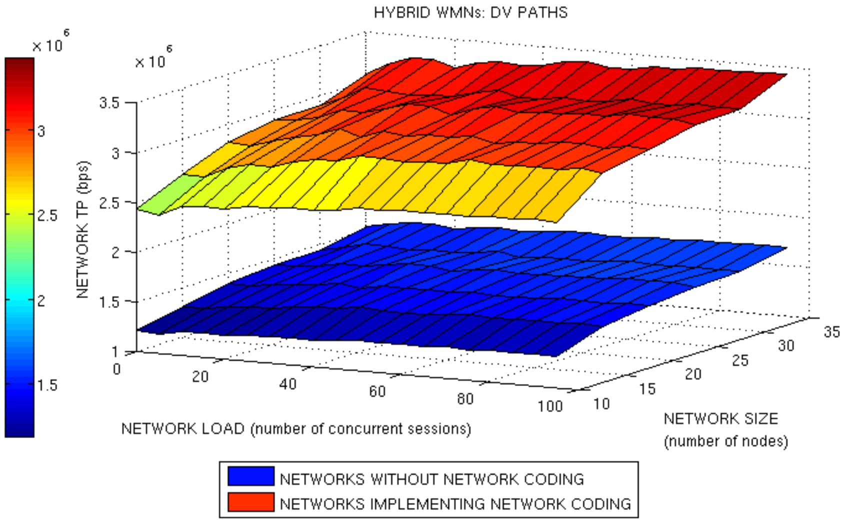
	Throughput level (Mbps)							
	@ 5 sessions				@ 100 sessions			
	Distance vector		Link-state		Distance vector		Link-state	
Size:	N:[1]	C:[2]	N:	C:	N:	C:	N:	C:
10 nodes	1.234	2.492	2.369	4.757	1.337	2.724	2.644	5.381
15 nodes	1.331	2.689	2.520	5.090	1.542	3.179	3.035	6.256
20 nodes	1.470	2.971	2.751	5.556	1.616	3.380	3.206	6.682
25 nodes	1.436	2.903	2.834	5.715	1.744	3.623	3.466	7.178
30 nodes	1.530	3.073	2.865	5.747	1.789	3.719	3.561	7.380
35 nodes	1.582	3.173	2.984	5.970	1.997	4.098	4.029	8.252

[1] N: No Network Coding.

[2] C: Network Coding implemented.

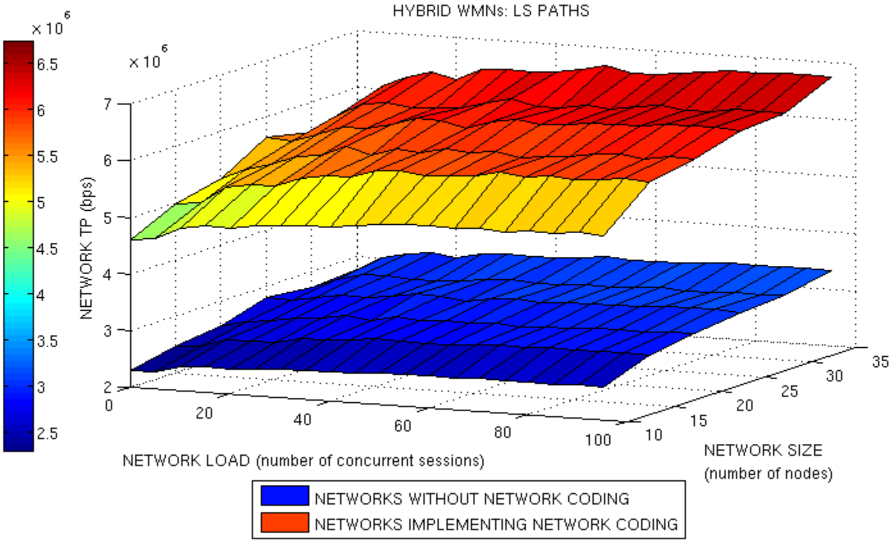
4.2.1.2. Hybrid WMNs

The same comparisons as in the previous section are performed for hybrid WMNs. Figures 4.2(a) and 4.2(b) present the aggregated throughput values. The aggregated throughput values for networks implementing Network Coding are higher than the aggregated throughput values for networks without a Network Coding implementation. Additionally, the aggregated throughput values for networks in which a link-state path determination method was utilised are higher than for networks in which a distance vector path determination method was used. However, the aggregated throughput values do not maintain a steep slope of increase as the number of concurrent sessions increase, or as the network size increase.



(a) Hybrid WMNs with distance vector paths

Figure 4.2.: Aggregated throughput for hybrid WMNs



(b) Hybrid WMNs with link-state paths

Figure 4.2.: *Continued*: Aggregated throughput for hybrid WMNs

Table 4.2 presents a summary of critical aggregated throughput values from the graphs in Figures 4.2(a) and 4.2(b).

Table 4.2.: Data ranges for different network sizes - hybrid WMNs

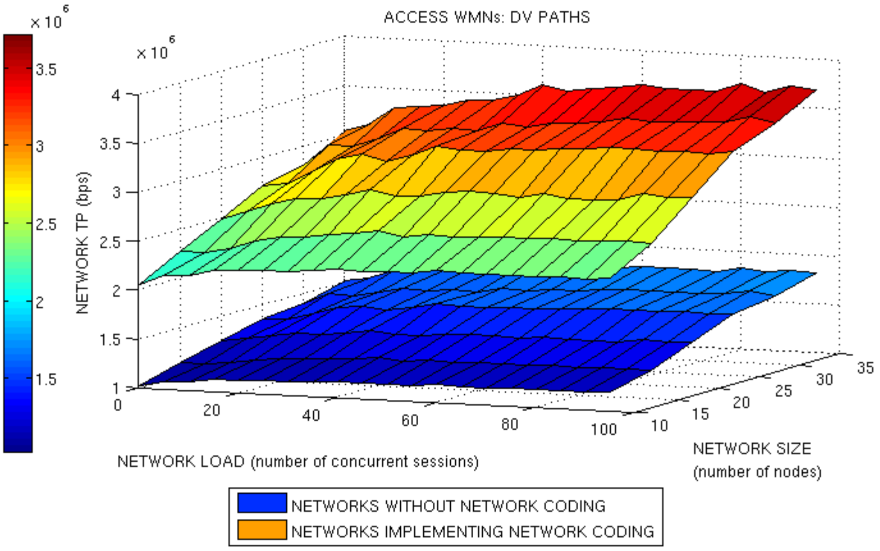
	Throughput level (Mbps)							
	@ 5 sessions				@ 100 sessions			
	Distance vector		Link-state		Distance vector		Link-state	
Size:	N:[1]	C:[2]	N:	C:	N:	C:	N:	C:
10 nodes	1.216	2.436	2.295	4.599	1.310	2.659	2.572	5.232
15 nodes	1.306	2.635	2.466	4.975	1.465	3.013	2.874	5.925
20 nodes	1.405	2.819	2.570	5.160	1.537	3.106	2.994	6.073
25 nodes	1.442	2.899	2.814	5.641	1.577	3.205	3.110	6.317
30 nodes	1.446	2.907	2.722	5.458	1.591	3.222	3.147	6.357
35 nodes	1.535	3.074	2.778	5.558	1.688	3.426	3.323	6.740

[1] N: No Network Coding.

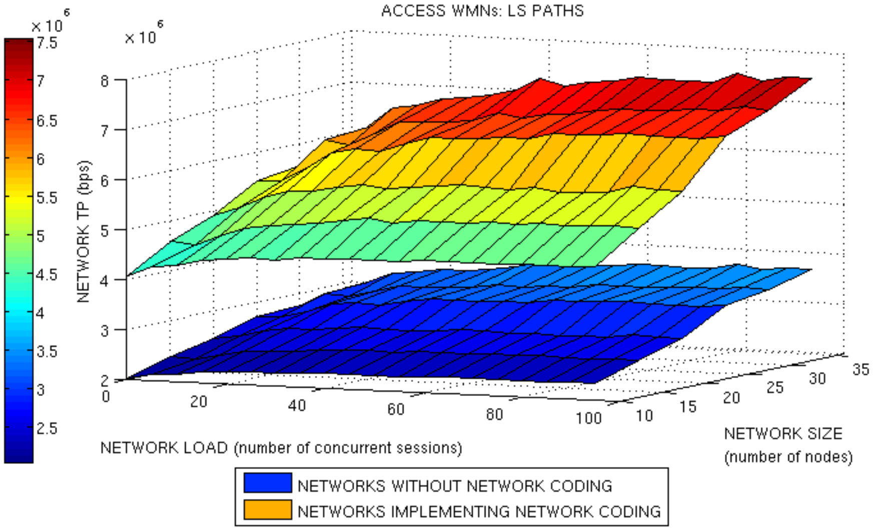
[2] C: Network Coding implemented.

4.2.1.3. Access WMNs

The aggregated throughput values of access WMNs investigated are presented by the graphs in Figures 4.3(a) and 4.3(b). The aggregated throughput values for networks implementing Network Coding are higher than the aggregated throughput values for networks without a Network Coding implementation. Additionally, the aggregated throughput values for networks in which a link-state path determination method was utilised are higher than for networks in which a distance vector path determination method was used.



(a) Access WMNs with distance vector paths



(b) Access WMNs with link-state paths

Figure 4.3.: Aggregated throughput for access WMNs

Table 4.3 presents a summary of critical aggregated throughput values from the graphs in Figures 4.3(a) and 4.3(b).

4.2.1.4. Overall comment on Test 1

The literature suggested that the implementation of Network Coding in WMNs will improve the utilisation of the shared wireless medium in these networks, and as a result cause an increase in the aggregated throughput. The data presented by Test 1 show that the results from the model implementation are consistent with this theory. Additionally, the data presented by Test 1 revealed that the magnitude of the throughput improvement, is dependent on the underlying type of WMN, as well as the method utilised to determine the paths of the network load.

Table 4.3.: Data ranges for different network sizes - access WMNs

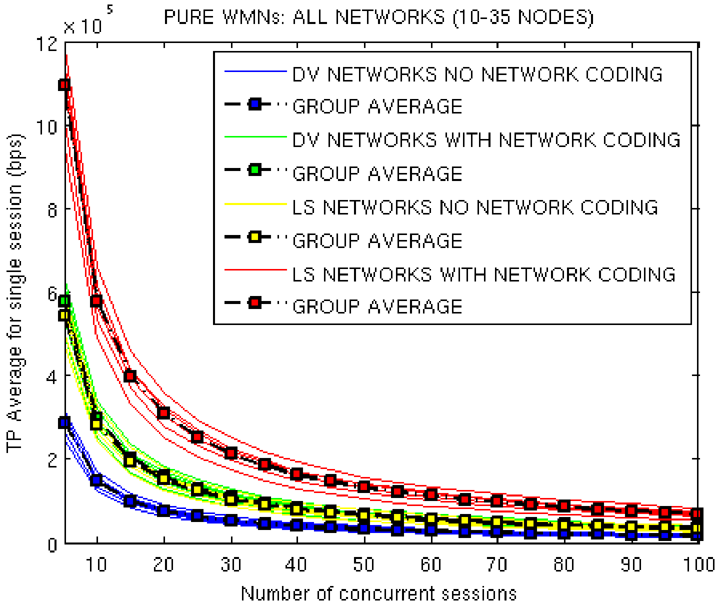
	Throughput level (Mbps)							
	@ 5 sessions				@ 100 sessions			
	Distance vector		Link-state		Distance vector		Link-state	
Size:	N:[1]	C:[2]	N:	C:	N:	C:	N:	C:
10 nodes	1.030	2.058	2.043	4.085	1.193	2.358	2.353	4.678
15 nodes	1.141	2.283	2.290	4.583	1.297	2.593	2.622	5.266
20 nodes	1.244	2.488	2.457	4.914	1.463	2.942	2.886	5.820
25 nodes	1.352	2.712	2.713	5.434	1.638	3.300	3.344	6.744
30 nodes	1.394	2.799	2.764	5.547	1.706	3.466	3.476	7.058
35 nodes	1.511	3.035	2.901	5.818	1.819	3.683	3.706	7.524

[1] N: No Network Coding.

[2] C: Network Coding implemented.

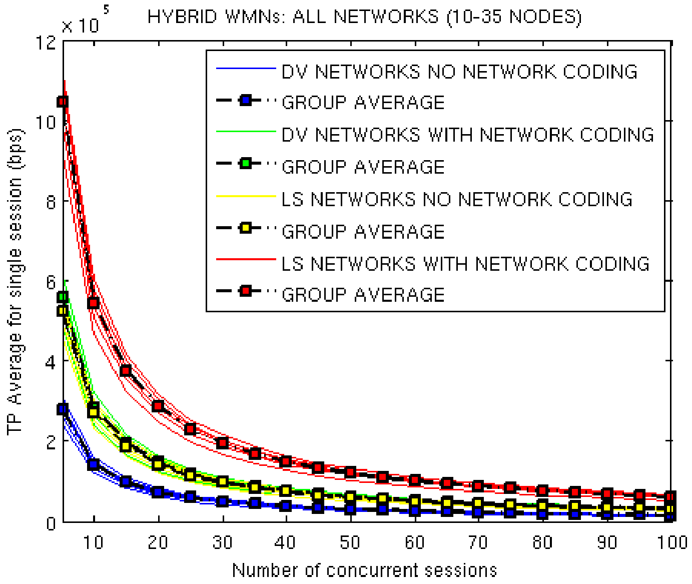
4.2.2. Results of test 2

The second part of the analysis of the result, regards the performance of the average throughput values of single sessions in networks implementing Network Coding, compared to the average throughput values of single sessions in networks that do not implement Network Coding. The average value of the throughput of a single session is calculated at each network load, for each network size. Figure 4.4 displays these values for (a) pure, (b) hybrid and (c) access WMNs.

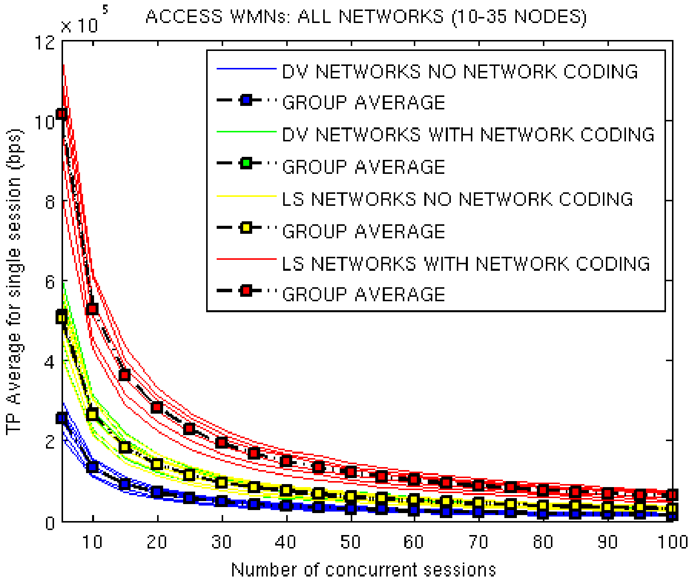


(a) Pure WMNs

Figure 4.4.: Average session throughput for WMNs



(b) Hybrid WMNs



(c) Access WMNs

Figure 4.4.: *Continued*: Average session throughput for WMNs

From the data displayed by Figure 4.4, it can be seen that:

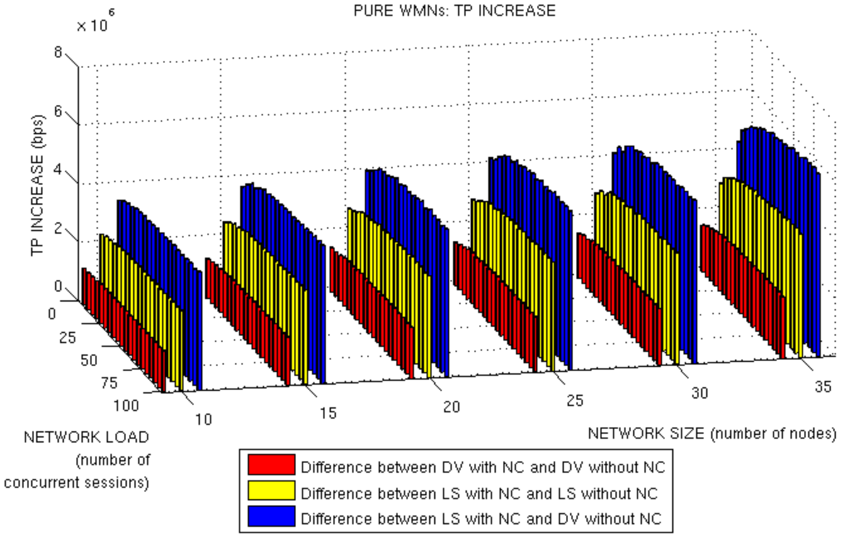
- The single session throughput rates for all the network variations and WMN types, show a steep drop in average throughput values, in the lower end of the range of concurrent sessions.
- For all the WMN types, the single session throughput rates are higher for networks implementing Network Coding, than for networks without Network Coding.
- The networks that utilised link-state path determination methods outperform networks with a distance vector method.
- A session from a network implementing Network Coding with paths determined by a distance vector method, have similar average throughput rates when compared to a session from a network without Network Coding with paths determined by a link-state method.

4.2.3. Additional relevant observations

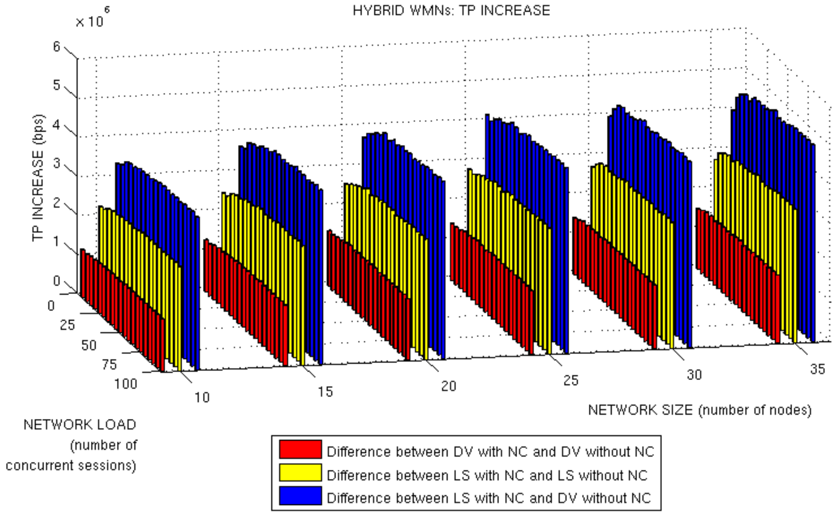
Apart from the observations discussed in the previous two sections, some other relevant observations could be made:

- Each network type show a unique characteristic in terms of aggregated throughput, as well as the magnitude of throughput increase achieved by implementing Network Coding.
- The magnitude of the throughput increase is highest between networks without Network Coding that used a distance vector path determination method, and networks that implement Network Coding and determine paths via a link-state method.

These observations are presented in Figure 4.5.

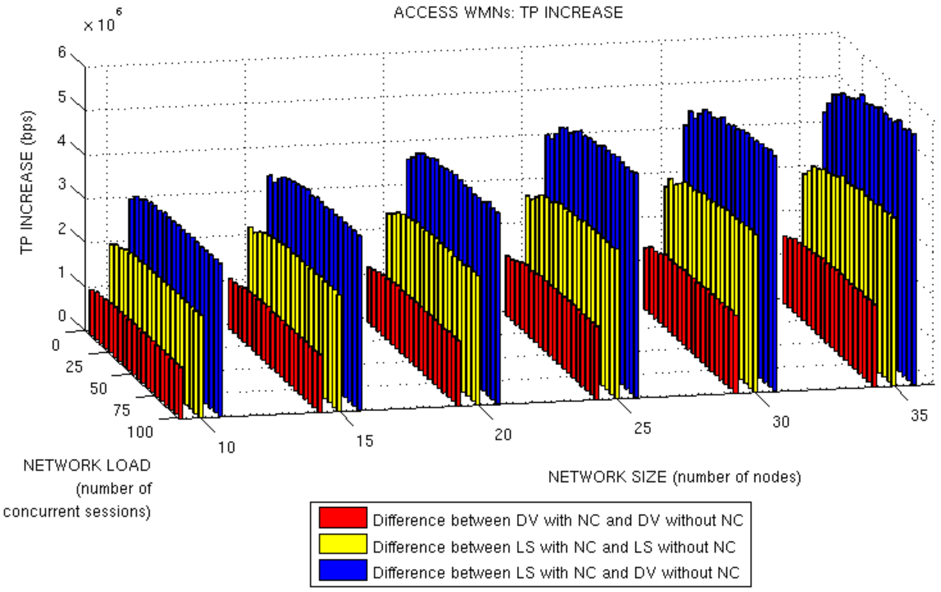


(a) Pure WMNs



(b) Hybrid WMNs

Figure 4.5.: Magnitude of aggregated throughput increase in WMNs



(c) Access WMNs

Figure 4.5.: *Continued*: Magnitude of aggregated throughput increase in WMNs

4.3. Interpretation of results and observations

In this section, the observations made in the in the previous three sections are interpreted. The discussion will have two parts: Firstly the aggregated throughput performance will be discussed; and secondly, the average throughput values for single sessions will be interpreted.

4.3.1. Aggregated throughput performance

In terms of aggregated throughput performance, three observations were made.

4.3.1.1. Observation 1: Networks implementing Network Coding outperformed networks that do not implement Network Coding

Networks that implemented Network Coding yielded higher aggregated throughput values than networks without Network Coding. These Network Coding networks also seem to have sustained a steeper slope of throughput increase. This observation is consistent with the background theory for Network Coding. Network Coding is a means of sharing network resources, such as bandwidth, to decrease the amount of transactions needed to convey multiple data flows over intermediate nodes, thereby increasing the data transfer rate.

The *multiple hops* property, together with the broadcasting nature of each node in WMNs, provide suitable conditions for ample Network Coding opportunities to exists in these type of networks. The advantage of Network Coding opportunities, is that the saturation threshold of the links increase, even though the capacity of the link remains unchanged. This increase in the saturation threshold, is due to the fact that two sessions, at double their permitted rate, share the capacity of the link simultaneously. These sessions are combined through Network Coding, and therefore do not interfere with each other over the particular link.

4.3.1.2. Observation 2: Networks utilising link-state paths outperformed networks that used distance vector paths

Networks utilising link-state paths hold a remarkable advantage over networks utilising distance vector paths. By changing the path selection method in a network from a distance vector (shortest path) method, to a link-state (fastest path) method, produce a similar increase in the aggregated throughput of the network to that Network Coding will offer.

Distance vector methods select paths on a hop count metric, resulting in the shortest path between the source and sink node of a session. As the load through the network increases, the nodes at the center of the network becomes congested, since nodes throughout the network initiate flow. This phenomenon is caused by the fact that the nodes toward the center of the network have a high probability of being intermediate nodes on the paths for most of the concurrent sessions. These are the “popular routes” through the network that always produce the shortest path. It also implies that a number of bottlenecks exist on these paths that decrease the aggregated throughput of the whole network.

Link-state methods select paths on a link-budget scheme that produce the fastest (not always shortest) path through the network. The method tries to avoid bottlenecks, by routing around “busy” links in the network. The result is a wider spread network traffic distribution, that offers relief to highly congested “popular” links. This decentralised flow of traffic also increases the saturation threshold of the links in the network, since the number of flows sharing each link are less than for distance vector paths networks. The maximum magnitude of throughput increase is achieved by implementing Network Coding in a network with a link-state routing method.

4.3.1.3. Observation 3: Each WMN type behaves uniquely in terms of throughput levels

In order to explain the unique behaviour of each of the WMNs, the three types will now be discussed separately.

PURE WMNs

Between the three types of WMNs investigated, pure WMNs are in the best position to perform well with the implementation of throughput improvement strategies. The properties of this type of network, entail that any node in the network have an equal chance to be a source or a sink for any session. The result is an even distribution of sources and sinks throughout the network.

The first advantage of evenly distributed source and sink nodes, is that the network traffic resulting from the network load (number of concurrent sessions), is also evenly dispersed over the whole network. This, in turn, affects the occurrence of bottlenecks in the network in a suppressing way.

The second advantage of evenly distributed source and sink nodes, is that the direction of the network traffic is free-form, meaning there is no definite structure in the traffic pattern and direction. The presence of many sessions through intermediate nodes from opposite directions, results in a higher number of possible Network Coding opportunities.

Pure WMNs offer the highest number of possible Network Coding opportunities, between all three types of WMNs. Implementing Network Coding in conjunction with a link-state forwarding method, improves the aggregated throughput performance up to fivefold.

HYBRID WMNs

The hybrid WMNs in the experiment are set-up with a routing backbone, which consists of one fifth of the nodes in the network with a direct connection between them. These nodes can never be selected as sources or sinks, and will always be intermediate nodes. The reason for the routing backbone in WMNs is to expand the reach of the network by connecting smaller groups of nodes into one large WMN.

The first implication of adding a routing backbone is that a default static route exists through the network. In the case of distance vector paths network variations, this backbone route becomes the single most "popular" route through the network, with the highest demand from the network load. Network traffic is directed from the edges, through the backbone, causing the links in the regions surrounding the backbone to reach a congested level very quickly (number of concurrent sessions in the lower quartile of the load range).

The second implication of adding a routing backbone is the bias of the network traffic direction towards and from the routing backbone. This phenomenon has a restricting effect on the number of Network Coding opportunities that exists in the network. Due to their position in the network traffic pattern, the intermediate nodes of the routing backbone have the highest probability for Network Coding opportunities. Some measure of relief is offered to the highly congested links in these regions. However, the number of concurrent sessions that need to share links are also much higher in these regions. The magnitude of the throughput increase offered by Network Coding is limited by the by the higher number of sessions requesting to pass through the links in these regions at once. This, in turn, causes a backlog in the scheduling process. Link-state paths network variations have the advantage of a more evenly distributed network traffic pattern. The number of possible Network Coding opportunities is therefore higher.

ACCESS WMNs

Access WMNs are the worst performing networks of all three types of WMNs investigated. The APs backbone present in the network, consists of one fifth the nodes in the network which are designated to be the ultimate sinks for all the concurrent sessions. Beyond the APs, the network traffic is external to the network and not considered any more.

Network traffic are mainly directed towards the AP backbone. This bias in the traffic direction limits the occurrence of Network Coding opportunities in the network. Link-state paths network variations will aim to spread the traffic in the regions towards the AP backbone. However, the AP backbone is still ultimately controlling the direction of the traffic. The regions around the AP backbone become highly congested, and limit the performance of the aggregated throughput for access WMNs.

4.3.2. Single session throughput performance

In terms of single session throughput performance, two observations were made.

4.3.2.1. Observation 1: Throughput rates decrease as the load increase

A steep drop in throughput values occurred for each examined network instance, with the most severe drop within a load of 15 concurrent sessions present in the network. This phenomenon is accounted for by the dense collection of RF activity in all three types of WMNs, and in support of this statement the following explanations are given:

- WMNs are *multiple hop* wireless networks. In conventional wireless networks, messages are transmitted over a distance of one hop, between the user nodes and the APs. After a message reaches the AP, it is no longer present in the network. Messages in WMNs, on the other hand, are routed through user nodes and backbone routers. This entails that the messages are continually forwarded, and can exist in the network for multiple hops without being terminated or discarded. As a result, the number of RF messages over the links increase for WMNs, without an increase in the number of initiated message flows.
- Each node in the WMN is connected to the network, and must therefore participate in the network activity. Conventionally, a node will only be "awake" while it is transmitting a message to the AP, or while it is receiving a message from the AP. The rest of the time the node is in *idle* or *sleeping* mode. All unintentional messages intercepted by a node are ignored and discarded. This is not possible in a WMN. When a node intercepts a message, and is not the intended sink of that message, it may still be an intermediate node en route to the sink, and must therefore forward the message. This entails that every node must process every message it receives in order to rightfully dismiss, or forward a message. This *message processing* addition to the tasks of the node, cause a delay in its own schedule, since it must first process a received message before it can send or receive its own intended messages. In practice a higher packet drop rate would be expected. The overall result is a decrease in the throughput rates of sessions.
- The final remark regarding the RF effects on the single session throughput rates, is the fact that the link capacities are effected by the concentration of links in a confined space. In open space, or at the instance where only 5 concurrent sessions are present, the links are not affected by other "busy" links in the same RF range. Each link can utilise the full capacity of the wireless link. Once more links are assigned with a load in the same area, the links must share the bandwidth and cannot utilise the full potential of the link. This effect is evident once the number of sessions increase. However, at some point the network converge to a saturated level, where the links cannot carry any more flow, regardless of an increase in the number of concurrent sessions.

4.3.2.2. Observation 2: The average throughput of a single session is relatively low

The final observation on the single session throughput values, is that the calculated average values are much lower when compared to conventional wireless networks, in particular IEEE 802.11n standard networks. The question pertaining to this observation, is whether the calculated values in this experiment is realistic, or not. In order to provide an answer to this question, the calculated average throughput values have to be evaluated in a fair comparison. The comparison of these values to conventional wireless networks, is not on an equal standard. Therefore, the calculated values are compared with measured throughput values from IEEE 802.11s networks, which is the IEEE 802.11 Std. amendment for *mesh* networking in WLANs.

J. D. Camp and E. W. Knightly present calculated throughput rates for IEEE 802.11s networks in the range of 100-450 Kbps, [47]. Y. Lin, et. al. present measured throughput rates for IEEE 802.11s networks in the range of 300-1000 Kbps, [48]. The comparison between the calculated average throughput values for single sessions from the model, and calculations and measurements from IEEE 802.11s networks, reveals that the model is realistic.

In this chapter the results obtained from the developed model implementation in the previous chapter were presented. Section 4.1 described the data sets utilised during the analysis of the results, as well as a description of the different tests involved in the analysis.

Each network type, pure, hybrid and access WMNs, were evaluated separately to examine the performance of different network variations. In every evaluation, the networks that implemented Network Coding outperformed the networks that did not implement it, in terms of calculated aggregated throughput.

The findings also showed that the networks that implemented a link-state path determination method outperformed the networks that implemented a distance vector path determination method.

5. Validation and verification

The aim of this chapter is to validate the developed model and verify that the the results obtained with the model are reliable. Some background concepts and definitions are presented in section 5.1, followed by the benchmark conditions for the validation and verification in section 5.2. The validation and verification process is described in section 5.3. Additional information regarding the efficiency of the solving method utilised by the model, is discussed in section 5.4.

5.1. Background concepts and definitions

As background to the validation and verification process followed in this dissertation, some concepts and definitions are first provided. These concepts and definitions are:

- **Model validation:** This is the process that demonstrates whether the developed model describes real-world WMN systems with an acceptable range of accuracy.
- **Model verification:** This is the process that demonstrates whether the model is correctly implemented to produce trustworthy results.
- **Model data:** The data results produced by the *model* needs to be scrutinised in order to establish its credibility, and is referred to as the *model data*.
- **Control:** In order to have counter measurements to compare the model data to, a different, independently developed model is implemented to produce valid data as a comparative standard. This reference model, is the *control*.
- **Control data:** The data samples produced by the control.

Mathematical models are not *global* representations of real-world systems, but representations of an isolated set of experimental conditions. In the case of this study, the model was specifically developed to compute the optimal aggregated throughput of WMNs. Therefore, the mathematical representation utilised in the control follows an throughput optimisation approach in the form of a network flow problem. The solving method selected for the control, is CPLEX[1]. A description of the control development is given in appendix C.2.

Normally the best way to validate a model, would be to perform an experimental validation, where the model results are compared to results obtained from a practical implementation of the experiment [49].

---

[1] See appendix C for an in depth discussion on the ILOG CPLEX Optimizer and the development of the control model

For the WMNs investigated in this study, a real-world implementation was not possible. The implication being that a pragmatic approach had to be followed.

Examples of typical pragmatic approaches are [50]:

- Find similarities among corresponding events of the real system and the events of the model.
- Comparing the model to other models that have already been validated.
- If known results of analytical models are available, it can be used to compare against outputs of the model under consideration.

5.2. Benchmark conditions

A prerequisite for the validation and verification procedures, is to to establish the benchmark conditions and assumptions for the model and control. The reason for these conditions is to assure that the comparisons between the model and control are justified. For this study, the benchmark conditions were:

- Both the model and control problems must have the property of being integer problems. This is to ensure the flows is kept on *single paths*.
- The test must only include basic elements, and all elements should participate in the network activity. This entails:
  - Nodes and links must be part of the test network, no other third-party interfering nodes can be present.
  - All interfering effects, apart from two or more separate flows being allowed to share a common link, are excluded in the test.
- The goal of the model and control is to determine the maximum aggregated throughput of the test network.

In order to restrict the collection of data samples to a common region for both the model and control, the following parameter selections were utilised:

- Network size: 20 Nodes. This selection is based on the assumption that 20 nodes offer a network large enough to contain many different paths for the sessions, yet the dimension of the problem size is small enough for the CPLEX optimiser to solve the control relatively "quickly" (within a acceptable number of iterations).
- Data samples are collected for 4 stages of varying load: 20, 40, 60 and 80 concurrent sessions. This is to test the model's ability to keep throughput calculations consistent for variance between a small number of variables to a large dimension of variables.
- Each stage computes a sample of 100 data values. Identical session sources and sinks are used as feed for both the model and control to keep the evaluation conditions equal.
- The flow commodity is generic, therefore measured in *units of flow*, rather than *bps*.
- A link capacity of 1000 units of flow is selected. This setting also accelerate the flow augmentation of the solving methods, by limiting the search space of the algorithms between 0 and 1000.
- Every link operates at 100%, and there is no weighted scheme implemented in the network

graph.

- An accuracy acceptance threshold is defined as $\phi = 0.5$. The model will be accepted as *accurate* if its measured accuracy level is $1 \pm \phi$ when compared to the control.

5.3. Validation and verification procedures

The procedures involved in the validation and verification of the model and results were conducted in the form of three independent studies [50]: 1. The conceptual validation and verification study; 2. The statistical validation study; and 3. The operational verification study.

5.3.1. Conceptual validation and verification

The aim of the conceptual validation and verification study is to defend the logic involved in the model development, as well as the assumptions and test conditions that were selected for the model implementation. The study comprises of the following procedures:

- *Similarities test:* Find similarities among corresponding events and properties of a real-world system and the model assumptions. This process **validates** the logic of the model assumptions and setup conditions.
- *Format comparison:* Compare the format of the model to the format of the control. This process **verifies** if the logic followed by the model design is acceptable.

The goal of the similarities test is to find similarities between the WMN literature descriptions, presented in chapter 2 of this dissertation, and the properties of the developed model. Table 5.1 summarises the findings of the similarities test, by giving the characteristic descriptions from the background literature juxtaposed with the similar property assigned to the model.

Table 5.1.: Similarities between real world WMNs characteristics and the model assumptions and conditions

Characteristic description:	
Background literature	Model property
NETWORK ELEMENTS: topology, nodes & network types:	
Collection of wireless nodes.	Grouping of nodes connected to other nodes based on a wireless channel model.
In a mesh topology.	Placed to form a mesh.
Formed in an ad hoc manner (no fixed formation schedule).	Node coordinates and position in mesh selected randomly.
Mesh nodes can be one or a combination of these types: client, router or AP.	Nodes are generic, but can be part of three types of node sets: users, router backbone, or AP backbone.
There are three types of WMNs: pure, hybrid and access WMNs.	Three variations are created: pure, hybrid and access WMNs.

Table 5.1.: *Continued:* Similarities between real world WMNs characteristics and the model assumptions and conditions

Characteristic description: Background literature	Model property
NETWORK OPERATIONS: routing & the network coverage:	
Multiple hop networks.	Session paths are able to span over several links.
All nodes can perform routing, including clients.	Flow can be relayed through all the nodes in the model's network.
Network coverage for all participating nodes.	All nodes are connected to the network and reachable for all other nodes.
NETWORK RESOURCES: channels, links & the RF environment:	
Faced by all the physical constraints in a RF environment.	Links subjected to a mathematical path-loss and shadowing model.
Bandwidth shared by all nodes, leading to high degree of transmission scheduling.	Links shared by multiple sessions, divides the capacity.
Rate of transmissions constraint by physical capabilities of the nodes and links.	Flow rates constraint by upper boundary (channel capacity value).

The *fundamentals* of real-world WMNs, as it is documented in available literature, were captured by the model. Therefore, it is reasonable to accept the model as an accurate description of the real-world system.

The design of the developed model is logically verified through a format comparison between the mathematical representation of the model, and the mathematical representation of the control. The investigated aspects were: 1. The variables of the optimisation problems; 2. The boundaries laid down by the constraints; and 3. The objectives of the model and control respectively. Table 5.2 summarises the findings of these aspects, by stating each aspect represented in the control juxtaposed with the representation in the model.

Table 5.2.: Format comparison between the control and model design

Investigated aspect:	Control representation:	Model representation:
FUNCTION VARIABLES		
Link flows:	$x_a \, \forall \, a \, \in \, A$ where A is the set of arcs.	$f_{e_{ji}} \, \forall \, e \, \in \, E$ where E is the set of links and $j, \, i \; e \, \in \, V$, the set of nodes.
Session flows: (commodity flows)	$\lambda_k, \, \forall \, k \, \in \, \xi$ Where ξ is the set of sessions containing the paired source and sink node of each session. The control and model utilised the same ξ.	$\lambda_i, \, \forall \, i \, \in \, \xi$
Single path restriction variables:	$y_a \, \forall \, a \, \in \, A$ $y_a \, \in \{0, 1\}$	Not applicable.

Table 5.2.: *Continued:* Format comparison between the control and model design

Investigated aspect:	Control representation:	Model representation:
BOUNDARIES OF THE VARIABLES		
Mass-balance constraint:	$\sum_{\substack{a\,\in\,T\\k\,\in\,\xi}} x_{ak} - \sum_{\substack{a\,\in\,H\\k\,\in\,\xi}} x_{ak}$ $= \begin{cases} -\lambda_k \text{ if } n = S_k \\ \lambda_k \text{ if } n = T_k \\ 0 \text{ otherwise} \end{cases}$ ∀ nodes $n \in N$; and arcs $a \in A$; and commodities $k \in \xi$	$\sum_{\forall j \text{ in}} f_{e_{ji}}(s) + \sum_{\forall k \text{ out}} f_{e_{ik}}(s)$ $= \begin{cases} -\lambda_s \text{ if } i = S_s \\ \lambda_s \text{ if } i = T_s \\ 0 \text{ otherwise} \end{cases}$ ∀ nodes $i, j, k \in V$; and sessions $s \in \xi$
Capacity constraint:	$l_a \le x_{ak} \le u_a$ $l_a = 0$ $u_a = C$ ∀ arcs $a \in A$; and commodities $k \in \xi$;	$f_i(s) = \sum_{s \in \xi} \left(\sum_{\forall j \text{ in}} f_{e_{ji}}(s) \right)$ $0 < f_i(s) \le C$ $f_i(s) \in \Re$ ∀ nodes $i, j, k \in V$; and sessions $s \in \xi$
Path constraint:	$\sum_{\substack{a\,\in\,T\\k\,\in\,\xi}} y_{ak} \le 1$ $\sum_{\substack{a\,\in\,H\\k\,\in\,\xi}} y_{ak} \le 1$ $x_{ak} \le \lambda_k y_{ak}$ where $y_{ak} \in \{0, 1\}$	Not applicable. The model utilises the coefficients represented in the path-links matrix
OBJECTIVE		
Goal function:	$TP_{\text{control}} = \max \sum_{k=1\ldots s} \lambda_k$ ∀ commodities $k \in \xi$	$TP_{\text{model}} = \max \sum_{i=1\ldots s} \lambda_i$ ∀ sessions $s \in \xi$

In terms of network flow representation *format*, the control and model are similar with regards to function variables, constraints and the objective. However, the model did not include the single path restriction variables vector nor the path constraint function. Instead, the model utilised a path-links data structure yielded by the network setup procedures of the model implementation. The path-links matrix intrinsically contains the session flow coefficients for single paths.

5.3.2. Statistical validation

From the central limit theorem [46], a normal distribution for a sample is *assumed* if the sample size is greater than 30. The number of calculations in each of the model data and control data samples are 100. Since 100≫30, it is assumed that the data samples are normally distributed. Additionally, the mean values utilised in the analyses, presented in this section, are assumed to be acceptable approximations of the unknown population mean values. The following notations are used:

- Data sample mean and median: μ and $\tilde{\mu}$ are used and not $\bar{x}$ and $\tilde{x}$.
- Data sample standard error (deviation): σ are used and not s.

A collection of calculated statistics for the model data and control data samples of each stage are presented by Table 5.3.

Table 5.3.: Control data and model data statistics

	Mean, $\bar{\mu}$	Median, $\tilde{\mu}$	Variance	Std. deviation σ	Minimum	Maximum	Interquartile range	Skewness	Kurtosis
Stage 1: 20 concurrent sessions:									
Control:	14180	14000	2.7×10^6	1643.3	11000	20515	1966.5	0.756	4.651
Model:	13852	13808	2.2×10^6	1485.5	9408.1	19700	1967.2	0.278	4.649
Stage 2: 40 concurrent sessions:									
Control:	21525	21410	4.4×10^6	2103.3	16680	26293	2954.5	-0.0015	2.455
Model:	20921	20971	7×10^5	835.42	18612	22424	1106.2	-0.498	2.845
Stage 3: 60 concurrent sessions:									
Control:	25022	25000	5.4×10^6	2326.3	18916	31000	2844.5	0.089	3.056
Model:	24020	23919	7.6×10^5	873.37	21716	25798	1199.5	-0.021	2.761
Stage 4: 80 concurrent sessions:									
Control:	29845	29836	7.2×10^6	2679.3	24000	35329	3837.5	0.022	2.443
Model:	29134	29136	1.6×10^6	1250.7	24821	31797	1694.8	-0.523	3.7151

The statistical validation procedures established whether the model and control behave alike when subjected to the same statistical analyses [46], [50]. Three analyses were performed to **validate** the behavioural properties and characteristics of the model. These analyses are:

- *Test for normal distributions:* This test examines whether the model data and the control data samples are both from the normal distribution family. The aim is to indicate whether the data yielded by the control and model share similar properties in terms data distribution.
- *Test for equal distributions:* This test investigates similarity between the distributions of the model data and control data samples. The outcome of the test for equal distributions is a confirmation that the control and model produce equal results.
- *Test for equal means:* This test evaluates the similarity or difference in the means of the respective model data and control data samples.

Two complimentary techniques were utilised by the statistical validation procedures, namely graphical assessments and hypotheses tests. The graphical assessment of the data samples entails different visual presentations of the data as it pertains to the analysis currently conducted. These graphical presentations compliments the examination of different hypotheses assumed for the behaviour of parameters of the data samples.

Statistical hypotheses are the conjectures about a parameter or parameters of the model and control data samples [46], [50]. The purpose of a hypothesis test is to prove or dismiss these hypotheses by means of contradiction against a null hypothesis. The following terminology are associated with the process of a hypothesis test:

- Null hypothesis, H_0: This is the assumption to be tested. The "null" status of this hypothesis is due to the fact that their is an absence of a characteristic or the lack of an effect, such as no increase or decrease in the mean.
- Alternative hypothesis, H_A: This is a contrasting assertion about the sample that can be tested against H_0.
- Significance level: This is the probability threshold α, typically set to 0.05. The hypothesis test returns a *p*-value, which indicates the probability of the test under the assumption of H_0, to obtain a value of the test statistic as extreme or more extreme than the value computed from the sample. Small *p*-values cast doubt on the validity of H_0. The following actions are possible:
  - If $p < \alpha$: Reject H_0.
  - If $p > \alpha$: Insufficient evidence to reject H_0.
- Type I error: Reject H_0 when it is actually true.
- Type II error: Accepting H_0 when it is actually false.
- Power of test, $1 - \beta$: The distribution of the test statistic under H_0 determines the probability α of a type I error. The distribution of the test statistic under H_A determines the probability β of a type II error. $1 - \beta$ is the probability of correctly rejecting a false H_0.

Additional information and setup parameters necessary in preparation for the hypotheses tests are:

- Standardisation: For normally distributed populations, a standardised variable, z, exists from the standard normal distribution *z*-curve ($\sigma = 1$ and $\mu = 0$) [46]. For a sample size greater than 30, a normal distribution for the sample is usually assumed from the central limit theorem. The model and control data samples both have 100 calculated values for each stage in the evaluation, leaning towards a *z*-curve standardisation utilised for the analyses. However, after considering the skewness and kurtosis values from Table 5.3, the model and control data were standardised to a *t*-curve normal distribution, with standardised variable, t and 99 degrees of freedom.
- In all the hypotheses tests, a significance level of 5% is set. All the tests are set for a two-sided testing, and a critical *t*-value of are interpolated from the standard table for critical values for the Students's *t*-curve. The values are:
  - $t_c = 1.9867$ at a 95% confidence interval.
  - $t_c = 1.6623$ at a 90% confidence interval. This value is very close to $z_c = 1.67$ at a 90% confidence level for the standard normal distribution.

5.3.2.1. Test for normal distributions

The Lilliefors test was utilised to test for normality in the distributions of the different model data and control data samples. This test is a two-sided goodness-of-fit test, suitable when a fully-specified null distribution is unknown and its μ and σ parameters must be estimated. The null hypothesis and alternative hypothesis are:

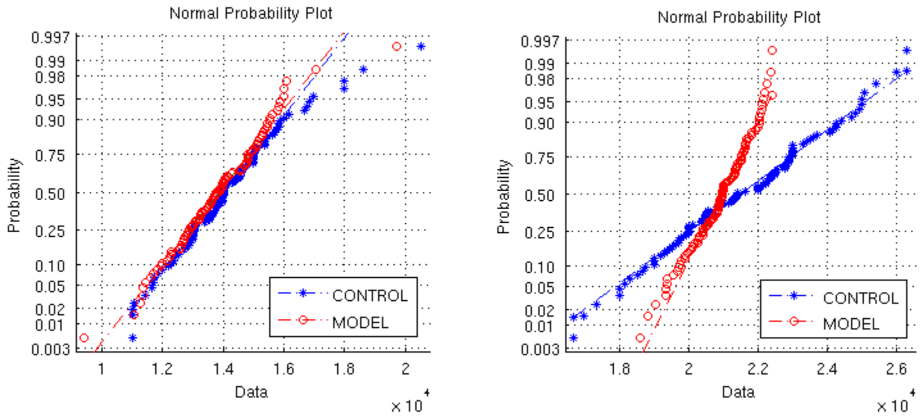
- H_0: The tested sample have a normal density.
- H_A: The tested sample does not have a normal density.

Table 5.4 summarises the results of the Lilliefors test conducted for each different model data and control data sample:

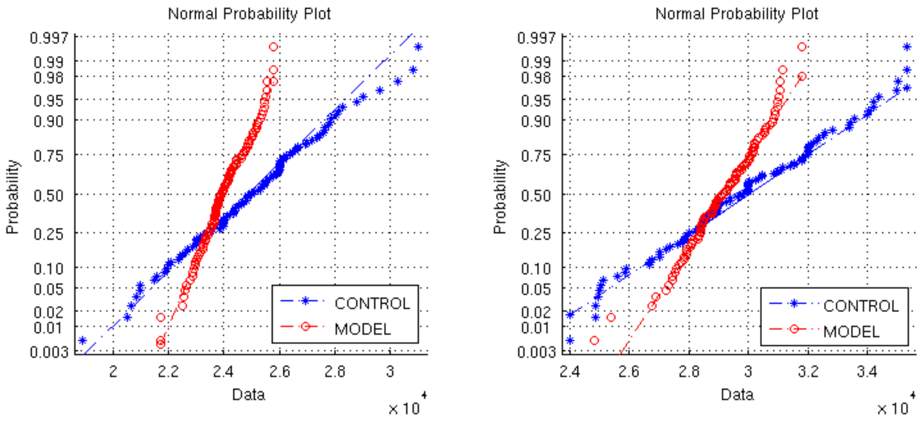
Table 5.4.: Results of the Lilliefors test for normality at a 5% significance level

	Stage 1: 20 sessions	Stage 2: 40 sessions	Stage 3: 60 sessions	Stage 4: 80 sessions
Control data:	Accept H_0	Accept H_0	Accept H_0	Accept H_0
Model data:	Accept H_0	Accept H_0	Accept H_0	Accept H_0

The results of the Lilliefors test show that all the model and control data samples are normally distributed. These results can also be graphically shown by normal probability plots. A normal probability plot graphically assess whether data samples are normally distributed, indicated through the linearity yielded by the plotted data. The normal probability plots of the four different load stages of the model data and control data samples are presented in Figure 5.1.



(a) Stage 1, 20 sessions (b) Stage 2, 40 sessions



(c) Stage 3, 60 sessions (d) Stage 4, 80 sessions

Figure 5.1.: Normal probability plots: control data and model data samples

5.3.2.2. Test for equal distributions

Table 5.3 revealed that the model data and control data samples have different variance values at each investigated load stage. Therefore, the Ansari-Bradley test is selected to prove similarity in the distributions of the data samples, because it does not use variance as a decision parameter.

Three possible alternative hypotheses for the Ansari-Bradley test are provided:

- H_0: The two independent samples, control data and model data, come from the same distribution: $\sigma_{\text{control}} = \sigma_{\text{model}}$.
- H_{A1}: The dispersion parameters of the control data and model data are not equal, **two-tailed** test: $\sigma_{\text{control}} \neq \sigma_{\text{model}}$.
- H_{A2}: The dispersion parameters of the the control data are less than dispersion of model data, **left-tailed** test: $\sigma_{\text{control}} < \sigma_{\text{model}}$.
- H_{A3}: The dispersion parameters of the the control data are greater than dispersion of model data, **right-tailed** test: $\sigma_{\text{control}} \geq \sigma_{\text{model}}$.

The output of the Ansari-Bradley test are shown in Table 5.5.

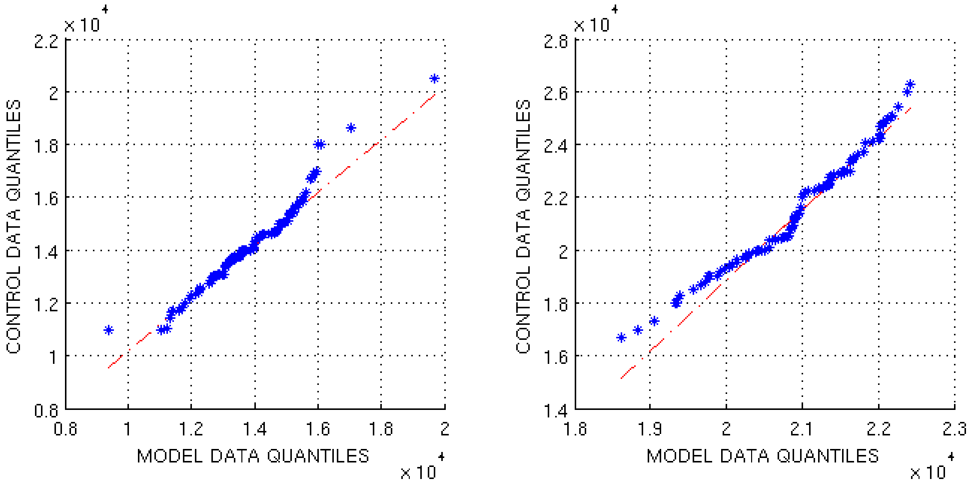
Table 5.5.: Results of the Ansari-Bradley test for equal distributions at a 5% significance level

	Stage 1: 20 sessions	Stage 2: 40 sessions	Stage 3: 60 sessions	Stage 4: 80 sessions
Two-tailed test:	Accept H_0	Insufficient	Insufficient	Insufficient
Left-tailed test:	Accept H_0	Accept H_0	Accept H_0	Accept H_0
Right-tailed test:	Accept H_0	Insufficient	Insufficient	Insufficient

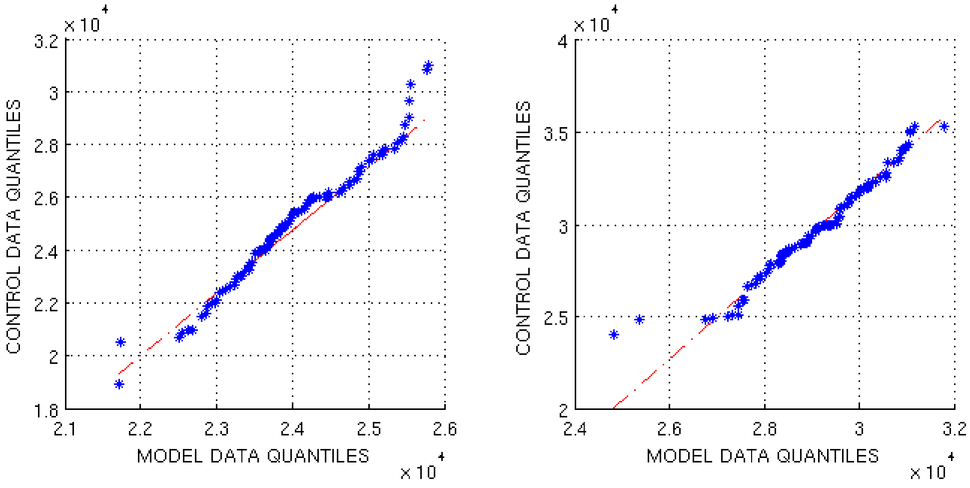
The output of the Ansari-Bradley shows that the model and control produce data from equal distributions at a load of 20 concurrent sessions. For 40, 60 and 80 concurrent sessions the test accepted the null hypothesis for the left-tailed test. The returned p-values for the two-tailed and right-tailed tests were insufficient evidence to accept the null hypothesis at these loads. This output suggests that the distributions from the model and control data samples are equal, however, the values of the control data samples are wider spread than the values from the model data samples.

This observation can also be seen in the values for the standard deviation statistic in Table 5.3. At 20 concurrent sessions the standard deviation values for the model data sample and control data sample are very similar. For 40, 60 and 80 concurrent sessions, the standard deviation values for the model data samples are less than for the control data samples.

Confirmation in terms of equal distributions for the model data and control data samples are provided by the Quantile-Quantile plots shown in Figure 5.2. A linear Quantile-Quantile plot indicates that the data samples plotted against each other are equally distributed.



(a) Stage 1, 20 sessions (b) Stage 2, 40 sessions



(c) Stage 3, 60 sessions (d) Stage 4, 80 sessions

Figure 5.2.: Quantile-Quantile plots: control data and model data samples

5.3.2.3. Test for equal means

The final hypothesis test is concerned with investigating the behaviour of the means of the model data and control data samples. The *t-test* is utilised to test the mean parameter, μ, of the two samples against each other. Three possible alternative hypotheses are provided for the t-test, given as follows:

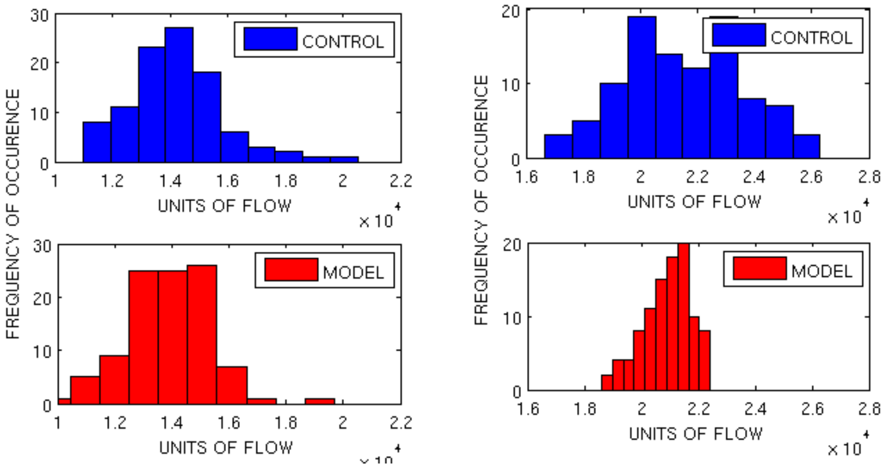
- H_0: The two independent samples, control data and model data, have the same mean: $\mu_{\text{control}} = \mu_{\text{model}}$.
- H_{A1}: The mean of the control data and model data data are not equal, **two-tailed** test: $\mu_{\text{control}} \neq \mu_{\text{model}}$.
- H_{A2}: The mean of the the control data are less than the mean of model data, **left-tailed** test: $\mu_{\text{control}} < \mu_{\text{model}}$.
- H_{A3}: The mean of the the control data are greater than the mean of model data, **right-tailed** test: $\mu_{\text{control}} \geq \mu_{\text{model}}$.

The results of the *t*-test are shown in Table 5.6.

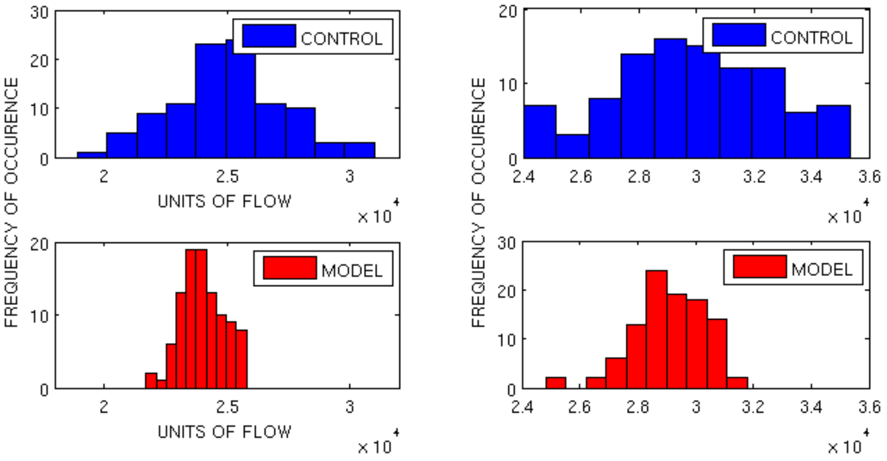
Table 5.6.: Results of the *t*-test at a 5% significance level

	Stage 1: 20 sessions	Stage 2: 40 sessions	Stage 3: 60 sessions	Stage 4: 80 sessions
Two-tailed test:	Accept H_0	Insufficient	Insufficient	Insufficient
Left-tailed test:	Accept H_0	Accept H_0	Accept H_0	Accept H_0
Right-tailed test:	Accept H_0	Insufficient	Insufficient	Insufficient

The output of the two-sided *t*-test reveals that the model and control data samples have equal means at 20 concurrent sessions. For 40, 60 and 80 concurrent sessions the test accepted the null hypothesis for the left-tailed test. The returned *p*-values for the two-tailed and right-tailed tests were insufficient evidence to accept the null hypothesis at these loads. This suggests that the means of the model data samples are shifted towards a lower level when compared to the control data samples, since the alternative hypothesis that the means are greater is rejected. This outcome is confirmed in the calculated mean values presented in Table 5.3, and the histograms displayed in Figure 5.3.



(a) Stage 1, 20 sessions (b) Stage 2, 40 sessions



(c) Stage 3, 60 sessions (d) Stage 4, 80 sessions

Figure 5.3.: Histograms: control data and model data samples

The histograms displayed in Figure 5.3 show the shift in the means for the model data for 40, 60 and 80 concurrent sessions. The less spread dispersion of the model data samples as revealed by the Ansari-Bradley test are also shown.

5.3.3. Operational verification

The final study in the validation and verification process, is the operational verification of the model. This study was aimed to examine the precision and accuracy of the results, in order to verify that the model produce trustworthy results. Two processes were involved in the operational verification study, namely: 1. Investigation of confidence intervals; and 2. The observed accuracy of the model.

5.3.3.1. Confidence intervals

Confidence intervals were utilised to measure the precision of the model data and control data samples. Confidence Intervals (CIs) are *interval estimates* within a sample range that describes the level of uncertainty associated with the sample estimate of a population parameter θ. A CI defined for a certain confidence level, for example 95%, is interpreted as follows: *It is expected that 95% of the interval estimates would include the population parameter θ.*

CIs are created by calculating the upper and lower bounds of the error of the estimate, i.e. the *margin of error*. The following steps describes the calculation of a CI:

1. Calculate α: $\alpha = 1 - \frac{\text{confidence level}}{100}$
2. Calculate the critical probability p^*: $p^* = 1 - \frac{\alpha}{2}$
3. Get the critical value t^*: Use the cumulative probability tables for the standardised *t*-curve, and set $t^* = t-$value where $p^* =$ the cumulative probability at the correct value for the degrees of freedom (df = number of samples (*n*) - 1). Use interpolation if the df values are not available in the table.
4. Margin of error: ME $= t^* \times \frac{\sigma}{\sqrt{n}}$
5. CI $= \mu \pm$ ME

The precision measurement offered by a CI is contained in the range of the interval. The shorter the length of the CI, the greater is the precision of the interval estimate of the population parameter θ. However, CIs do not show the significance of the difference between two sample means. An overlap of CIs is not the correct indicator to determine whether two means are significantly different or not. In stead, the *CI around the the difference* in the concerned means is considered. If the CI for the means difference between the model data and control data samples do not contain zero, the means of the two samples are significantly different. This indication, after algebraic manipulation, is mathematically expressed as follows:

If:

$$(\mu_{\text{control}} - \mu_{\text{model}}) - t^* \sqrt{\sigma^2_{\text{control}} + \sigma^2_{\text{model}}} > 0 \tag{5.1}$$

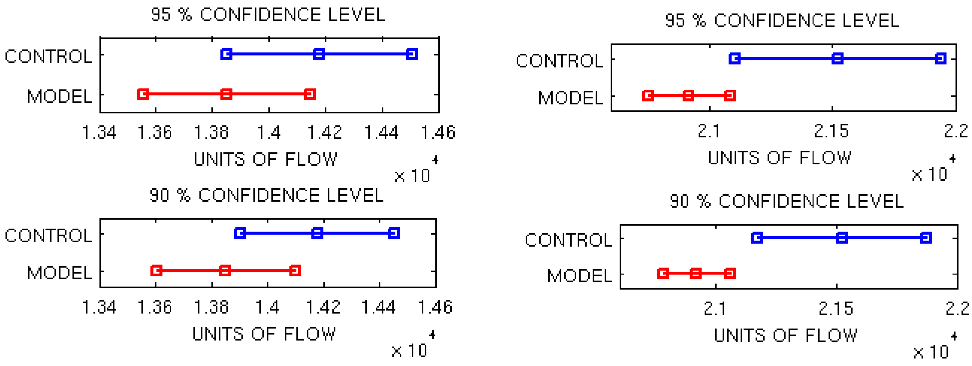
μ_{control} and μ_{model} are significantly different.

The statistics μ and σ from Table 5.3 of the model and control data samples, at the different load stages, were utilised during the examination of CIs in this operational verification study. The aim was to determine how precise these parameters estimated the true population mean of a system that calculates the aggregated throughput of WMNs. A notation for this unknown population mean is selected as θ_μ.

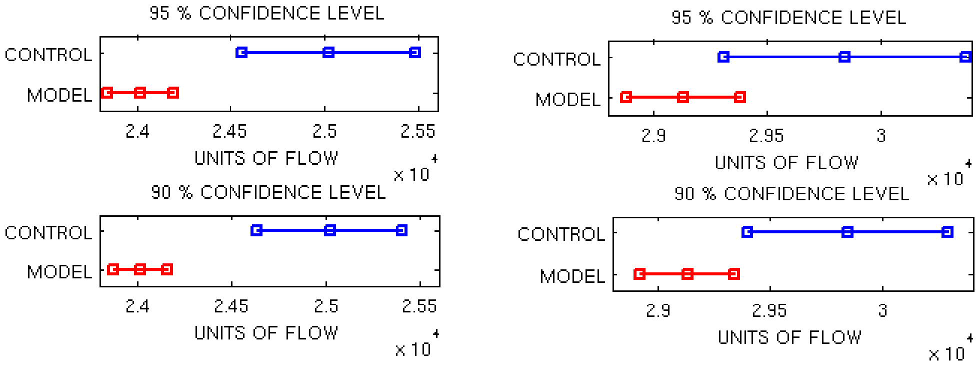
Two CIs were calculated at each stage. These intervals are:

- $\alpha = 0.05$: For a CI with a 95% confidence level.
- $\alpha = 0.1$: For a CI with a 90% confidence level.

The graphical presentation of the investigated CIs are given in Figure 5.4.



(a) Stage 1, 20 sessions (b) Stage 2, 40 sessions



(c) Stage 3, 60 sessions (d) Stage 4, 80 sessions

Figure 5.4.: CIs for model data and control data samples, $\alpha = 0.05$ and $\alpha = 0.1$

The findings from the examined CIs show that the μ and σ statistics of the model data samples yielded greater precision in the estimates of θ_μ at $\alpha = 0.05$ and $\alpha = 0.1$ when compared to the control data samples. This observation is true for the stages with 40, 60 and 80 concurrent sessions. At 20 concurrent sessions the model data and control data samples yielded similar estimates.

The presented CIs indicated the lower levels of mean values for the model data samples, in coherence with the results yielded by the left-tailed *t*-test during the statistical validation study. However, the overlapping and non-overlapping of these CIs cannot be utilised to comment on the significance of this means difference. Therefore, the calculation from Equation 5.1 were utilised to gain an understanding in the significance of the difference between the means of the model data and control data samples. The results of these calculations are summarised in Table 5.7.

Table 5.7.: Results for Equation 5.1

α	Stage 1: 20 sessions	Stage 2: 40 sessions	Stage 3: 60 sessions	Stage 4: 80 sessions
0.05	-4.0734e+03	-3.8922e+03	-3.9338e+03	-3.3920e+03
0.1	-3.3549e+03	-3.1582e+03	-3.1278e+03	-4.2045e+03

The negative output of Equation 5.1 in Table 5.7 shows that the difference between the means of the model data and the control data is *not significant*. Therefore, the data samples yielded by the model are accepted estimates for the population of a system that calculates aggregated throughput of WMNs, with a high measure of precision (i.e. greater precision than the control).

5.3.3.2. Observed model accuracy

The final examination of the operational verification study of the model is to investigate the level of accuracy yielded by the model. Table 5.8 summarises the μ statistics of the control data and the model data samples with the percentage of accuracy yielded by the model data.

Table 5.8.: Accuracy yielded by the model data samples

Stage	Control data μ (units of flow)	Model data μ (units of flow)	Accuracy (%)
1. 20 Concurrent sessions	1.4180×10^4	1.3852×10^4	97.69
2. 40 Concurrent sessions	2.1525×10^4	2.0921×10^4	97.19
3. 60 Concurrent sessions	2.5022×10^4	2.4020×10^4	96.00
4. 80 Concurrent sessions	2.9845×10^4	2.9134×10^4	97.62

The benchmark conditions stated an acceptance threshold, $\phi = 0.5$, that allows the output data of the model to be within a certain range from the control data. If the values of the model accuracy hold to the condition $|\frac{\text{model accuracy\%}}{100}| \leq \phi$, the data is within the acceptable accuracy range. The results presented in Table 5.8 show that the model operated within this acceptance threshold, and is therefore considered to yield data with sufficient accuracy.

5.4. Additional: model efficiency

The multi-commodity integer programming problem represented by the model and control, were solved with two fundamentally different methods, respectively. The solving method utilised in the model was the *heuristic* Simulated Annealing algorithm, while the control implementation utilised the *deterministic* CPLEX algorithm. The number of iterations needed by the control and model to produce the different data samples for the validation and verification process are presented in Figure 5.5.

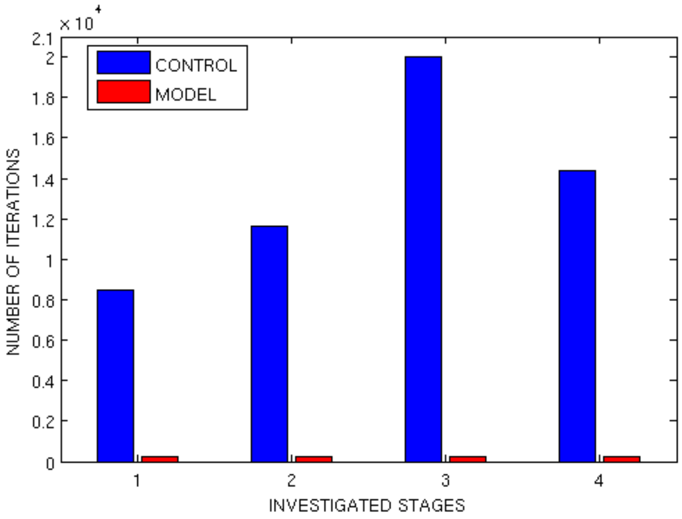


Figure 5.5.: Number of iterations needed in each stage

The 2.21% to 4% accuracy trade-off accepted for the model in the previous section, can be justified by the gain in execution time offered by the implementation of the Simulated Annealing algorithm. Figure 5.5 shows that the model outperformed the control in terms of execution time. The parameter settings of the Simulated Annealing algorithm allows for the terminating temperature to be adjusted in order to select the point where the algorithm would terminate. The calculated throughput at each iteration is shown in Figure 5.6.

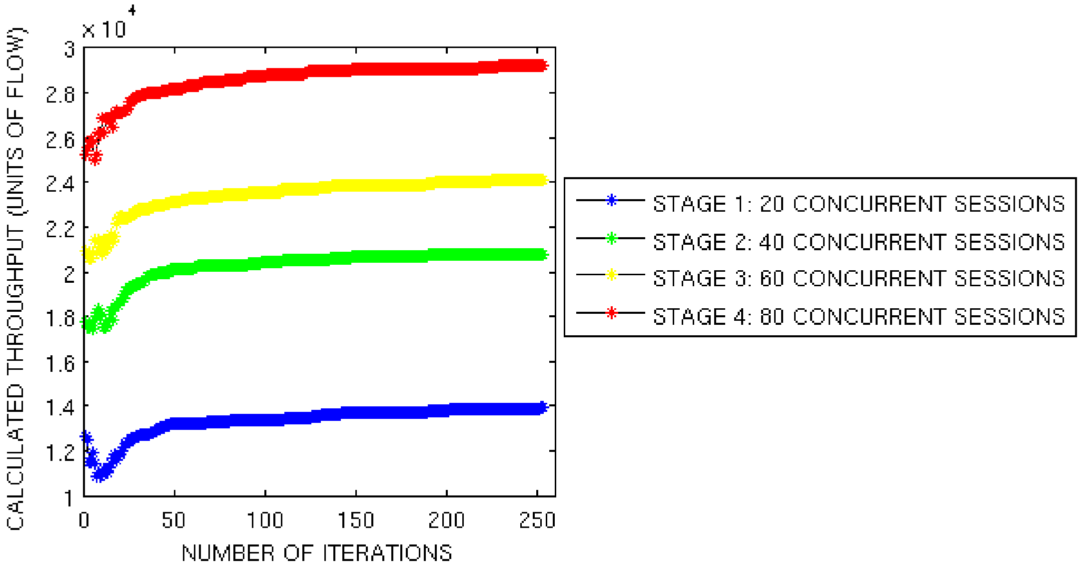


Figure 5.6.: Aggregated throughput at different algorithm iterations

The findings of Figure 5.6 show, that the Simulated Annealing algorithm yields throughput values, that approaches higher levels of accuracy as the number of iterations increase. Therefore, a trade-off in accuracy exists for a gain in the execution time. The overall impression is that the model yields data output with sufficient accuracy within efficient time.

The aim of this chapter was to validate the developed model and verify that the the results obtained with the model were reliable. Some background concepts and definitions were presented in section 5.1, followed by the benchmark conditions for the validation and verification in section 5.2.

Section 5.3.1, showed that the logic involved in the model development was an accurate description of real-world WMNs, as described by the background literature from chapter 2. The model format was conceptually verified against the format of the control.

Section 5.3.2, established the statistical validity of the model, and section 5.3.3 verified that the model operates within the acceptable accuracy range.

Additional to the validation and verification of the model, Section 5.4 provided information with regards to the efficiency of the solving algorithm implemented by the model. It was shown that the trade-off for accuracy, was accounted for by a drastic decrease in the number iterations needed by the model to solve multi-commodity integer flow problems.

The overall conclusion is that the model is a valid representation of WMN systems, that produces results with sufficient accuracy in efficient time.

6. Research conclusion

In this concluding chapter, the work done by the research is summarised in section 6.1, followed by a discussion pertaining to the findings of the study in section 6.2. The final remarks, given in section 6.3, provides the aspects left open by the study, which can be research topics in future work.

6.1. Overview of work done

This dissertation, "An optimisation approach to improve the throughput in wireless mesh networks through network coding", addressed issues that were concerned with the aggregated throughput performance of Wireless Mesh Networks (WMNs). WMNs are wireless networks that temporarily exist when nodes connect with each other, in an ad hoc manner, to form a mesh. The advantages offered by WMNs include low initial set-up costs, robustness, increased coverage ranges and the distribution of services.

WMNs are multiple hop wireless networks, in which routing through every network node is possible. This routing property of nodes, implies several operational and environmental challenges for WMNs. Current message forwarding mechanisms, together with the harsh conditions of a RF environment, have a decreasing effect on the aggregated throughput performance of WMNs.

Compared to conventional wireless networks, where messages only exist between each user and the AP, the messages in WMNs are conveyed by each node it encounter. The result is an increase in the load through the network, without an increase in the original number of concurrent sessions. The mesh topology also implies that each node have several neighbours, all active participants in the network. This causes multiple active links within the same broadcast wireless medium. The bandwidth, available for transmission, has to be shared amongst all these links, resulting in a steep decrease in the throughput yielded by WMNs.

Network Coding is a coding and decoding process at the network level of the OSI stack, aimed to improve the boundaries of saturated links. The technique implies that the bandwidth is simultaneously shared amongst separate message flows, by combining these flows at common intermediate nodes. The number of transmissions needed to convey information through the network, is decreased by Network Coding. The result is in an improvement of the aggregated throughput. This process is successful, if the designated receiver can restore the intended messages (decode the coded flow).

The research approach was concerned with developing a model to represent WMN systems. The model was utilised to observe the effect that the implementation of Network Coding have on the throughput performance of WMNs. The development of the model, was achieved by developing each aspect of a WMN system. These aspects were: 1. The representation of the physical environment and network elements as data structures; 2. The expression of the network load through the links, as mathematical equations; and 3. The method utilised to solve these equations.

The first aspect in the model development process, consisted of placing the number of nodes specified by the setup parameters in a mesh topology. The links between the nodes were established by implementing an indoors log-distance path loss channel model. Each link was also assigned a RSSI factor, indicating the quality of the link, related to the transmitting and receiving power at the respective nodes on the link.

The next step was to calculate the different paths through the network, for each concurrent session of the increasing load in the setup parameters. These paths consisted of multiple consecutive links between the source and sink nodes of each session, and were determined by distance vector and link-state methods. Some of these concurrent session paths yielded valid Network Coding opportunities, that were noted by an adjustment of the flow coefficients of the links involved. All this information was captured by the data structures produced by the set-up and representation procedures, conducted in MATLAB®.

The second model aspect, pertained to the session load through each link in the network. A network flow programming problem were developed to represent these flows, bounded by the capacity and coding constraints. Additionally, the flows of the consecutive links on a path were also subject to the law of conservation of flow, i.e. the mass balance constraint at each node. The goal of the network flow programming problem was to compute the maximum flow for multiple concurrent sessions over single paths.

The properties of the network flow programming problem, implied that the problem was considered to be mathematically equivalent to a multi-commodity integer programming problem. A complexity analysis revealed that the problem is NP-hard, therefore a heuristic approach was appropriate to solve the problem. The selected method for the problem was Simulated Annealing, which was implemented in a C++ program. This was the final aspect of the model development.

The model was conceptually validated by comparing the logic in the model design, juxtaposed with the properties of WMNs from the background literature study. In order to determine the validity of the model, and verify that the results it yielded was accurate, a control was developed and implemented by the deterministic CPLEX method. Data samples from an implementation of the developed model was compared against data samples yielded by the control, through statistical hypotheses tests. These procedures confirmed that the model yields results with sufficient accuracy and precision, in efficient time.

6.2. Findings and conclusion

The results of this study show that networks implementing Network Coding, outperforms networks without Network Coding. The network instances that implemented Network Coding nearly achieved double the calculated aggregated throughput values compared to networks that did not. Additional to this finding, networks that selected paths based on a link-state approach outperformed networks that selected paths based on a distance vector metric. These observations were true for every network instance investigated by the model implementation.

Pure WMNs yielded the highest aggregated throughput, with the highest improvement in network instances that utilised link-state path determination. This observation was due to the widely spread distribution of source and sink nodes throughout the network, providing multiple paths in opposite directions for ample Network Coding opportunities. The downside of pure WMNs, is that there is no connectivity to external networks.

Hybrid WMNs presented the observation that the links in the network reached a saturated level fairly quickly, compared to pure and access WMNs. The calculated throughput values increased only slightly as the size of the networks and the load through the network increased. This observation was due to the presence of the router backbone, that provided default shorter routes through the network. These routes became congested, which in turn delayed the aggregated throughput of the network. Network Coding opportunities existed in the regions surrounding the backbone, since the direction of the flows were mainly towards and from the backbone. However, these combined transmissions still had to share single links, and the scheduling effect limited the throughput increase potential. Hybrid WMNs are not connected to external networks, but offers the advantage of connecting smaller clusters of nodes, that would otherwise be disconnected from the network.

Access WMNs showed the least performance in terms of aggregated throughput. The direction of flow in the access WMNs were biassed towards the APs in the network. This characteristic caused the links towards the APs to became congested. It also implied that the occurrence of Network Coding opportunities were limited. The Network Coding opportunities had a higher probability to exist at the edges of the network instead of at the regions surrounding the APs, where the congestion was the most intense and would have benefit more from Network Coding. The advantage of access WMNs, is that connectivity to external networks is possible.

From these findings, the following conclusion is formulated: The addition of a Network Coding implementation in WMNs, yields an increase in the aggregated throughput of these networks, with the magnitude of the increase subject to the following:

- the dispersion of the traffic load through the network; and
- the directional pattern of the traffic through the network.

Therefore, WMNs will experience an increase in aggregated throughput by implementing Network Coding, that can be further improved by selecting flow paths based on a link-state method.

6.3. Recommendations for future work

The following questions are left open by this research study, and may provide research topics for future work:

- Wireless multiple networks experience the phenomenon that the quality of the message in the transmitted signal degrades along each hop en route to the destination. It is common practice to limit the number of hops on the paths of these network types, in order to preserve the transmitted messages. An open question is, does the implementation of Network Coding influence the critical value for the number of hops permitted? And if so, is it a positive or negative influence?
- The IEEE 802.11 Standard provides for devices with multiple channels. How can the channel assignment routines in these devices be utilised in conjunction with Network Coding, to provide optimal throughput increase in WMNs?
- The dispersion of the traffic through WMNs can be manipulated to be more widely spread if the flows are split and conveyed over multiple routes, in stead of a single path. Can these multiple paths be selected on a metric that will seek the conditions for optimal throughput increase offered by Network Coding?

This concludes the research study of this dissertation.

References

[1] S. Methley, *Essentials of Wireless Mesh Networking*. Cambridge University Press, 2009.

[2] I. F. Akyldiz and X. Wang, *Wireless mesh networks*. Wiley, 2009.

[3] E. Hossain and K. K. Leung, *Wireless Mesh Networks: architectures and protocols.* Springer, 2007.

[4] R. Bruno and M. Nurchis, "Survey on diversity-based routing in wireless mesh networks: Challenges and solutions," *Computer Communications*, vol. 33, no. 3, pp. 269–282, 2010.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *Computer Communication Review*, vol. 36, no. 4, pp. 243–54, Oct. 2006.

[6] R. Blake, *Wireless communication technology*. Delmar Thomson Learning, 2001.

[7] J. G. Proakis and M. Salei, *Digital communications*. McGraw-Hill, 2008.

[8] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, and S. Gheorghiu, "Multipath code casting for wireless mesh networks," in *Proceedings of 2007 ACM CoNEXT Conference - 3rd International Conference on Emerging Networking EXperiments and Technologies, CoNEXT*, 2007.

[9] S. Barros, "Mixing packets: pros and cons of network coding," in *The 11th International Symposium on Wireless Personal Multimedia Communications (WPMC08)*, 2008.

[10] R. K. Ahuja, T. L. Magnantl, and J. B. Orlin., *Network flows: theory, algorithms and applications.* Prentice Hall, 1993.

[11] J. W. Chinneck, *Practical optimization: a gentle introduction*. Carleton University Publication, 2007.

[12] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*, 4th ed. Wiley, 2010.

[13] P. Kosso, *A Summary of Scientific Method.* Springer, 2011.

[14] W. Goddard and S. Melville, *Research methodology: an introduction*, second. ed. Juta., 2007.

[15] *ILOG CPLEX 10.0 Users Manual*, 10th ed., ILOG S.A. and ILOG, Inc., January 2006.

[16] IBM. Ibm ilog cplex optimizer. [Online]. Available: http://www-01.ibm.com/software/

integration/optimization/cplex-optimizer/

[17] (2010, July) Mesh networks. [Online]. Available: http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Wireless+Broadband+Networks/Mesh+Networks/

[18] (2010, July) Meshdynamics mesh technology in operation. [Online]. Available: http://www.meshdynamics.com/Mesh-Network-Applications.html

[19] D. Johnson, K. Matthee, D. Sokoya, L. Mboweni, A. Makan, and H. Kotze, *Building a Rural Wireless Mesh Network*, second edition ed., Wireless Africa, Meraka Institute, South Africa, August 2007.

[20] B. A. Forouzan, *Data communications and networking.* McGraw-Hill, 2007.

[21] R. Olexa, *Implementing 802.11, 802.16, and 802.20 Wireless Networks. Planning, Troubleshooting, and Operations.* Newnes, Elsevier, 2004.

[22] M. L. Sichitiu, "Wireless mesh networks: Opportunities and challenges," in *Proceedings - 6th World Wireless Congress, WWC.* San Francisco, CA, United states: Delson Group Inc., 2005, pp. 318–323.

[23] C. Rosenberg, "Challenges in multi-hop networks," in *2006 2nd Conference on Next Generation Internet Design and Engineering, NGI 2006.* Valencia, Spain: Inst. of Elec. and Elec. Eng. Computer Society, 2006, pp. xvii–xvii.

[24] H. Schulzrinne. (2008, October) Long-term traffic statistics. [Online]. Available: http://www.cs.columbia.edu/~hgs/internet/traffic.html

[25] T. S. Rappaport, *Wireless Communications - Principles and Practice.* Prentice Hall, 1996.

[26] D. Apostolakis and P. Constantinou, "Man made noise measurements and modelling," *2nd International Conference on Universal Personal Communications, Personal Com- munications: Gateway to the 21st Century*, vol. 2, p. 585 589, 1993.

[27] J. Irvine and D. Harle, *Data communications and networkins: An engineering approach*, 1st ed. Wiley, 2001.

[28] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks." *Ad Hoc Networks*, vol. 2, pp. 1–22, 2004.

[29] M. R. Garey and D. S. Johnson, *Computers and intracability: a guide to the theory of NP-completeness.* Bell Telephone Laboratories, Incorporated., 1979.

[30] G. Longo and M. Venturini Zilli, "Complexity of theorem-proving procedures: Some general properties." *Revue Francaise d'Automatique Informatique Recherche Operationnelle*, vol. 8, pp. 5–18, 1974.

[31] C. Fragouli, J.-Y. L. Boudec, and J. Widmer., "Network coding: An instant primer," *Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[32] R. W. Yeung, *Information theory and network coding.* Springer, 2008.

[33] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[34] R. Koetter and M. Medard, "An algebraic approach to network coding." *IEEE/ACM Trans. Netw.*, vol. 11, pp. 782–795, 2003.

[35] T. Ho, M. Medard, J. Shi, M. Effros, and D. Karger, "On randomized network coding." *Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing.*, 2003.

[36] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *IEEE International Symposium on Information Theory (ISIT)*, 2003.

[37] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen., "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51 no 6, pp. 1973–1982, 2005.

[38] R. Yeung, S. Li, N. Cai, and Z. Zhang, "Network coding theory, foundation and trends," *Communications and Information Theory*, vol. 2(3,4), pp. 241–381, 2005.

[39] *IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Electronic, IEEE Std., 2007.

[40] WildPackets, *Converting Signal Strength Percentage to dBm Values*, WildPackets, Inc., November 2002.

[41] V. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "Algorithmic aspects of capacity in wireless networks," in *Performance Evaluation Review*, vol. 33, no. 1. Banff, AB, Canada: Association for Computing Machinery, 2005, pp. 133–144.

[42] D. Wang and Y. Kuo, "A study on two geometric location problems," *Information Processing Letters*, vol. 28, p. 281286, 1988.

[43] P.-J. Wan, "Multiflows in multihop wireless networks," *Department of Computer Science Illinois Institute of Technology, MobiHoc'09*, 2009.

[44] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science. New Series 200*, vol. 4598, p. 671680, 1983.

[45] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm." *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[46] J. Devore and N. Farnum, *Applied statistics for engineers and scientists.*, 2nd ed. Thomson Brooks/Cole, 2005.

[47] J. D. Camp and E. W. Knightly, "The ieee 802.11s extended service set mesh networking standard," *Communications Magazine, IEEE*, vol. 46:8, pp. 120 – 126, 2008.

[48] Y.-D. Lin, S.-L. Chang, J.-H. Yeh, and S.-Y. Cheng., "Indoor deployment of ieee 802.11 s mesh

networks: lessons and guidelines," *Ad Hoc Networks*, vol. 9:8, 2011.

[49] A. M. Law and M. G. McComas, "Secrets of successful simulation studies," in *Winter Simulation Conference Proceedings*. Phoenix, AZ, USA: Publ by IEEE, 1991, pp. 21–27.

[50] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings - Winter Simulation Conference*. Baltimore, MD, United states: Institute of Electrical and Electronics Engineers Inc., 2010, pp. 166–183.

[51] I.-R. W. coordinator. (2010, July) V.431 : Nomenclature of the frequency and wavelength bands used in telecommunications. International Telecommunication Union - ITU. [Online]. Available: http://www.itu.int/rec/R-REC-V.431/en

[52] J. C. Nash, "The (dantzig) simplex method for linear programming," *Computing in Science and Engineering*, vol. 2, no. 1, pp. 29–31, 2000.

[53] W. Cunningham, "Theoretical properties of the network simplex method." *Mathematics of Operations Research*, vol. 4, no. 2, pp. 196–208, 1979.

[54] L. Systems. Optimization software: Integer programming, linear programming, nonlinear programming, stochastic programming, global optimization. LINDO Systems. [Online]. Available: http://www.lindo.com/

[55] AIMMS. Xpress solver for linear and mixed integer programming. AIMMS. [Online]. Available: http://www.aimms.com/features/solvers/xpress

[56] Abacus - a branch-and-cut system. COIN-OR. [Online]. Available: https://projects.coin-or.org/ABACUS

[57] CORAL. Minto - mixed integer optimizer. [Online]. Available: http://coral.ie.lehigh.edu/~minto/

[58] F. W. Glover, "Tabu search, part 1," *ORSA Journal on Computing*, vol. 1, pp. 190–206, 1989.

[59] F. Glover, "Tabu search, part 2," *ORSA Journal on Computing*, vol. 2, pp. 4–32, 1990.

[60] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *SIMULATION*, vol. 76, pp. 60–68, 2001.

[61] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *International Journal of Bio-Inspired Computation*, vol. 1, 2009.

A. Additional information on the RF environment

A.1. RF Spectrum allocation

RF signals act differently at different frequencies, which influence the travel pattern and application of the signal type. To support the amount of spectrum needed in broadband systems, only frequencies above 700 MHz are really of interest. Above a certain point, around 6 GHz, RF wavelengths become very short (acting like a light beam), and does not bend around objects in its travel path any more . At this frequencies the applications are point-to-point based [21].

The use of frequency bands of the RF spectrum is regulated by several governments and other regulatory bodies in a initiative referred to as *Spectrum allocation* management. These management bodies include:

- The International Telecommunication Union, ITU.
- The European Telecommunications Standards Institute, ETSI.
- The International Special Committee on Radio Interference, CISPR.

Table A.1 gives a summary of the different frequency bands, as assigned by the International Telecommunication Union (ITU) [51]:

Table A.1.: High bandwidth systems RF spectrum allocation

Frequency Range:	Symbol:	Applications:
300 to 3000 MHz	UHF	Microwave ovens, television broadcast, GPS, mobile phone (GSM, UMTS, 3G, HSDPA), cordless phones,Wi-Fi, Bluetooth
3 to 30 GHz	SHF	DBS satellite television broadcasting, Wi-Fi, WiMAX, radars
30 to 300 GHz	EHF	Directed-energy weapon (Active Denial System), security screening (Millimeter wave scanner), intersatellite links, WiMAX, high resolution radar

A.2. IEEE802.11 Standard

The Institute of Electrical and Electronics Engineers (IEEE) formulated the first IEEE Standard 802.11-1997 [39], in 1997 for WLAN technologies. This standard defines the medium access control and physical layers for local area networks in the the RF environment. This standard has since been updated with several amendments. The 2008 amendment, IEEE Std. 802.11s, pertains to wireless local area **mesh** networks. Table A.2 list some of the standard amendments, together with several characteristics for these network types [39], [47].

Table A.2.: Summary of IEEE wireless network standards

	WLAN: Wi-Fi Standard			
Amendment	Spectrum (GHz)	Data rates (Mbps)	Modulation	Range (m)
802.11a	5.15 - 5.35 5.425 - 5.675 5.725 - 5.875	6 - 54	OFDM	Indoor: 30 Outdoor: 120
802.11b	2.4 - 2.497	1 - 11	CCK	Indoor: 40 Outdoor: 140
802.11g	2.4 - 2.497	1 - 54	OFDM & CCK	Indoor: 40 Outdoor: 140
802.11n	2.4 - 2.497 5.15 - 5.35	7 - 72 15 - 150	OFDM	Indoor: 70 Outdoor: 230
802.11s	2.4 - 2.497 5.15 - 5.35	1 - 11 (actual throughput: 0.1 - 0.45)	WiP*	Indoor: 40 Outdoor beacons: 100 - 500

* Work in Progress. Aim: to find efficient Modulation and Coding Schemes (MCSs).

A.3. The wave front and Fresnel Zone

Waves radiating from a single point (isotropic radiator) has a spherical wave front. A *wave front* is explained as the plane in which the wave travels, where all the waves has the same phase [21], [6]. Isotropic radiating antennas exist only theoretical. In practice, dipole antennas exists that generates a 360° wave front. These are omnidirectional antennas [21], [6].

The power density at any point away from the source is given as:

$$P_D = \frac{\text{EIRP}}{4\pi r^2} \tag{A.1}$$

were EIRP is Effective Isotropic Radiated Power.

Free space do not absorb energy, so there is no loss of energy. However, as the distance away from the wave source increase, the energy is spread over a larger wave front, leading to attenuation. Additionally, there exists an area of concentric ellipsoids around the straight LOS that connects a receiver and transmitter. This area corresponds to the pattern the RF waves spread out to, after it leaves an antenna, and before it enters an antenna. This phenomena is explained by the fact that microwave beams widen, together with Huygens principle, stating that at each point of a wave front new circular waves start.

The area of concentric ellipsoids is called the Fresnel Zone, shown in Figure A.1.

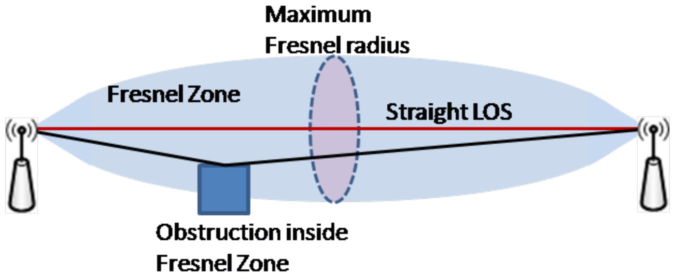


Figure A.1.: Effect of obstructions within the space of the Fresnel Zone

Even with a clear LOS, an obstructed Fresnel Zone hinders the propagation of RF waves. This effect causes many reflections (constructive and destructive) on the waves, including multipath distortions, fading and shadowing [21], [6].

A.4. The hidden node problem

The hidden node problem arises when a node in the RF environment is “visible” from an intermediate node, but “hidden” to the other nodes connected to this intermediate node. This phenomenon leads to difficulties for medium access control, since the hidden node cannot detect current transmissions from the other nodes to the intermediate node. If the hidden node starts to transmit itself, this will cause interference with the current transmission. Figure A.2 shows the hidden node effect:

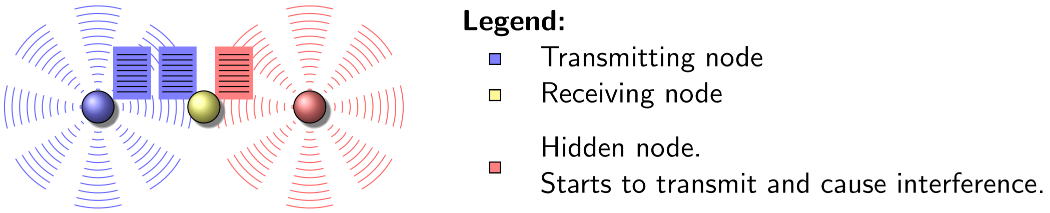


Figure A.2.: Interference caused by a hidden node.

B. Network flow problems: some algorithms

B.1. Elementary: the augmenting path algorithm

The intuitive method to find the maximum network flow, is to gradually increase (augment) the flow over the links of the path, P, between the source, s, and sink, t nodes. This increase continues until the upper boundary (capacity) is reached, an is called the Augmented Path between s and t. Given a certain flow through a link, $f_{(j,i)}$, and the capacity of that link, $c_{(j,i)}$, two possibilities exist: the link can either be *saturated* ($f_{(j,i)} = c_{(j,i)}$), or there exists a potential buffer that depicts the amount with which the flow can still increase ($f_{(j,i)} \leq c_{(j,i)}$).

This is the **residual capacity** of the link, and stated as:

$$c_f(j,i) = c_{(j,i)} - f_{(j,i)}, \text{ where } c_{(j,i)} \geq 0.$$

In a network where $c_f(j,i) > 0$ for a feasible flow $f_{(j,i)}$ from s to t, the **residual network** G$_f$ with respect to f is given as:

$$G_f = (V, E_f) \text{ with edge pairs } (j,\ i).$$

The residual network is a measurement of the amount of capacity "available" in the network. For a path $P = (u_1, u_2, u_3, \ldots, u_k)$ in the residual network where $u_1 = s$,$u_k = t$ and $c_f(u_i, u_{i+1}) > 0$, P is called the **augmenting path**.

$$\delta(P) := min\{c_f(j,i)|(j,i) \in P\} \text{ and } \delta(P) > 0 \text{ by definition.}$$

A network is at maximum flow if and only if there is no augmenting path in the residual network. To augmenting the flow on an edge (j,i) in the residual network, flow $f_{(j,i)}$ is replaced by flow $f'_{(j,i)}$, defined by the following method:

Let P be an augmenting path for G with respect to a flow f. $\forall$ (j, i) $\in P$, let

$$f'(j,i) = f(j,i) + min\{\delta(P), c(j,i) - f(j,i)\}$$

and

$$f'(j,i) = \begin{cases} f(j,i) - \delta(P) - (c(j,i)f(j,i)) & \text{if } \delta(P) > (c(j,i) - f(j,i)) \\ f(j,i) & \text{otherwise} \end{cases}$$

Then, f' is feasible and $|f'| = |f| + \delta(P)$.

B.2. Deterministic: simplex

A very successful method for solving linear programming minimum and maximum problems is the **simplex** algorithm, developed by Geroge B. Dantzig (1947) [11], [52]. In the subject of network flow programming, the simplex method still forms a backbone for many solving techniques to this date. In the absence of degeneracy, the simplex algorithm leads to convergence of a feasible solution, in a finite number of iterations. The simplex method is used for linear programming problems in the standard form, $Ax \leq b$. Define the following:

$$A_I = \begin{bmatrix} a_{i_1}^T \\ \vdots \\ a_{i_K}^T \end{bmatrix}, \; I = \{i_1, \ldots, i_K\}$$

where I is the set of active constraints at x: $a_k^T x = b_k, \; k \in I$ and $a_k^T x < b_k, \; k \notin I$.

One iteration of the simplex method are as follows:

Algorithm 6 Simplex Iteration

```
1. Calculate $z \in \Re^m$ with $A_I^T z + c = 0, \quad z_j = 0, \quad j \notin I$
2. if $(z \geq 0)$
3.    $x$ optimal, break
4. else
5.    do
6.    Choose $k$, $z_k < 0$
7.    Calculate $\Delta x \in \Re^n$ with $a_i^T \Delta x = 0, \quad i \in I - k, \quad a_k^T \Delta x = -1$
8.    until $A\Delta x \leq 0$
9. end
```

The objective of linear programming problems is solved through the Dantzig Algorithm, given as follows:

Algorithm 7 Dantzig (Simplex) Algorithm, 1947

```
1. Rewrite objective to equal 0.
2. Remove inequalities (add slack values s and t).
3. Place in tableau form.
4. Determine pivot column.
5. Determine pivot row.
6. Divide (pivot=1).
7. do
8.      ITERATION in Algorithm 6
9. until top element < 0.
```

Some interesting properties embark when casting the network flow problem as a linear program. A closer look at the constraints coefficients reveal that they are either 0, +1, or -1 [53]. The reason is that in a network, the node relationships are summations and subtraction of flows, and the remainder of the constraints are the bounds. The consequences of this observation are:

- During linear programming execution the pivot steps consists of addition and subtraction, no multiplication is needed. This eliminates floating-point computer operations leading to much faster arithmetic operations.

- The addition and subtraction operations can be replaced by logical operations, and when applied to network flow problems resolve to different solving algorithm that is significantly faster than the original simplex method. This is the **network simplex method** [11], [53].
- The unimodularity property for integrality still holds - if all of the constraint remainders are integers, and if all of the pivot operations are additions and subtractions, then the solution values of the variables at the optimum will also all be integers. This means that many integer programming problems (solved with slower methods) can now be solved with linear programming for the same problem (a faster solving method).

In cases where the network simplex method cannot be applied directly, some solvers actually identify portions of the problems that are in the form of network flow problems and solve that via the network simplex method [11].

Examples of commercial (free and licensed) and academical solvers are:

- IBM - ILOG CPLEX Optimizer [16].
- LINDO Systems - Optimization Software: Integer Programming, Linear Programming, Non-linear Programming, Stochastic Programming, Global Optimization [54].
- AIMMS - XPRESS Solver for Linear and Mixed Integer Programming [55]
- ABACUS (A Branch-And-CUt System) [56]
- MINTO - Mixed INTeger Optimizer [57]

B.3. Heuristic methods

For dimensionally large optimisation problems, such as in the case for multi-commodity integer problems, exact deterministic methods tend to become processing intensive and running time for algorithms becomes exhausted and impracticable. In cases like this, another approach to the solving method are considered, namely an **heuristic** or **probabilistic** approach.

Heuristic methods uses an experience-based discovery method in order to solve a given problem, such as *trial and error*. The decision to accept a particular state in the algorithm as the solution, is dependant on some defined probability function. Heuristics are intended to gain computational performance and execution speed, at a potential cost in *accuracy*. The accepted solution may not be the exact optimum, but an approximation close enough to the exact optimum.

B.3.1. Simulated Annealing

Simulated Annealing is a probabilistic solving method independently developed by Kirkpatrick, Gelett and Vecchi (1983) [44], and Cerny (1985) [45]. In the metallurgical and material science environment, *annealing* is a purification technique that entails controlled heating and cooling of materials. As the material is heated the atoms of its structure becomes excited and disassociate itself from their initial energy level and position, causing it to flow freely and in a random fashion through states of higher energy levels. When the material is cooled down slowly, the atoms settles in a configuration with a lower energy level than its initial state. This enlarge the size of the material's crystals and therefore reduce its defects and impurities.

The Simulated Annealing technique works analogues to this process. The initial energy level is seen as the initial configuration of the variable parameter, and it is currently in a local minimum or maximum. The next step is to replace the current configuration with a randomly selected nearby configuration. This new configuration is selected on a probability basis, depending on the change in function values based on the current value of the global parameter *T*. *T* is the *temperature* variable and gradually decreases during the Simulated Annealing process, like in the physical annealing technique. Simulated Annealing is characterised as a generic probabilistic and heuristic method, to solve the global optimum of a function as an acceptable approximation when the search space or domain is very large and tends to exhaust traditional numeration techniques.

B.3.2. Tabu search

Tabu search is a heuristic optimisation method that examines a trajectory sequence of solutions and moves to the best neighbour of the current solution [58], [59]. Tabu search can be be superimposed onto other procedures to prevent them from being trapped at local optimum solutions in order to move towards the global optimum.

To avoid cycling, solutions that were recently examined are kept in memory structures. These solutions are forbidden, or taboo, for a number of iterations to prevent the algorithm from determining it repetitively. The Tabu search algorithm uses iterations to replace one possible solution, X with a neighbouring solution X' within the search space $\mathcal{X}$, until the stopping criteria is reached.

B.3.3. Harmony search

The *Harmony Search* method mimics the improvisation process of composing musicians [60]. In this analogue, the *musicians* are the vector of **decision variables**, generating *notes* or the **values** with the objective of finding the best *harmony*, the **optimal solution**.

B.3.4. Intelligent water drops

The behaviour of natural water drops in rivers flowing from its source to destination are the phenomenon mimicked by the *Intelligent Water Drops* method [61]. Rivers often find "good" paths amongst a choice of possible paths en route from the source to the destination of the river. The actions and reactions between the different water drops flowing in the river, and the water drops and the river bed are the input variables in a attempt to determine the optimal river path, i.e. the optimal solution to the underlying problem.

B.3.5. Heuristic methods comparison

The four heuristic methods discussed in Section B.3.1 to B.3.4 is small selection of the number of methods available in mathematical communities today. These methods represents the main different types of algorithm groupings:

- Controlled random search: Simulated Annealing, Tabu Search.
- Pure stochastic search: Harmony Search.
- Swarm-based optimisation: Intelligent Water Drops.

Important to notice is that not all heuristic methods are as well developed, and therefore care must be taken in selecting the right method when developing an algorithm.

A comparison of the four methods in Section B.3.1 to B.3.4 is shown with regards to the following criteria:

- Applicability to network flow problems.
- Established success rate (based on number of citations).
- Ease of implementation.
- Sufficient documentation available on the particular method.

Table B.1 shows the heuristic methods, assigned with a "⋆" rating, indicating how well the method adhere to the criteria stated above. A "⋆⋆⋆" rating indicates sufficient adherence to the criterion, while a "⋆" rating indicates insufficient adherence.

Table B.1.: Comparison of heuristic methods

Method:	Network flow problems	Success Success	Ease of implementation	Documentation available
Simulated Annealing	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆
Tabu Search	⋆⋆⋆	⋆⋆⋆	⋆⋆	⋆⋆⋆
Harmony Search	⋆⋆⋆	⋆	⋆	⋆
Intelligent Water Drops	⋆⋆⋆	⋆	⋆	⋆

C. Optimisation with CPLEX

C.1. Background information on CPLEX

A implementation of the simplex algorithm in the C language was originally developed by Robert E. Bixby, as the CPLEX algorithm. CPLEX continues to be actively developed under IBM as the ILOG CPLEX Optimizer[1], since it was acquired by the company in January 2009 [16]. CPLEX runs under the control of the ILOG License Manager (ILM). A free license is offered for a trial version of CPLEX, however this version is very limited as to the size of the variable vector it can solve. The fully licensed program needs to be purchased. However, there is an option to register at the Academic Initiative from IBM, to obtain a downloadable full version of the algorithm together with a one-year licence.

The following types of problems can be solved via the ILOG CPLEX Optimizer [15]:

- Integer and Mixed Integer Programming (MIP) problems.
- Linear Programming (LP) problems.
- Convex and Non-convex Quadratic Programming (QP) problems.
- Convex QP problems with quadratic constraints.

The following steps are executed in the creation of a CPLEX application [15]:

1. Extract the model object of the problem. Several modelling objects can be created to specify the optimisation problem. Those objects are grouped into an IloModel object representing the complete optimisation problem.
2. Solve the model by handing the model object over to the CPLEX solver. An instance of IloCplex are used to solve modelling objects. An instance of IloCplex reads a model and extracts its data to the appropriate representation for the CPLEX optimisers.
3. Access the solution information access to interpret results from the optimisation after the model is solved.

Concert Technology objects embedded in the optimiser are used by CPLEX applications as a modelling layer to provide interfaces to the C++, C♯, and Java languages. Figure 6.1. shows a diagram on how the the architecture of a C-based language CPLEX Application incorporate the ILOG classes and Concert.

---

[1] American spelling convention for “Optimization” is used and not “Optimisation”, since the word is included in the actual name of the solver.

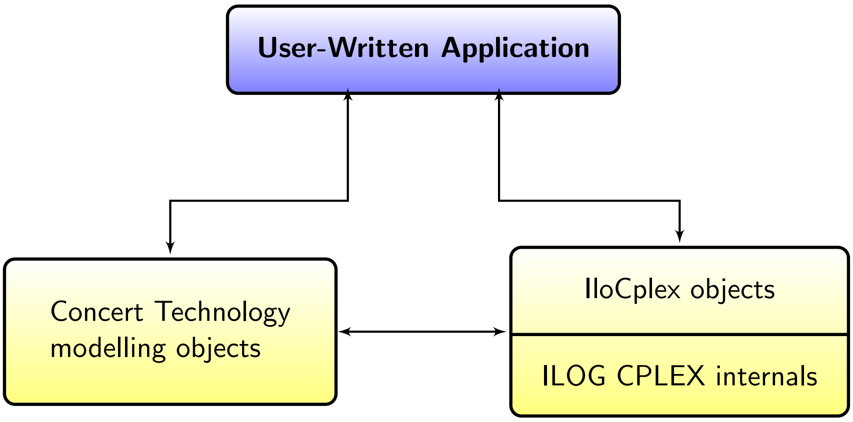


Figure C.1.: Diagram of the CPLEX Application, taken from the ILOG CPLEX User's manual

The steps necessary to solve a CPLEX model are as follows:

1. Create an ILOG CPLEX environment.
2. Instantiate a problem object. This is the IloModel object.
3. Populate the problem object with data.
4. Call a CPLEX optimiser. For a network flow problem, the appropriate optimiser will be the ILOG CPLEX Network Optimizer[2].

C.2. Creating and implementing the CPLEX control

The control model utilised in the validation and verification procedures in chapter 5 of this dissertation, was selected as a CPLEX model. This section explains how the control is developed in the correct format, i.e. an IloModel object [15].

C.2.1. Basic case network flow problem for the control

Developing the CPLEX control model, was based on the basic case network flow problem provided as an example in the user manual of the ILOG Optimizer [15]. The *objective* is to find the minimal-cost flow through a network, where the *network* consists of a set N (the *nodes*) and a set A (the *arcs*) connecting the nodes. An arc a in the set A is an ordered pair (i, j) where i and j are nodes in the set N. Node i is referred to as the *tail* and node j the *head*, with the direction *from* tail *into* head on the arc.

Four values are associated with each arc. These values are:

- x_a: The amount of *flow* passing through arc a from its tail to its head. The flow values are the modelling variables of a network-flow problem.
- l_a: The lower bound. This value determines the minimum flow allowed through arc a.
- u_a: The upper bound. This value determines the maximum flow allowed through arc a.

---

[2] American spelling convention for "Optimizer" is used and not "Optimiser", since the word is included in the actual name of the optimiser.

- c_a: The objective value, determines the contribution to the objective function of one unit of flow through the arc. Also referred to as the *cost* of the flow on arc a.

Each node, n is also associated with one *supply* value of the node, s_n. There are three possibilities for the value of s_n:

- $s_n > 0$: Node n is a *supply* node or a *source*.
- $s_n < 0$: Node n is a *demand* node or a *sink*.
- $s_n = 0$: Node n is a *transshipment* node or an *intermediate node*.

Additionally, every node n is also associated with two sets of arcs expressed as:

- T_n: The set of arcs whose tails are node n.
- H_n: The set of arcs whose heads are node n.

Analytically, the format of the control is presented by the following equations:

Minimise:

$$\sum_{a \in A} c_a \, x_a \tag{C.1}$$

subjected to:

$$\sum_{a \in T} x_a \; - \; \sum_{a \in H} x_a \; = \; s_n \qquad \forall \, n \, \in \, N$$

with bounds:

$$l_a \; \leq \; x_a \; \leq \; u_a \qquad \forall \, a \, \in \, A$$

In order to calculate the maximum throughput of a path flow, the optimisation problem is modified as follows:

$$\max \sum_{a \in A} c_a \, x_a = 1 - \min \sum_{a \in A} c_a \, x_a$$

C.2.2. Validation and verification benchmark conditions

The benchmark conditions for the validation and verification procedures, had certain implications on the format of the expression of the control model. In order to adhere to the parameters of the benchmark conditions, some additional properties were assigned to the control. These properties are:

- To keep the flows of the control over single paths, the integer problem property were implemented in the coefficients of the supply factors, s_n, $\forall \, n \, \in \, N$:
  - coefficient of $s_n = 1$ for source nodes.
  - coefficient of $s_n = -1$ for sink nodes.
  - coefficient of $s_n = 0$ for intermediate nodes.

  Additionally to these coefficient settings, a path variable vector, y, was implemented in order to keep the link flows on a single path. The values of these variables was $y_a \, \in \{0, 1\} \, \forall \, a \, \in \, A$
- All links must operate at 100%, therefore no cost reduction were implemented: $c_a = 1 \, \forall \, a \, \in \, A$.
- The flow should not become negative (change direction on a link), therefore the lower bound were selected as $l_a = 0$.

- A channel capacity value, C, is assigned to each arc as an upper bound, $u_a = C$.
- A commodity flow, λ_k, was assigned to the control which represented the flow of each commodity k.

The goal function was modified as follows to account for multiple commodities:

$$TP = \max \sum_{k=1\ldots s} \lambda_k \tag{C.2}$$

where k is a commodity in the set of concurrent sessions ξ, λ is the calculated flow of each commodity and TP is the aggregated throughput.

D. Conference contribution

A contribution to the Southern Africa Telecommunication Networks and Applications Conference (SATNAC), was made from the work done in this dissertation. The publication, “The Effect of Network Coding on the Network Throughput of Wireless Mesh Networks” were presented at the 2011 SATNAC proceedings, in East London, 4 to 7 September 2011.

The Effect of Network Coding on the Network Throughput of Wireless Mesh Networks

Corna van der Merwe & Leenta M.J. Grobler
School of Electrical, Electronic and Computer Engineering
North-west University, Potchefstroom Campus, South Africa
Email: 20047290@nwu.ac.za, leenta.grobler@nwu.ac.za
Telephone: (+27)18 299-1978 Fax: (+27)18 299-1977

***Abstract*—Wireless Mesh Networks (WMNs) are communication networks where users, routers, APs and any other device with networking capabilities can forge a wireless ad hoc network amongst themselves. This offers many advantages such as a decrease in network set-up costs, increased network coverage, access opportunities to external networks (previously not possible without the right equipment) and even access to different network types such as sensor networks, cellular networks, computer networks, wired networks etc. It also implicates several concerns or issues regarding security, network management, and ever present scalability issues. Since the wireless nodes in a WMN broadcast their data by nature, there will always be redundant traffic present in data transmissions. Network coding offers a way to improve on the network performance, by sharing network resources while at the same time keeping information separate. This paper studies the effect of network coding on the aggregated throughput in WMNs by means of an optimisation model. From the results it is concluded that bandwidth utilisation improves for the networks that implement a network coding scheme, and therefore the scalability of these networks improves - a higher throughput can be obtained with an increase in the number of nodes without other altercations to the network such as different routing methods.**

***Index Terms*—network coding; optimisation; scalability; wireless mesh network;**

I. INTRODUCTION

Wireless Mesh Networks (WMNs) are collections of wireless nodes that establish network connectivity amongst themselves in an ad hoc manner to form a mesh topology. The novelty in this configuration is that any node, be it a router or client, can perform routing. This concept allows for a distributed load through the network and gives intra-network access to any destination if a node is in the vicinity of the broadcasting and reception range of another node. If the WMN contains bridging nodes to external networks this allow for any node in the WMN to also access the external network. As a result WMNs can connect multiple networks (and implicitly different network types, like sensor networks, computer networks, mobile networks and even wired networks). The issues associated with WMNs are contributed to the characteristics of this network type, and are classified as network management issues, routing method issues, security issues and scalability. This paper intends to address the **scalability issues** by introducing a way of decreasing network load in the network via network coding. Section II gives brief background literature and descriptions of the concepts, technology and terminology used in this paper, Section III describes the system model developed to examine the effect of network coding in WMNs, Section IV presents the results of this model with the concluding remarks of the results in Section V.

II. BACKGROUND

A. Wireless Mesh Networks

WMNs are infrastructure-less networks that exist when a collection of wireless users temporarily establish and manage connectivity amongst themselves in an ad hoc manner [1], [2], [3].

Advantages of WMNs include: increased reliability through path redundancy between sources and sinks [2]; distributed management - there are no main nodes to form a single point of failure; routing through clients are possible [1], [3]; access to external networks through bridging nodes - limits on the technical specifications of clients are therefore overcome [1], [3]; lower initial cost due to the fact that permanent infrastructure such as base stations are not needed to create a WMN and additional nodes can be added incrementally to the network as they are obtained or needed [1], [2], [4].

An additional benefit is the improvement in coverage performance. Spectral efficient modulation schemes gave the data transfer rates of IEEE802.XX wireless devices a great boost, for example 11 Mbps to 54 Mbps. However, this increase in rate tends to decrease the connection and coverage in centralised networks when the distance between the APs (Access Points) and end-users becomes too great. Multiple hops and channels utilised in WMNs can enable longer distance wireless communication without degrading the network performance [1], [2].

Three types of WMNs are identified: pure mesh networks, access mesh networks and hybrid mesh networks.

- **Pure Mesh Networks:** Also referred to as *wireless clients networks* [2]. This type of WMN only contains users (client nodes), forming a static ad hoc network amongst themselves, with no backbone routers [1], [3].
- **Access Mesh Networks:** Also referred to as *wireless client-router network* [2]. This WMN type contains a router backbone with gateway functionality, allowing for external-network connectivity. This can either be with other WMNs, the Internet, or even different network types such as wired, cellular or sensor networks [1], [3].

- **Hybrid Mesh Networks:** Users have access to other users through a backbone of routers. There is no external network connectivity [1], [3].

B. Scalability issues in WMNs

The origin of most scalability issues in multiple hop wireless networks can be contributed to the shortcomings of the routing methods that currently exist for wireless networks.

Proactive Routing protocols entails that nodes periodically interchange routes amongst each other in order to maintain an updated view of all possible destinations in the network from their position, even when these paths are not currently needed [5], [6], [7]. This type of routing is useful in networks where topology changes are frequent, but has the disadvantage that the constant flooding of update messages through the network add complexity to the network management. Every path (regardless of need) must be maintained in order for successful and accurate routing, resulting in a massive control message overload consuming the available bandwidth. Examples are DSDV, WRP, OLSR, GSR and HSR.

Reactive Routing Protocols aim to reduce and minimise the unnecessary network load resulting from constant updates. Paths are acquired on demand (when needed) through a process called route discovery. Only the paths actively in use are maintained [5], [6], [7]. Reactive Routing Protocols have have proved to improve network performance in small networks with limited or very low bandwidth available, however it still does not scale well with large networks, especially when route discovery needs to be conducted over multiple intermediate nodes. An added constraint is the acquisition latency resulting from route discovery, which leads to delays in packet forwarding and even packet dumps due to full buffers [5]. Examples are DSR, AODV, ABR, SSR and TORA.

Hybrid Routing Protocols consists of both proactive and reactive components, in an attempt to combine the advantages offered by these methods and overcome their shortcomings [5], [7], [8]. Examples are ZRP, ADV, SHARP and DST.

Every routing protocol has a main algorithm that base the decision of the best next hop on an *evaluation metric* such as Hop Count, Packet Loss Ratio, End-to-end Delay, Route Stability etc., [8]. In most cases for mobile ad hoc networks the Hop Count metric are implemented, which tends toward the shortest path selection. This effect increase network congestion around popular routes. The distributed node configuration of WMNs allows for methods that try to avoid this phenomena by routing for load balancing, link sates, and routing over multiple paths [9]. In effect it just shows that hop count is not a preferred metric for WMNs, which was the focus of Jun and Sichitiu [10] where a new routing method MRP together with additional evaluation metrics is proposed. MRP attempts to reduce the overhead by eliminating message flooding through a spanning tree scheme. This method shows potential improvement for access mesh networks, but no consideration is given to pure mesh networks. Another disadvantage of MRP is that a spanning tree method is used, creating single points of failure in the network should a tree's root node becomes inactive. A broader examination of routing protocols reveals that in aim of achieving sustainable routing over larger networks, the traffic load through the network plays an important role to the performance of the routing algorithm. Therefore, routing performance can be enhanced by altering the load through the network, more specifically *reducing* the load through the network. This is the aim of network coding, which is discussed in the following section.

C. Network Coding

Network Coding refers to a form of data encoding and decoding scheme at the Network Layer of the OSI protocol stack. It is a means of sharing network resources such as the bandwidth in wireless networks, in order to decrease the amount of transactions needed to convey information between sources and destinations and hence, increasing the flow transfer rate.

Data from separate flows are mixed or diffused together at intermediate nodes and forwarded as a single transmission. The key to success of this tactic lies in the designated node's ability to decode this encoded data upon receipt, in order to restore the original intended message, and is true for the case of multiple unicast or multicast sessions present in a network [11], [12], [13], [14].

The benefit of network coding in wireless mesh networks is evident through the findings of Katti et. al. [13] with the introduction of COPE, a routing-coding technique which entails that nodes logically combine (XOR) packets on the principle that the maximum amount of neighbouring nodes can decode packets intended for them in a single transmission. The shortcoming of COPE is that additional resources must be assigned to each node in the network, such as additional memory for the over-heard packet buffer, additional processing to determine the packet combination as well as added power consumption due to this calculations.

A more basic method of network coding is therefore considered for the model presented in this paper, namely network coding for multiple unicast sessions in the wireless environment [9], [13]. Consider the case where two nodes, A and B exchange data through an intermediate node C (A and B are out of range from another, but both nodes are within range from C). A coding opportunity exists at intermediate node C to mix the messages received from A and B, and broadcast a single compounded message. Because both A and B are within reception range from C, they both receive this compounded message at the same instance. To retrieve the intended message **b** at node A, it uses the cached message **a** in its memory to resolve message **b**. The same is done at B to resolve intended message **a**. If the coding scheme is maintained for the whole duration where both sessions flow

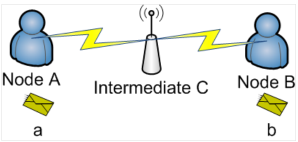


Fig. 1: Wireless multiple unicast example

in opposite directions through a common node, each session can double its flow for the two consecutive links (A to C, C to B and B to C, C to A) because the coded packet size retains the size of one packet, and since two streams are coded together equally (while the data rate remains unchanged at C) the amount of data transferred between A and B in effect doubles.

III. SYSTEM MODEL

In this section the system model for different variants of mesh networks is presented. The goal function of the model is solved with a heuristic optimisation technique called simulated annealing [15], [16]. The results of the model is presented in the following section. The variables used throughout the model is given by Table I.

TABLE I: List of symbols used in the system model

Symbol:	Description:
$G(V, E)$	Network graph based on graph theory notation
V	Set of vertices: the nodes
E	Set of edges: the directed links
H_i	Set of neighbouring nodes for node *i*
ξ	Set of sessions
s_i	Session *i* contained in ξ
S_i	Source node for session *i*
T_i	Sink node for session *i*
λ_i	Flow of session *i*
$f_{(i,j)}$	Flow into node *j* from node *i*
C	Channel Capacity
TP	Aggregated (system) throughput

The convention of flow direction in this model follows a receiver-based approach, meaning that a flow value assigned to a node is positive when the direction of flow is into a node, and negative when flow is exiting the node.

A. System Objective - the goal function

The goal of the model is to maximise the aggregated system throughput of the network, subjected to several constraints explained further in this text. The aggregated throughput is calculated as the sum of all concurrent session flows at one point in time (where time is not the variable parameter). The goal function is stated as:

$$TP = \max \sum_{k=1 \ldots i} \lambda_k \tag{1}$$

$\forall \quad i \quad \text{in } \xi$.

B. Constraint functions

In order to confine a search space to find solutions for Equation 1, the following constraints are used:

- **Conservation of flow constraint:** Conservation of flow [17] dictates that the accumulated information entering a node must equal the accumulated information that exit that node. For every intermediate node on the path of a session, the resultant flow value is zero - the flow entering the node is cancelled out by the flow exiting the node. S_i is the originator of the data transfer for s_i, and will not receive any flow, but flow will leave S_i, so resultant flow for the source will be negative. T_i is the ultimate receiver on the path for s_i and the message is not conveyed any further. There will only be flow entering this node, and no flow leaving giving it a positive resultant flow. If δ_R is the resultant flow at any node, the conservation of flow constraint is given as:

$$\delta_q(s) = \; f_{(i,k)}(s) \; - \; f_{(k,i)}(s) \tag{2}$$

$$\text{where } \delta_q = \begin{cases} \lambda_s & \text{if } q = S_s, \\ \text{-} \; \lambda_s & \text{if } q = T_s, \\ 0 & \text{otherwise.} \end{cases}$$

$\forall \; s \; \in \; \xi \; ; \; \forall \; (i, \; k) \; \in \; E \; ; \; \forall \; i, \; k \; \in \; V$. For a certain session s with flow λ_s, the receiving flow at any node on the path of a single unicast will be 0 for intermediate nodes and λ_s for the sink node T:

$$\text{f}_{(i,k)}(s) = 0 \quad \text{OR} \quad \text{f}_{(i,k)}(s) = \lambda_s$$

For a single path routing algorithm, Equation 2 becomes:

$$\frac{f_{(i,k)}(s)}{\lambda_s} \quad \in \{0 \; 1\} \tag{3}$$

- **Scheduling constraint:** For this constraint the flow on links are considered, and viewed as the flow entering the receiving node. The total flow of every session on the link, whether the receiving node is an intended next hop or just receiving interfering nodes from broadcasting neighbours, cannot surpass the channel capacity C of the link. This is in correspondence with the commonly accepted model described in [18] for the scheduling effect:

$$f_{ij} + \sum_{(k,l)} f_{kl} \; \leq C \tag{4}$$

with $f_{i\,j}$ as the flow rate on link (*i*, *j*) and (*k*, *l*) an arbitrary link interfering with (*i*, *j*). From Equation 4 the scheduling constraint is derived for this model:

$$f_i = \sum_j \lambda_j + \sum_z \lambda_z \tag{5}$$

$\forall \; i \; \in \; \xi$, with j an intended session flow and z an interfering session flow, where $f_i \leq C$.

- **Coding rate constraint:** Fig. 1 shows that two flows, λ_a and λ_b can be mixed together at Node C. From the figure an abstract is made to establish the arrangement needed in order for a coding opportunity to exist: flow λ_a through consecutive links (*j*, *i*) and (*i*, *k*) and flow λ_b through consecutive links (*k*, *i*) and (*i*, *j*). λ_a and λ_b is equally combined at node *i* to form a single combined flow, with a rate equal to both the single contributors individually. To achieve this, the two rates λ_a and λ_b must be equal when they enter node *i*.

$$\begin{aligned} \lambda_a &= 0.5 \cdot \lambda_{ab} \\ &= \lambda_b \end{aligned} \tag{6}$$

where λ_{ab} is the combined flow from λ_a and λ_b.

To solve Equation 1 subject to Equations 3, 5 and 6 for all positive Real solutions for λ_i, it is obvious that the search space for the optimisation problem is fairly large and exponentially increases as the channel capacity used in the model increase. This leads to a case where solving $\lambda_i \, \forall \, i$ is computational extensive. However, the answer for Equation 1 can be easily verified if all λ_i is known. Therefore, this problem is typified as NP-hard. Kirkpatrick et. al., [15] and Cerny [16] independently developed the optimisation technique known as *simulated annealing* which offers a rapid method of approximating global solutions for this type of problems without fixating at a local optimum point. The method uses the analogy of a metallurgical cooling process called *annealing*, used to minimise defects in the materials. Excited atoms move through different energy levels as they are slowly cooled down until they settle in the global optimum energy level. In this analogy, the atoms are current values for λ_i and the energy levels are different configurations of λ_i to find the maximum TP in Equation 1. The symbols used in the optimisation process are given by Table II. Some settings for these parameters:

TABLE II: Defining the basic elements used for Simulated Annealing in this model

	SA element:	Defined for this model:
1.	Search space	$0 \le \beta \le C$ $\beta \in \Re$
2.	The configuration: a finite set of candidate values for λ_i	Ψ
3.	Neighbouring configuration	Ψ' where $\Psi' \subset \Psi - \{i\}$ $\forall \, i \in \Psi$
4.	Aggregated throughput at configuration c	$TP(c)$
5.	Acceptance probability function	$p = e^{\frac{TP(\Psi')-TP(\Psi)}{\tau}}$
6.	Control parameter (*temperature*)	τ
7.	Update factor for τ	θ
8.	Initial configuration (*initial state*)	$\Psi(0) \in \Psi$
9.	Starting temperature	$\tau(0)$
10.	Terminating minimum temperature	T_{min}

- θ is a multiply-decrease update factor for τ, and empirically valued between 0.85 and 1. For this model the value $\theta = 0.87$ is used.
- The initial temperature is selected as $\tau(0) = 10C$, where C is the channel capacity of each link.
- The terminating or minimum temperature value is set empirically to $T_{min} = 10^{-3}$.

The optimisation and solution is given by Algorithms 1 through 3.

Algorithm 1 Calculating the initial configuration $\Psi(0)$

```
loop $\forall \, s \in \xi$
    Calculate paths from $S_s$ to $T_s$
end loop
loop $\forall$ receiving nodes $\in V$
    Find all coding opportunities and update the Flow Structure
end loop
repeat
    Increase flow with $\triangle \, \forall \, i \in \lambda_i$
until Links are saturated $\forall \, e \in E$
```

Algorithm 2 Selecting neighbouring configuration Ψ'

```
repeat
    loop $\forall \, \lambda_i \in \Psi$
        Select random values for each flow from $\beta$
    end loop
until Constraints are met.
```

Algorithm 3 Simulated Annealing algorithm for this system model

```
repeat
    Generate neighbour configuration $\Psi'$ from algorithm 2
    if $TP(\Psi') > TP(\Psi)$ then
        $\Psi = \Psi'$
    else if $p(\Psi') > random[0, 1)$ then
        $\Psi = \Psi'$
    endif
    $\tau = \theta\tau$
until $\tau < T_{min}$
```

C. Modelling environment and experiment set-up

The model implementation is as follow:

- **Network constructions and calculations:** Four types of model inputs are generated in MATLAB:
    1) *The networks*: presented as node-to-node adjacency matrices;
    2) *Paths*: Matrices that represent the path of each session flow over the links in the network;
    3) *Interfering nodes*: Matrices that represent the neighbouring nodes that interfere each session on each link of the path; and
    4) *Capacities*: The calculated capacity for each link.
- **Model variables:** In order to calculate the goal function, see Equation 1, the values of concurrent session flows are to be calculated. This is achieved by creating a single vector of flows and is solved by the array of constraint equations accompanied by the matrices developed in MATLAB.
- **Solving the model:** The simulated annealing algorithm is implemented in C++, and Equation 1 is solved for the following cases:
    1) Pure mesh networks: All nodes have an equal chance to be randomly selected as a source or a destination.
    2) Hybrid mesh networks: One fifth of the nodes are pre-selected as the router backbone, and are excluded from source and destination node selection. All the nodes in the backbone are connected resulting in shorter paths for the sessions over the backbone.

   3) Access mesh networks: One fifth of the nodes are pre-selected as access points and are reserved as priority destinations. The remaining nodes are randomly selected as sources.

- **Experiment scenarios:** The experiment is set-up to evaluate the following parameters:
  1) Network load: The number of concurrent sessions are increased to simulate an increase in load through the network.
  2) Network size: The number of nodes are increased to simulate scaling to larger networks.
  3) Topology: The experiment was batched for three different topology arrangements for each type of network, and run five times for different session paths through the networks.

Figure 2 shows an example of a typical mesh network layout used during the experiment:

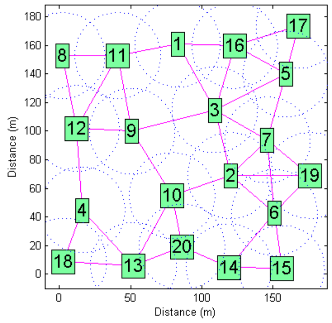


Fig. 2: Example layout for mesh network

IV. RESULTS

To evaluate the effect of network coding, results are compared for two cases: with the implementation of network coding and without the implementation of network coding. Both sets of results were conducted on exactly the same network configurations with the same parameters for each run. The modelled results are presented by Figures 3, 4 and 5. In all cases the top graph resembles the case where network coding was implemented whilst the bottom graph is for plain networks.

From Figures 3, 4 and 5 the following observations are visible:

- Across the board there is an increase in aggregated throughput between the case where no network coding is implemented and when network coding is implemented.
- The amount of increase is different for each type of network (pure, hybrid and access mesh networks).
- With regards to plain networks (no coding implemented): It would appear as if the networks have a limited increase in throughput from small network toward larger networks. The throughput seemingly reaches a roof capacity and tend to level out and remain relatively constant. However, if the throughput is divided amongst the different individual sessions it means that the throughput for each session has dramatically decreased as the load increased to sustain this constant level of total network throughput. This is not very appealing, since the network has slowed down heavily from the user's perspective.

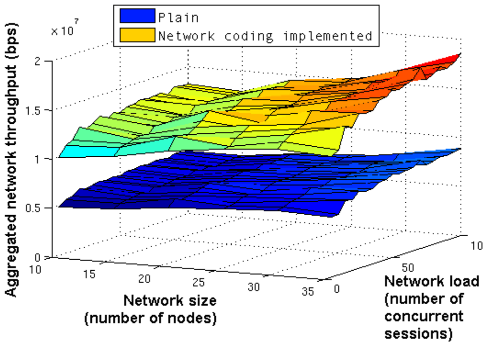


Fig. 3: The aggregated network TP for several pure WMNs

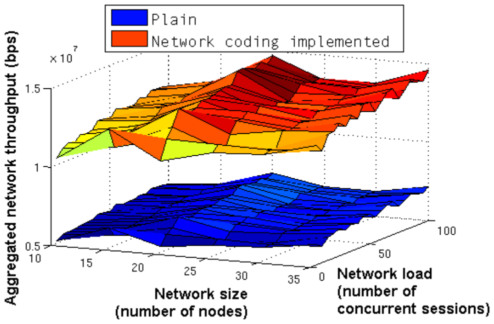


Fig. 4: The aggregated network TP for several hybrid WMNs

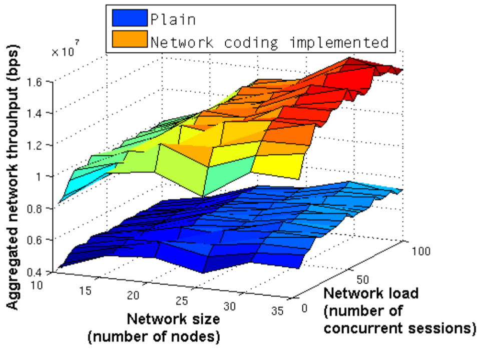


Fig. 5: The aggregated network TP for several access WMNs

- With regards to networks implementing network coding: With the exception of Hybrid mesh networks, it appears that the network throughput for Pure and Access mesh networks sustain to increase as the network load and size increase. For Hybrid networks the slope of throughput increase is fairly dim, however, the overall network throughput in networks implementing network coding is still higher than for the case of plain networks.

Possible explanations for the behaviour in these observations are:

- Pure mesh networks: The amount of coding opportunities is directly related to both the size and load of the network. Since every node has an equal chance to become either a source or destination for the concurrent sessions, the load of the information conveyance through the network is flatly dispersed over the entire network. As the number of sessions increases, the chance of coding opportunities in nodes where opposite direction of session flows occur increases. In tandem with this occurrence, as the number of nodes increases, the chance of a node to become central to opposite direction of session flows also increases, giving a rise in coding opportunities.
- Hybrid mesh networks: The router backbone present in this network type leads to a situation where popular routes between sources and sinks develop around and through the backbone. Most likely all the network load is conveyed through these nodes at some point, resulting in a roof capacity for network coding opportunities. However, since coding opportunities is present at these points, the overall network throughput is evidently higher for network implementing the coding scheme as compared to plain networks.
- Access mesh networks: As with the hybrid mesh networks, popular routes develop towards the AP nodes in this network type. However, since the APs are only destinations and not intermediate nodes, there appear to still be more coding opportunities available for network load conveyance en route to the APs when compared to hybrid mesh networks. Due to the fact that the concurrent session flows tend to be directed towards fixed APs, the amount of flow in the opposite direction decreases, which leads to fewer coding opportunities than in pure mesh networks. It would seem that the effect of network coding scale with the network size.

An important factor to bear in mind is that the occurrence of a network opportunity at one link on the path does not necessarily means that the flow rate of the particular session will increase. It is still limited to the data rate of the "slowest" link on the path, from the conservation of flow constraint in Eq. 2. The total flow on each link is also limited to channel capacity divided by al flows on that link, see Eq. 5.

V. CONCLUSION

There is an overall improvement in network throughput if network coding is implemented. This result is significant because the the network performance is increased without tampering with the routing algorithm. With regards to *scalability*, the results from Section IV has shown that by implementing a network coding scheme the aggregated network throughput improves two-fold:

- An overall improvement when the network load is increased.
- An overall improvement when the network size is increased.

Adding network coding to a system holds a definitive benefit to all networks examined.

ACKNOWLEDGEMENT

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

REFERENCES

[1] E. Hossain and K. K. Leung, *Wireless Mesh Networks: architectures and protocols*. Springer, 2007.

[2] I. F. Akyldiz and X. Wang, *Wireless mesh networks*. Wiley, 2009.

[3] S. Methley, *Essentials of Wireless Mesh Networking*. Cambridge University Press, 2009.

[4] D. S. L. M. A. M. David Johnson, Karel Matthee and H. Kotze, *Building a Rural Wireless Mesh Network*, second edition ed., Wireless Africa, Meraka Institute, South Africa, August 2007.

[5] P.-Y. K. David Tung Chong Wong and Y.-C. Liang, *Wireless Broadband Networks*. Wiley, 2009.

[6] A. B. P. C. Bruce Alan Fette, Roberto Aiello and D. M. Dobkin, *RF and wireless technologies*. Newnes, Elsevier, 2008.

[7] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks." *Ad Hoc Networks*, vol. 2, pp. 1–22, 2004.

[8] I. W. Sudip Misra and S. C. Misra, Eds., *Guide to Wireless Ad Hoc Networks*. Springer, 2009.

[9] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, and S. Gheorghiu, "Multipath code casting for wireless mesh networks," in *Proceedings of 2007 ACM CoNEXT Conference - 3rd International Conference on Emerging Networking EXperiments and Technologies, CoNEXT*. New York, NY, United states: Association for Computing Machinery, 2007, pp. Cisco; Thomson; Alcatel–Lucent; IBM; IMDEA–.

[10] J. Jun and M. L. Sichitiu, "Mrp: Wireless mesh networks routing protocol," *Computer Communications*, vol. 31, no. 7, pp. 1413–1435, 2008.

[11] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[12] R. W. Yeung, *Information theory and network coding*. Springer, 2008.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *Computer Communication Review*, vol. 36, no. 4, pp. 243–54, Oct. 2006.

[14] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science. New Series 200*, vol. 4598, p. 671680, 1983.

[16] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm." *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[17] R. Yeung, S. Li, N. Cai, and Z. Zhang, "Network coding theory, foundation and trends," *Communications and Information Theory*, vol. 2(3,4), pp. 241–381, 2005.

[18] V. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "Algorithmic aspects of capacity in wireless networks," in *Performance Evaluation Review*, vol. 33, no. 1. Banff, AB, Canada: Association for Computing Machinery, 2005, pp. 133–144.

Corna van der Merwe is a post graduate TeleNet Telkom Centre of Excellence student at the North-West University, Potchefstroom campus in Potchefstroom, South Africa. She obtained a bachelor degree in Electronic and Computer Engineering in 2008. Her research interests include network coding, routing methods and wireless networks.