Author:
Dilmore, Michael J

Title:
Towards Dynamic, SDN-assisted Interface Bonding for Heterogeneous 802.11 Devices

Towards Dynamic, SDN-assisted Interface Bonding for Heterogeneous 802.11 Devices

Efficient decision-making algorithms and performance metrics

By

MICHAEL JOHN DILMORE

Department of Engineering Mathematics
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

JANUARY 2021

Word count: 77,748

Abstract

While state of the art mobile devices offer multiple options for wireless connectivity, there are still technical challenges, open research questions, and organisational barriers with respect to the application of interface bonding techniques in real-world networks such as 802.11. Several works have already demonstrated the benefit of leveraging Software Defined Networking towards the interface bonding problem. By centralising bond functions such as link monitoring and interface selection, the SDN controller is able to exploit its global network view and dynamically update the bond configuration in responses to changes in link state. These works are so far limited to small-scale, hardware-based proof of concepts, and a rigorous analysis of the performance of such approaches has not been provided. Towards this research area, the work presented investigates robust dynamic interface selection and related performance metrics for heterogeneous 802.11 device types using Software Defined Networking. Using a custom 802.11 interface bonding framework implemented in MatLab, the performance of the traditional static approach to bond configuration is evaluated for dual-11ac and heterogeneous 11ac-11ah, point-to-point interface bonds at Layer 2, under a wide range of applied load scenarios, with the aim of boosting access speeds over the highly contested wireless last-hop. The static bond results are then used as a benchmark against which to independently evaluate the performance of the main thesis reference work: the SDN architecture and decision-making framework for dynamic MPTCP link selection for dual-802.11 bonds by the authors Nam et al. The simulation results show that although a substantial performance improvement was seen from the use of the dynamic approach, the algorithm was unable to protect at all times against the counter-productive scenario where the bond throughput is reduced below that of the single fastest 802.11 link. To address the limitations of the contemporary work, a novel, predicted-throughput-based algorithm for dynamic selection of heterogeneous 802.11 devices is proposed. The algorithm was found to increase the achieved mean link bandwidth utilisation by 6.16 % and the mean bond throughput by 0.79%, while guarding against the aforementioned scenario and matching the Nam algorithm performance at all other times.

Dedication and acknowledgements

Thank you to my wife, Alice, and daughter, April. I look forward to making up for lost time. I would also like to thank my PhD supervisors, Professor Angela Doufexi and Dr George Oikonomou, for their continued support and encouragement, and all the staff involved at the Centre for Doctoral Training in Communications at the University of Bristol.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

In order to exploit the multiple options for wireless connectivity provided by state of the art mobile devices, the work presented in this thesis investigates robust dynamic interface selection and related performance metrics for heterogeneous 802.11 device types using Software Defined Networking. Several recent works have already demonstrated the benefit of the SDN approach to interface bonding: by centralising bond functions such as link monitoring and interface selection, the SDN controller may exploit its global view and dynamically update the bond configuration in response to changes in link state. However, such works are still limited to small-scale, hardware-based proof of concepts, and a rigorous analysis of the performance of the approach has not yet been provided in the relevant research literature. Therefore, towards the efficient exploitation of device multi-connectivity, the presented work implements a custom 802.11 interface bonding framework in MatLab, which is used to evaluate the performance of the traditional static approach to link selection for dual-11ac and heterogeneous 11ac-11ah, point-to-point interface bonds at Layer 2 of the protocol stack, with the goal of boosting access over the highly contested wireless last-hop. The results are then used as a benchmark against which to evaluate the performance of the main thesis reference work: the SDN architecture and decision-making framework for dynamic MPTCP link selection for dual-802.11 bonds by the authors Nam et al. The test results show that while a significant throughput improvement was seen, the algorithm was unable to protect against the counter-productive scenario where the bond throughput is reduced below that of the single fastest 802.11 link. To address this, a novel, predicted-throughput-based algorithm for dynamic selection is proposed. The algorithm was found to increase the achieved mean link bandwidth utilisation by 0.79 % and the mean bond throughput by 6.16%, while providing specific protection against the aforementioned counter productive scenario, and at least matching the Nam algorithm performance at other times.

1.1 Thesis Research Overview

The research work presented in this thesis investigates efficient decision-making algorithms and related performance metrics for the dynamic, policy-driven, and SDN-assisted control of 802.11 interface bonding, with a focus on boosting wireless access over the last hop by aggregating traditional free-to-use resources such as the 2.4/5 GHz Industrial, Scientific, and Medical (ISM) bands together with new unlicensed opportunities such as sub-1 GHz, deregulated Television White Space (TVWS), and 60 GHz milli-metre wave.

1.1.1 The Proliferation of Wireless

Since the worldwide adoption of the smart phone, wireless data communications has now become an essential enabling technology for the modern way of life. During the current COVID-19 crisis the significance of wireless, as well as our reliance upon it, has increased considerably. Today it is leveraged to keep global economies moving under lockdown restrictions by connecting millions of remote workers to their customers, their corporate networks and resources, and to the Internet. Devices such as smart phones and tablets help connect distanced family members and friends, and also provides a vital life line for self isolating and/or vulnerable individuals. Mobile applications are used to track and trace potential carriers and victims, and contactless payments using NFC help limit the physical transmission of the virus.

However, the adoption of wireless world-wide is still very much an ongoing process. According to the recent Cisco Annual Internet Report [40], the number of Wi-Fi hot-spots is expected to grow from 169 million to 628 million between 2018 and 2023, with new 6th Generation Wi-Fi hot-spots making up 11%. By the same year there will be an estimated 29.3 billion IP networked devices. Approximately 27% of these will be smart-phones and more than half will be Internet of Things (IoT) or Machine-To-Machine (M2M) devices.

1.1.2 The IEEE 802.11 WLAN Standard

The 802.11 WLAN specification [7], better known by the brand name Wi-Fi, is one of the most widely used and recognisable wireless technologies. Since its first publication in 1997, the standard has become the globally predominant WLAN technology. Originally intended to connect hard-to-reach desktops and laptops in enterprise, campus, and residential networks, the killer application was Internet access. Wi-Fi rode the back of the rapid global uptake of smart phones and tablets, and has more recently found use in smart speakers, watches, monitors, as well as various forms of home surveillance and automation. For many the Wi-Fi brand name is synonymous with wireless data communications, and is usually the first to receive blame when the perceived Quality of Experience (QoE) becomes less than acceptable.

1.1.3 Wi-Fi Diversification

The 802.11 suite of protocols has since been developed to encompass a wide range of different frequency bands, transmission ranges, PHY data rates, and target network applications. In the last decade the traditional 2.4 GHz and 5 GHz Wi-Fi variants have been extended into new frequency bands such as the 60 GHz milli-metre wave range (802.11ad [11]) for high-capacity, short-range transmissions, the sub 1 GHz range (802.11ah [13]) targeting low power and long range for the Internet of Things and wireless sensor networks, and the 6 GHz ITS band (802.11p [10]) supporting smart vehicles. More information on the 802.11 standard and the different protocol versions can be found in the background information (Section 2.2.1) of Chapter 2.

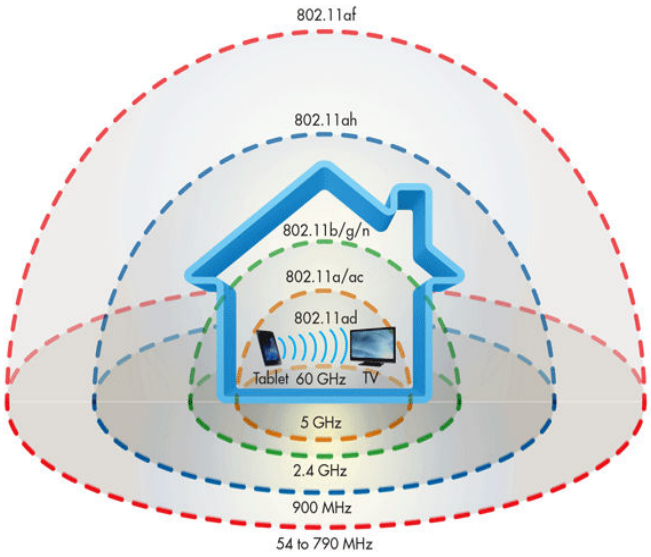

Figure 1.1: Heterogeneous 802.11 Technologies [49]

1.1.4 Increased Multi-Connectivity

The proliferation and diversification of wireless technology has increased the number of connectivity options available to state of the art consumer devices. Take for example the latest 5G-enabled iPhone 12, which still provides backwards support for 4G LTE and as well as legacy 3G CDMA/UMTS and 2G GSM/EDGE services. In terms of WLAN connectivity, there is support for the new 802.11ax high efficiency standard and legacy compatibility all the way back to the OFDM and HR-DSSS PHY types defined in 802.11a and 802.11b respectively. The available WPAN connectivity options include Bluetooth 2.0 and Near Field Communications (NFC). The different wireless connectivity options for the iPhone 12 and several other popular state of the art smart phones are summarised in Table 1.1 below.

	iPhone 12 Pro Max	**Galaxy S20 FE**	**Nokia 8.3**	**Huawei P30 Pro**
5G NR	✓	✓	✓	✓
FDD-LTE	✓	✓	✓	✓
TD-LTE	✓	✓	✓	✓
CDMA EV-DO	✓	✓	✓	✓
UTMS/HSPA+	✓	✓	✓	✓
GSM/EDGE	✓	✓	✓	✓
802.11	✓	✓	✓	✓
Bluetooth 5.0	✓	✓	✓	✓
NFC	✓	✓	✓	✓
GPS	✓	✓	✓	✓

Table 1.1: Wireless Connectivity Options for State of Art Mobile Devices

1.1.5 Exploiting Redundant Links

A number of established methods exist for exploiting the multi-connectivity of wired network devices such as switches and routers, which may be applied to wireless technologies such as 802.11 with varying degrees of success. For example, a common use case for interface bonding is to improve the reliability and fail-over response of network switches, routers, servers, and VNFs, in response to a node, server, or link failure. In addition, using kernel-based solutions such as Linux bonding a mobile user is able to boost link capacity/speed over a highly congested wireless last hop or end-to-end path.[1]

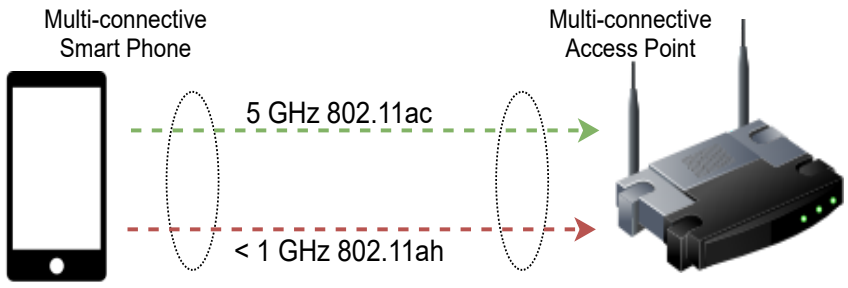

Figure 1.2: Boosting Wireless Access using Point-to-Point 802.11 Interface Bonding

Such solutions operate on top of individual NICs such as the dual Wi-Fi USB dongles illustrated in Figure 1.6, using a packet-based distribution of data frames and without requiring specialist driver interfaces such as *ethool*, and provide a flexible, technology agnostic alternative to PHY layer techniques such as 802.11n Channel Bonding and LTE Carrier Aggregation. A notable practical interface bonding use case was when Apple added support for MPTCP in iOS version 7, to improve the responsiveness of its Siri voice recognition and maps software [42].

Figure 1.3: Edimax Wi-Fi USB Dongles

[1]This use case is the main focus of the thesis research work.

Link aggregation can also be used to facilitate network load balancing in order to increase the overall throughput and capacity. For example, as shown in Figure 1.4 below, a top of rack switch may balance the traffic load across multiple redundant data centre gateway devices, usually by means of a hash of source/destination IP and MAC addresses that assigns a particular output interface to a given communication session between two end-points.

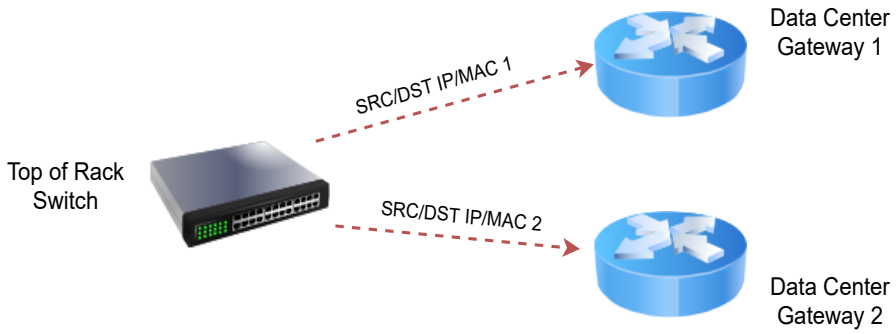

Figure 1.4: A Top of Rack Switch Load Balancing over Redundant DC Gateways

Despite the existence of such techniques, there are still many practical difficulties in exploiting the available multi-connectivity. These are mainly related to factors such as the dynamic nature of the wireless channel and heterogeneous link technologies and performance, as shown by the test results in Chapter 3. However, as mentioned in the SDN background section of Chapter 4, there are also barriers that exist with respect to device interoperability and coordination between networks belonging to different providers or operators.

1.1.6 Dynamic 802.11 Interface Bonding

Current interface bonding solutions such as Linux and MPTCP are mostly statically configured. Therefore, to address the associated performance limitations due to the inherent slow-down effect and the inability to adapt in response to changes in link state, a large number of *dynamic* interface bonding techniques are proposed in the literature with approaches at all layers of the protocol stack, including the Link, Transport, and Application layers.

Dynamic interface bonding implements a continuous feedback loop between the individual enslaved bond links, the link monitoring unit, bond scheduler, and interface selector. The current link state information, such as the individual link delays and throughputs, are periodically inspected by the link monitoring unit and passed to the scheduler and interface selector components, which then updates the bond configuration.

Such solutions have been shown to greatly improve the overall bond throughput when compared to the traditional statically defined approaches, but have yet to be significantly adopted. For example in [180], the authors So et al. define a dynamic scheduling method for point-to-point interface bonding at Layer 2 using dual 802.11 links, but at the current time of writing, the New Load Balancing module proposed has yet to be integrated within the Linux kernel.

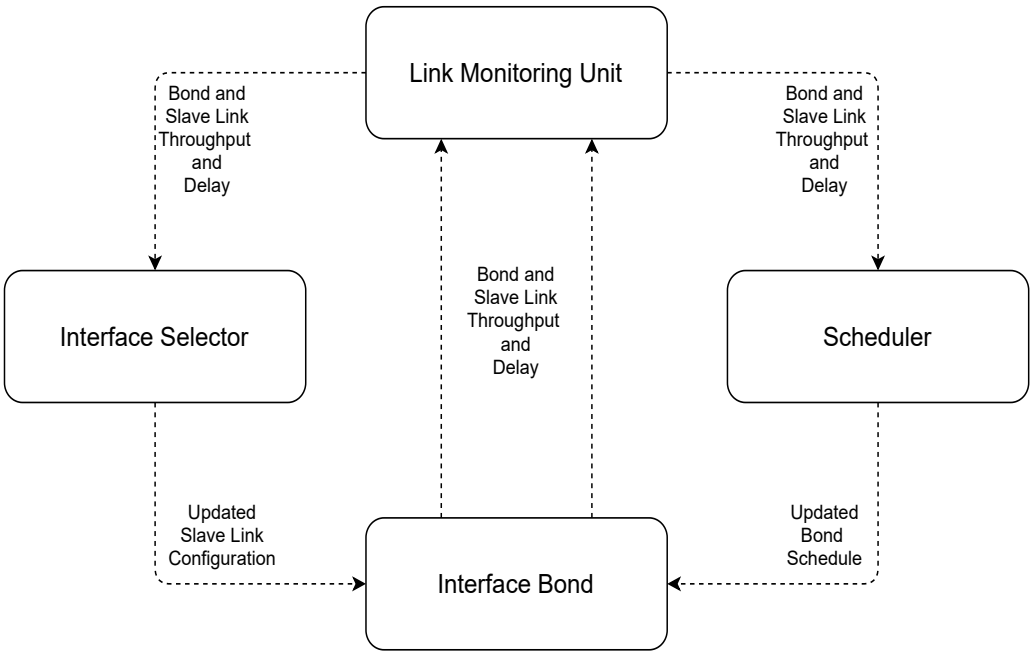

Figure 1.5: Feedback Loop used in Dynamic Interface Bonding Systems

1.1.7 Software Defined Networking

The emergence of Software Defined Networking (SDN) had a profound effect on the organisation and operation of wired networks such as data centres and the Enterprise WAN. SDN decouples the control plane from the data plane devices and moves the network intelligence inside a logically centralised controller. The paradigm was developed in high performance data centres in order to speed up the provisioning and security policy configuration of virtual machines across clustered applications and server farms, but has also found application in many other network types as the Enterprise WAN, the Internet of Things, as well as wireless cellular and WLAN networks such as 802.11, which use a vastly expanded set of control and data plane functions.

Central to the architecture is the SDN controller, which represents the displaced control plane element and performs such centralised functions as routing, addressing, tenant isolation, and finely grained security policies. The controller is able to inspect the state of SDN-enabled network devices using a southbound interface such as the OpenFlow and ForCES protocols. A separate northbound interfaces, for example using REST or SOAP, is presented to SDN applications running in the management plane, that allow network administrators to define the desired configuration. The controller translates the abstractions used (for example or L3 domain or router instance) and then programs the desired low-level detectives into the devices in the data plane via the southbound interfaces.

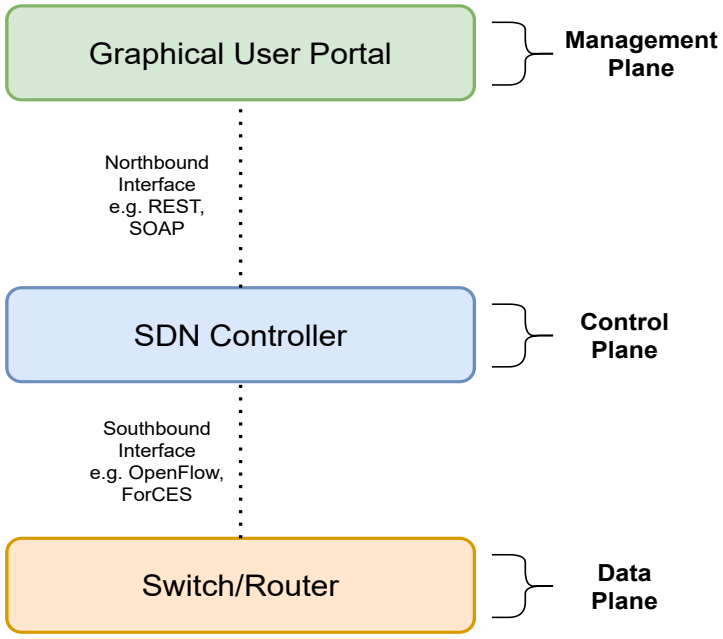

Figure 1.6: SDN Architect and Control Plane Decoupling

SDN therefore represents a shift towards a highly programmable and automated network infrastructure, in contrast to the traditional approach where both control and data planes are vertically integrated within a single network device. The benefits include simplified network administration, increased efficiency, improved network security, reduced operational expenditure, and faster application development and experimentation. However, the centralised control plane introduces a new set of technical challenges, for example, the control and data plane scalability and the additional latency incurred between the controller and data plane devices. In terms of security, the logically centralised controller also presents a single attack surface and point of failure for the entire network. A more detailed overview of SDN, including its application to wireless networks, is found in the background information section of Chapter 4.

1.1.8 SDN-assisted Interface Bonding

An interesting application of SDN is its use in dynamic interface bonding architectures.[2] In such systems, the SDN controller assumes responsibility for bond functions such as link monitoring and link selection which are performed centrally using its globalised network view, normally using a custom southbound interface or a modified version of the OpenFlow protocol. In recent years multiple independent research works have shown the benefit of the SDN-based approach to interface bonding using small-scale, hardware-based experimental test-beds. For example Zannettou et al. in [225] explore path diversity using MPTCP-aware SDN, Rocha et al. proposed an architecture for dynamic interface bonding at Layer 2 using Open vSwitch in [157], and Xu et al. propose SDN/NFV based multi-path architecture called MP-SDWN, which provides support for seamless handover and flow-level QoS.

The most advanced solution appearing in the published research literature is the work by the authors Nam et al. in [130][3], which presents an architecture, decision-making framework, and related performance metrics for dynamic link selection for MPTCP using heterogeneous 802.11 devices over a multi-hop, end-to-end interface bond, as illustrated in Figure 1.7 below. In this particular work, the southbound interface is implemented by spoofing the required MPTCP messages such as MP_JOIN, which are sent from the controller to the SDN agent application running on the MPTCP client and server devices, in response to changes in the state of the end-to-end path. More information on SDN-assisted interface bonding, including the main concepts, performance benefits, design challenges, and related research work (with a detailed account of the main reference work by Nam et al), is provided later in Chapter 4 of the thesis.

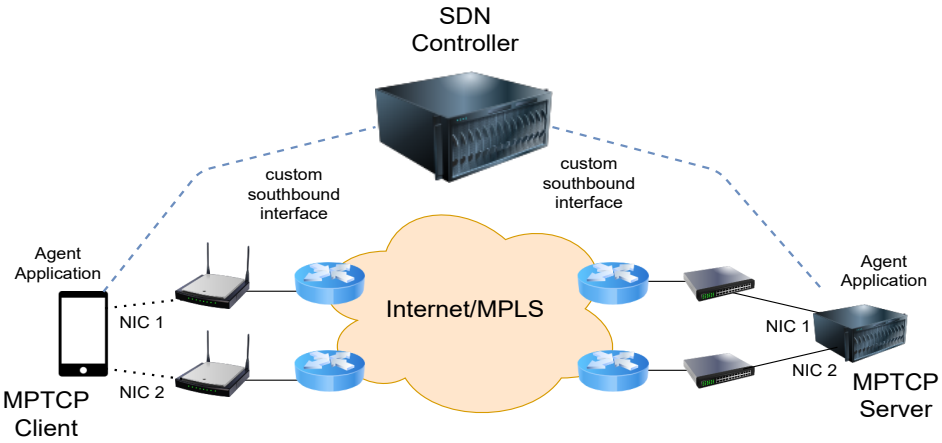

Figure 1.7: SDN-assisted Interface Bonding at the Transport Layer using MPTCP [130]

[2]Note that such SDN-based solutions are the main focus of the thesis work.
[3]The main decision-making algorithms are independently evaluated in Chapter 4

1.1.9 Research Aims and Objectives

In order to efficiently exploit the increased opportunity for link aggregation in current devices and the availability of flexible, kernel-based solutions for boosting wireless access at Layer 2, for example Linux bonding and Open vSwitch, the main objectives of the research work presented in the thesis are given as follows:

1. **Evaluating State of the Art Approaches for Dynamic 802.11 Interface Bonding using Software Defined Networking:** As a flexible method of boosting capacity over the congested wireless last hop, the thesis research aims to evaluate and compare the current proposals for dynamic and SDN-assisted centralised control of link selection at all layers of the stack. Several small-scale, hardware-based test beds have been developed that demonstrate the benefit of software-defined control of interface bonding functions such as scheduling and interface selection. However, there has not yet been a rigorous analysis of the robustness of the main decision-making logic and related performance metrics under of a wide range of applied load conditions. The fulfillment of this analysis and the further exploration and evaluation of novel algorithms for efficient, robust, and dynamic link selection using heterogeneous 802.11 devices, are therefore the main aims of the research work presented in the thesis.

2. **Understanding the Impact of the MAC Layer on the Throughput Performance of Point-to-Point 802.11 Interface Bonds:** When bonding over multiple 802.11 links, severe instability and degradation effects are observed for Transport Layer protocols such as TCP and UDP that are caused by short-term variations in the link access delay and the stochastic CSMA/CA algorithm. In many cases the aggregate bond throughput is reduced below that of the single fastest link, thereby rendering the use of the technique counter productive to the aim of boosting wireless access speeds. The effects on upper layer protocols are observed experimentally and well documented in the literature. However, despite the relevance in the point-to-point Wi-Fi links and bonds used in the relevant publications, there is still a general lack of understanding with respect to the long-term performance of 802.11 interface bonding at the MAC layer and the impact the DCF CSMA/CA mechanism under heavily unbalanced applied loads.

1.1.10 Research Contributions

Towards fulfilling the objectives described in the previous section, the primary contributions of the thesis research work are:

1. **Presentation of a Novel Evaluation Framework for 802.11 Interface Bonding at the MAC Layer Implemented in MatLab:** In order to study the performance of 802.11 interface bonding with a clear focus on the impact of the DCF channel access mechanism, a novel evaluation framework is implemented using the MatLab programming language, and validated using the established Markov-based models by the authors Bianchi and Chatzimisios. The framework provides a flexible method of 802.11 bond evaluation at point-to-point for heterogeneous device types, simply by altering the DCF input parameters used in the simulation as per the relevant IEEE standard.

2. **Rigorous Analysis of the 802.11 Interface Bond Performance using the Traditional Static Link Configuration:** The implemented framework is used to carry out a rigorous analysis of the 802.11 interface bonding throughput performance at the MAC layer when using the traditional static link configuration under a variety of unbalanced applied load scenarios. In particular there is a quantitative and qualitative analysis of the inherent bond throughput slow-down effect for in-order delivery, and a detailed comparison of homogeneous 11ac-11ac and heterogeneous 11ac-11ah interface bond types.

3. **Detailed Performance Evaluation of Contemporary Solution for Dynamic 802.11 Interface by the authors Nam et al:** The developed link aggregation framework is then further extended in Chapter 4 to study the performance and general robustness of the state of the art solution for SDN-assisted 802.11 interface bonding at the Transport layer proposed by the authors Nam et al. in [130]. The throughput-gap-based approach to dynamic link selection is implemented, evaluated, and compared in detail against the traditional static round robin bond configuration with the same rigor and using an identical series of unbalanced applied load scenarios.

4. **Design and Evaluation of a Novel Algorithm for Dynamic 802.11 Link Selection using the Empirical Predicted Bond Throughput:** To address the negative impact of the 802.11 MAC layer on the aggregate bond throughput performance and the related shortcomings of the throughput-gap-based approach, a novel algorithm for robust, dynamic, and centralised bond link selection using heterogeneous 802.11 devices is designed and evaluated using the developed MatLab framework, and the achieved performance compared against the Nam method using the same set of applied loads and other simulation input parameters.

1.1.11 Summary of Main Findings

The main findings from the following research chapters are summarised as follows:

- **Implemented MatLab Framework Validation Testing:** The simulation output was verified for a number of raw PHY data rates and different 802.11 technologies, in particular the now legacy 802.11ac at 2.4/5 GHz and the new low-power, long-range 802.11ah standard in the sub-1 GHz range. The key finding from the preliminary work in Chapter 2 was that the implemented MatLab framework was able to simulate the 802.11 DCF with a reasonable degree of accuracy. The evaluated channel saturation throughput, expected access delay, and conditional collision probability from the output from the established Bianchi and Chatzimisios analyse were all found to fall within the 99.9% confidence intervals from the simulated outputs, therefore verifying the correctness of the implemented simulator with a reasonably high level of certainty.

- **Comparison of 802.11ac vs 802.11ah DCF Performance:** The main finding from the results of the numerical analysis was that even when both 802.11ac and 802.11ah WLANs are configured with a 65 Mbps raw PHY data rate[4], the difference in DCF parameters at the MAC layer between the two Wi-Fi variants leads to a significant difference in the average channel saturation throughput and expected access delay of 23.74% and 67.14% respectively. Notably, this was in contrast to the conditional collision probability, which was found to be identical for both versions of the standard.

- **Static 802.11 Interface Bonding Performance Evaluation:** The main finding from the evaluation of the static 802.11 bond configuration in Chapter 3 was that over all eight tests for the dual-11ac and heterogeneous 11ac-11ah interface bonds, the mean link bandwidth wastage due to the inherent slow-down effect was equal to approximately 18.31% of the total available capacity. The performance of the heterogeneous 11ac-11ah interface bond was found to be significantly lower when compared to the homogeneous 11ac-11ac bond, despite the identical balanced applied load and 65Mbps raw PHY date used. Due to the difference in DCF parameters, for example the slot length and inter-frame, the achieved mean throughput for the heterogeneous bond was just 4.1 Mbps compared to 6.085 Mbps for the homogeneous 11ac-11ac bond, and the mean link bandwidth was also substantially lower at 78.75% compared to 84.62%. Furthermore, in 7 out of the 8 tests at certain combinations of applied loads the impact of the slow-down effect was so severe that the aggregate throughput of the static 802.11 bond was reduced to below that of the single fastest available individual link, therefore rendering the use of the technique counter productive to the aim of boosting wireless access speeds. The condition was not an

[4]MCS=6 for 11ac and MCS=7 for 11ah

uncommon occurrence and was observed in each of the individual applied load tests, with the exception of the balanced case for the 11ac-11ac bond in Test 1A.

- **Dynamic 802.11 Interface Bonding Replication Testing:** Presented in Chapter 4, the main finding from the evaluation of the contemporary approach to dynamic 802.11 link selection using SDN by the authors Nam et al in [130], was that the bond throughput-gap-based decision-making algorithm significantly increased the overall mean 802.11 bond throughput by 26.81%, and also increased the link bandwidth utilisation by 7.28%, when compared to the traditional static round robin approach. However, although the algorithm did provide a substantial benefit, the performance was not robust for all applied load combinations considered in the simulations. In particular, the bond delay gap metric used was not able to predict the condition when the achieved bond throughput is reduced below that of the single fastest link, which reduces the overall mean bond throughput performance as well as the mean link bandwidth utilisation. This was a common scenario and was observed in 7 out of the 8 tests performed, with a much higher number of occurrences for the heterogeneous 11ac-11ah bond when compared to the homogeneous dual-11ac bond.

- **Novel Dynamic 802.11 Interface Bonding Evaluation:** From the evaluation of the novel algorithm for dynamic 802.11 link selection in Chapter 4, the main research finding was that the predicted-throughput-based approach at least marginally outperformed the Nam algorithm in every test with the exception of the balanced load scenario in Test 1A. Taking an average of the simulation results against in all eight individual tests conducted, the overall mean bond throughput was increased by 0.79% and the mean link bandwidth utilisation by 6.16% when compared to the contemporary throughput-gap-based method by Nam. In particular, as per the intended design and in contrast to Nam, the decision-making of the proposed algorithm was robust for *all* applied load combinations considered in the testing. In particular, protection was provided at all times against the damaging scenario where the bond throughput is reduced below that of the fastest available single link whilst otherwise matching the Nam algorithm performance.

1.1.12 Research Output

The thesis research led to the submission of a conference paper titled *"Analysing Interface Bonding in 5G WLANs"*, which was presented at, and published in, the proceedings of the IEEE Computer Aided Modelling and Design (CAMAD) conference, Barcelona, Spain, September 2018 [53].

1.1.13 Thesis Organisation

The thesis research is work presented in the order in which it was carried out, with relevant background information and context given at the start of each chapter. The following chapters of the thesis are structured as follows. All required preliminary literature reviews, analysis, implementation, and simulation testing related to the saturated single-link 802.11 performance is given in Chapter 2. The initial benchmark tests for the static 802.11 bond configuration is presented in Chapter 3 using a custom bonding layer implemented on top of the single-link DCF simulations. The main research work of the thesis towards the efficient, dynamic, and centralised control of bond functions using SDN is presented in Chapter 4. Finally, Chapter 5 contains a conclusion and summary of the thesis work.

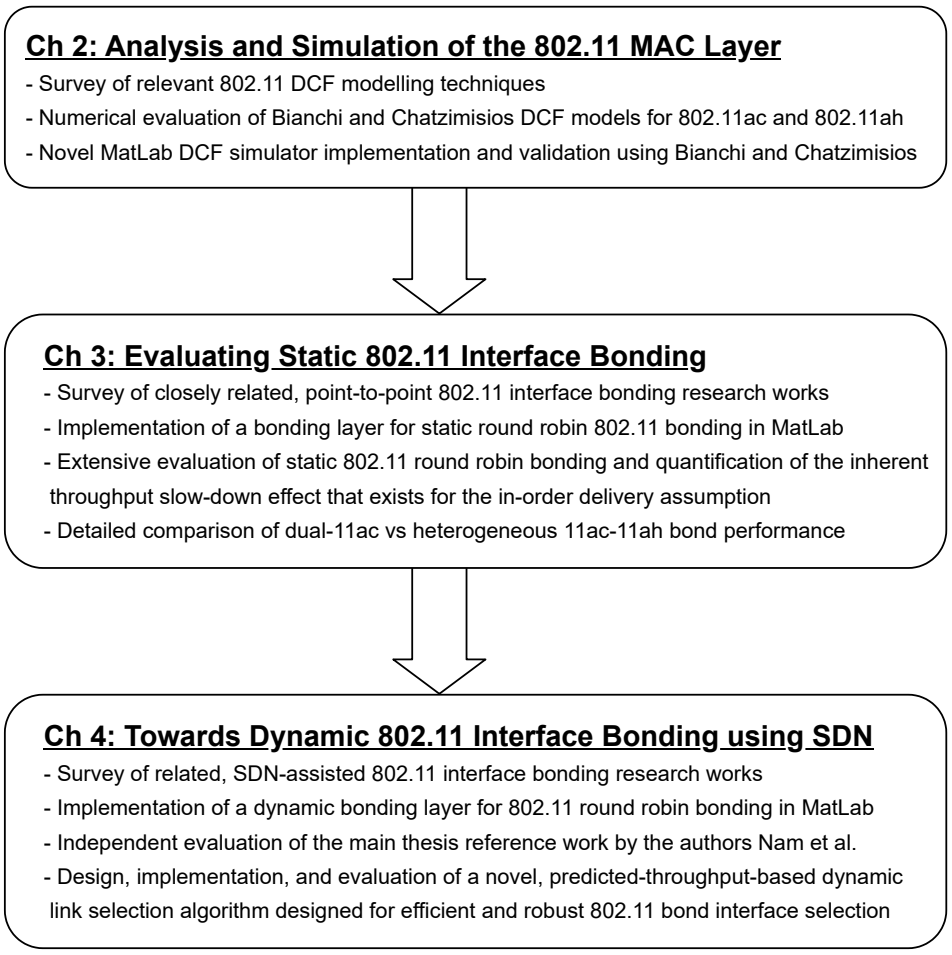

Figure 1.8: Thesis Research Organisation

CHAPTER 2

ANALYSIS AND SIMULATION OF THE 802.11 MAC LAYER

This first research chapter presents all of the initial single-link 802.11 simulations, testing, and modelling performed towards the main thesis research goals. As the globally predominant Wireless LAN technology, the preliminary work in this chapter discusses and analyses the performance of the 802.11 MAC layer channel access mechanism called the Distributed Coordination Function. The 802.11 standard is described in detail with a focus on the MAC sub-layer, and a review of state of the art modelling techniques is provided with example numerical outputs. The main chapter contribution is the design and implementation of a novel MatLab-based DCF evaluation framework, which is successfully validated against the results of the widely accepted analyses by authors Bianchi and Chatzimisios using a 99.9% confidence interval. As part of the initial testing, the performance of the 11ac and 11ah link types used in the interface bonding evaluation is analysed and compared under a range of different applied channel loads.

2.1 Chapter Research Overview

Since the publication of the first 802.11 standard in 1997, wireless connectivity has now become ubiquitous. Designed for home and enterprise Wireless Local Area Networks (WLANs) to connect consumer devices such as laptops and hard-to-reach desktop workstations, the original killer application for Wi-Fi was Internet access. Demand for wireless connectivity increased exponentially during the first decade of the millennium, with the global adoption of devices of smart phones and tablets that also required local connectivity in addition to cellular interfaces. Over the last decade the standard has evolved to accommodate a range of next generation applications such as vehicular networks (802.11p) and the Internet of Things (802.11ah), and has moved out of the

traditional 2.4 GHz and 5 GHz bands and into Television White Space (TVWS) and millimetre wave frequencies.

Until very recently the trend in each new development cycle and version of the standard was for an *ever-faster* PHY, moving from the Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) to the more advanced Orthogonal Frequency Division Multiplexing (OFDM) PHY with increasingly dense modulation. The basic channel access mechanism employed had changed very little in the 20+ years since the original publication, until the current 6th generation 802.11ax High-efficiency Wireless which introduced a partially scheduled approach to channel access called Orthogonal Frequency Division Multiple Access (OFDMA), representing a significant shift from the traditional contention-based Distributed Coordination Function (DCF).

The mathematical modelling of the legacy 802.11 DCF has therefore become a mature research area with several thousand related published works. The seminal analytical work in the field is the Markov-based model by Bianchi which was published in 1999 and plays a key role in the context of the overall thesis research by providing a benchmark against which to verify the correctness of the implemented MatLab simulator. While in some respects the Bianchi model can be now considered as an over simplified or outdated representation of the 802.11 channel access procedure, it is still considered suitable for the research contained in this thesis as many of the traditional DCF-based devices will remain in production for years to come and substantial research interest remains in learning how to aggregate these multiple 802.11 types effectively. In addition, the large body of work surrounding the Bianchi modelling provides a high level of confidence in the implemented MatLab simulation by acting as a reliable and fully substantiated performance benchmark. Note that Bianchi does not model the DCF access delay, and therefore the delay extensions to the core model by Chatzimisios are also heavily relied upon.

Towards the overarching research goals of the thesis research, the following initial work presented in this chapter analyses, evaluates, and simulates the performance of the 802.11 MAC layer as the current, globally predominant WLAN technology. The primary research aims, contributions, and findings of the chapter work are detailed in the following subsections.

2.1.1 Chapter Research Objectives

The aim of the initial work in this chapter is to provide the complete single-link foundation for the dynamic 802.11 interface bonding research later in the thesis. The preliminary work presented includes all relevant literature reviews, analytical tools, simulator functionality, and single-link testing required to inform, instruct, and enable the main thesis research goals. Three separate but related research objectives are identified:

1. **Survey of DCF Analytical Techniques:** The initial task of the chapter work was a literature review of analytical techniques for the modelling of the 802.11 MAC layer, in order to identify and evaluate analyses suitable for the validation of the MatLab-based DCF simulator presented later in Section 2.4.

2. **Single-Link 802.11 Simulation:** In order to facilitate the overarching thesis research, the main goal of the chapter work was the design and implementation of a novel MatLab-based simulator for the 802.11 DCF with performance statistics for an individual wireless STA, as well as the validation of the simulated DCF output using the widely accepted analytical models by the authors Bianchi and Chatzimisios.

3. **802.11ah vs 802.11ac Performance Comparison:** A secondary chapter research goal was to evaluate and compare the performance at the MAC layer of traditional 2.4 GHz 802.11ac against the recent sub-1 GHz 802.11ah standard, which are used together in Chapter 3 and Chapter 4 to evaluate the performance of dynamic and SDN-assisted selection of heterogeneous 802.11 bond links.

2.1.2 Chapter Research Contributions

Towards the fulfillment of the objectives described in the previous section, the main research contributions of the work presented in the current chapter are summarised as follows:

1. **Single-Link 802.11 Simulation in MatLab:** In the context of the main thesis research, the primary contribution of the chapter work is the implementation and verification of a novel MatLab-based simulator for the 802.11 DCF. While not constituting an original research contribution in its own right, the developed simulator provides the basis for the 802.11 interface bonding framework and research in Chapter 3 when evaluating the performance of traditional static link configuration, and is further extended in Chapter 4 to analyse the performance of state of the art dynamic link selection algorithms.

2. **802.11ah vs 802.11ac Performance Comparison:** The secondary contribution of the chapter work is the 11ac and 11ah numerical evaluation, which provides insight into the relative MAC layer performance of the two link types considered under similar levels of applied channel load and identical raw PHY data rates. The quantitative and qualitative analysis provided is essential for a proper understanding of the heterogeneous 802.11 interface bonding evaluation results presented in Chapters 3 and 4.

2.1.3 Summary of Chapter Findings

Within the context of the current chapter research goals, the main findings from the preliminary work presented are summarised as follows:

- **MatLab DCF Evaluation Framework Validation:** From the results of the simulation verification evaluation, the implemented MatLab framework was found to successfully simulate the 802.11 DCF channel with a high degree of accuracy. In particular, the DCF channel saturation throughput, expected access delay, and conditional collision probability from the Bianchi modelling were all found to fall within the 99.9% confidence interval when using a sample size of 8. On average the simulated DCF channel output was within $\pm 1\%$ of the numerical output from the established Bianchi and Chatzimisios analyses.
- **802.11ac vs 802.11ah Performance Comparison:** The secondary findings are from the results of the 11ac vs 11ah MAC layer performance comparison in Section 2.3.3. The numerical analysis conducted showed that although both 802.11 links were configured with the same raw PHY data rate of 65 Mbps (MCS=6 for 11ac and MCS=7 for 11ah), the difference in DCF parameters at the MAC layer between the 11ac and 11ah versions of the standard creates a significant difference in the average channel saturation throughput and expected access delay of 23.74% and 67.14% respectively. As expected the conditional collision probability was found to be identical in both versions of the standard for all applied loads considered.

2.1.4 Chapter Organisation

All the prerequisite background information needed to understand the 802.11 DCF simulator implementation and performance evaluation is presented in Section 2.2 of the chapter. Subsection 2.2.1 provides all necessary information on the 802.11 standard, including a primer of the 802.11 protocol stack, frame format, and related organisations and working groups, as well as an overview of all major versions of the standard from the original 802.11-1997 [2] to the unpublished 802.11be. Furthermore there is a detailed description of the 802.11 MAC layer channel access mechanism known as the Distributed Coordination Function (DCF). Section 2.2.2 contains a review of relevant DCF analytical techniques with a detailed focus on the modelling by Bianchi and Chatzimisios, which is integral to the validation of the implemented simulation framework.

Section 2.3 uses the aforementioned analytical models to numerically evaluate the performance of the 802.11 DCF for a number of different protocol versions. In particular the saturated channel throughput, expected access delay, and conditional collision probability are evaluated and compared for 802.11ac and 802.11ah. These two versions are of specific relevance in the context of the overall thesis research as they are used as the basis for a heterogeneous 802.11 dual-link interface bond in Chapters 3 and 4.

The main chapter research work is found in Section 2.4 which presents the MatLab-based implementation of a novel 802.11 DCF evaluation framework. Implementation details such as the simulation input parameters, the main algorithms and pseudocode, and the DCF related output data are described in Section 2.4.1. The correct operation of the implementation simulation framework is then verified in Section 2.4.2, which gives a direct comparison against the output of the widely accepted analytical modelling by Bianchi and Chatzimisios.

2.2 Chapter Background Information

This chapter section provides the required background information related to the main Link Layer technology used in the interface bonding research, i.e. the IEEE 802.11 standard for Wireless Local Area Networks.

2.2.1 The IEEE 802.11 Standard for Wireless Local Networks

A Wireless Local Area Network (WLAN) is defined by Tanenbaum [194] as a wireless network with a radius of less than 1 km. Typical examples include home, enterprise, and campus networks. The predominant WLAN standard is IEEE 802.11 [2] also known as Wi-Fi which is short for Wireless Fidelity. Since the release of the first specification in 1997 the standard has been widely adopted and is now ubiquitous. Wi-Fi Certified devices can be found in desktops, laptops, smart phones, smart speakers, tablets, video game consoles, smart watches, cars, and even fridges. According to the Wi-Fi Alliance there are currently more than 9 billion Wi-Fi devices in use globally providing an economic value of $2 trillion in 2018 expected grow to $3.5 trillion by the end of 2023 [117]. Cisco [40] report that the number of Wi-Fi hot-spots globally will grow 4-fold by the same year while the number of new 6th generation hot spots is expected to grow 13-fold with average link speeds increasing to 92 Mbps overall.

Recently Wi-Fi has found use in diverse network types and applications such as the Internet of Things (IoT), Machine-To-Machine (M2M) communication, and smart vehicular networks. This led to the publication of several new standards using frequencies outside of the traditional 2.4 GHz and 5 GHz bands. These include 802.11p at the 5.9 GHz Intelligent Transport Systems (ITS) band targeting Vehicular Ad Hoc Networks (VANETs) [10], the 802.11ah standard providing low-power, long-range communications in the sub 1 GHz range for IoT [13], and 802.11af which operates in unused Television White Space (TVWS) frequencies [13]. In doing so Wi-Fi entered into direct competition with several other wireless technologies. For example in wireless sensor networks and IoT there is some overlap between 802.11 and Wireless Personal Area Network (WPAN) standards such as 802.15.4 and 802.15.1. Designed for short range and low-power consumption both 802.15.1 and 802.15.4 also operate at 2.4 GHz and both target sensor applications such as wireless headsets, speakers, games controllers, and video cameras. This low-spec, low-cost approach works well for personal consumer devices but to connect the IoT longer range, lower latency, better scalability, and higher data rates are required. Cellular technologies such as General Packet Radio Service (GPRS), Long Term Evolution (LTE), and WiMAX can fulfil these performance requirements but they also consume more power, require an expensive infrastructure of base stations, and operate mostly in expensively licensed portions of the spectrum. The new 802.11ah standard aims to position itself between precisely within this niche. Traditional Wi-Fi also faces competition from cellular technologies such as LTE Unlicensed (LTE-U) used by operators seeking to reduce the burden on the costly licensed parts of the network and offload

data traffic to free-to-use unlicensed spectrum wherever possible. However unlike 802.11 LTE-U does not have an Listen-Before-Talk (LBT) mechanism which has been shown to adversely impact the performance of adjacent Wi-Fi networks. The co-existence of Wi-Fi and LTE-U in the 5 GHz band has been a major research topic in recent years for example [45, 73, 186] and the subject of a widely publicised debate between the Wi-Fi Alliance and 3rd Generation Partnership Project (3GPP) [33] around fair use of Industrial, Scientific, and Medical (ISM) resources.

In the following pages, the various 802.11 network components and topologies are summarised in Section 2.2.1.1 and the layered architecture of the protocol is illustrated in Section 2.2.1.2. The relevant 802.11 organisations and working groups are detailed in Section 2.2.1.4 and review of the main 802.11 standards is given in Section 2.2.1.5. Particular attention is given to the MAC sub-layer in Section 2.2.1.6 because as shown later in the thesis the individual 802.11 link access delays and throughputs are key contributors to the interface bonding performance and here the channel access mechanism plays a centre role.

2.2.1.1 Components and Topologies

The main components of an 802.11 WLAN are: (1) one or more wireless Stations (STAs), which are in most scenarios end-user devices such as smartphones or laptops, (2) one or more Access Points (APs), central nodes used by STAs to communicate and reach external networks such as the Internet, and (3) the Distribution System (DS), the wired backhaul connecting multiple APs. In a typical 802.11 WLAN a wireless STA connects to an AP in a single-hop star topology called a Basic Service Set (BSS) for which all internal network traffic is routed through the central node. This configuration is known as Infrastructure Mode and also supported is a single-hop, interconnected mesh topology called Ad Hoc mode in which all STAs communicate directly with each other.

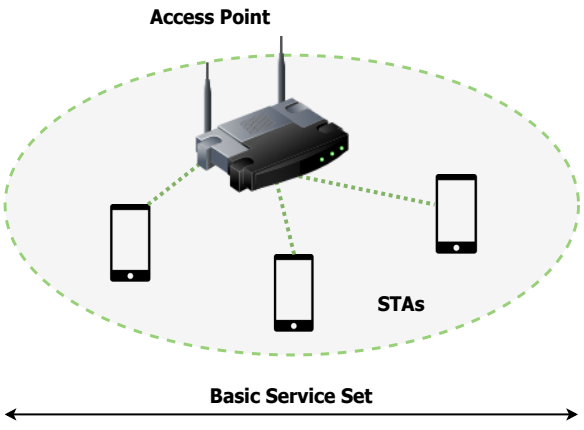

Figure 2.1: IEEE 802.11 Infrastructure Mode Topology.

2.2.1.2 Protocol Stack

The 802.11 standard is split into two layers: (1) the Physical Layer (PHY) and (2) the Medium Access Control (MAC) sub-layer that forms part of the larger Data Link layer (Layer 2). As per [194] the PHY concerned with transmitting raw bits over a physical communication channel, for example the radio frequency channel. The 802.11 PHY is comprised of two independent sub-layers: the Physical Layer Convergence Protocol (PLCP) sub-layer and the Physical Medium Dependent (PMD) sub-layer.

By contrast the role of the Link Layer is to present the raw communication channel as a link which appears free of undetected transmission errors. Like the PHY the Link Layer is also comprised of two independent sub-layers: the MAC and the Logical Link Control (LLC) sub-layer. In addition the 802.11 MAC is required to appear to the LLC as an 802.3 LAN and must therefore also incorporate functions normally dealt with by higher layers such as authentication and association.

Of particular interest for the research contained in this thesis is the 802.11 MAC sub-layer which is responsible for regulating access to the shared physical transmission medium, i.e. the unlicensed 2.4 GHz ISM band in most traditional use cases. Precisely how the channel access is regulated amongst the individual competing wireless stations is described in further detail in Section 2.2.1.6 below.

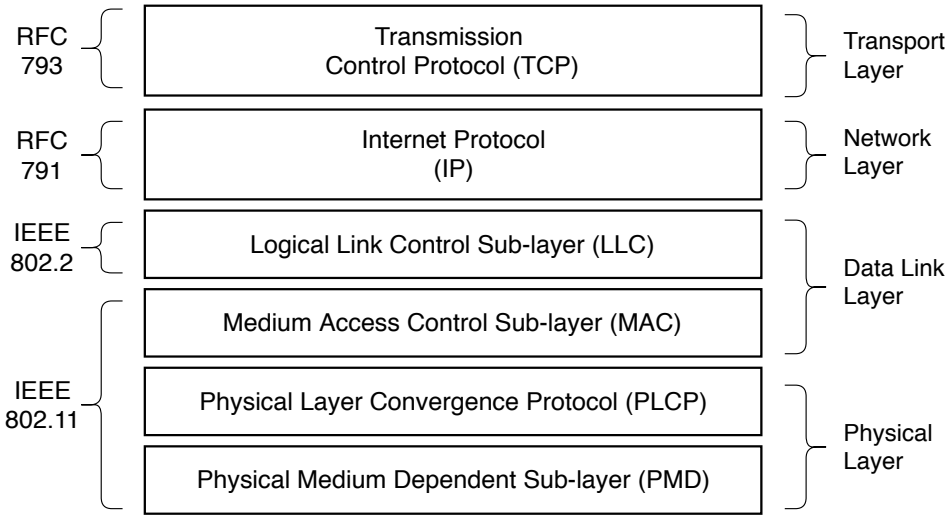

Figure 2.2: The IEEE 802.11 Protocol Stack

2.2.1.3 Frame Format

The 802.11 frame format is shown in Figure 2.3 below. Three frame classes are defined: data, control, and management. The first data frame field is the *Frame Control* field which is comprised of 11 individual sub-fields. The first sub-field is the *Protocol Version*, which is always set to 00. The frame *Type* and *Sub-Type* fields follow and for regular data frame are set to 10 and 0000 respectively in binary. The *To DS* and *From DS* fields indicate whether the frame is ingressing or egressing the network. The *More Fragments* field is used to indicate that further data fragments are to follow. The *Power Management* field when set to 1 indicates that the sender is going into power-saving mode. The *More Data* field indicates that the sender has additional frames to transmit. The *Protected Frame* field is a single bit that indicates that the frame payload has been encrypted. The *Order* is a single bit field used to tell the receiver that the upper layer is expecting a frame sequence to arrive in-order.

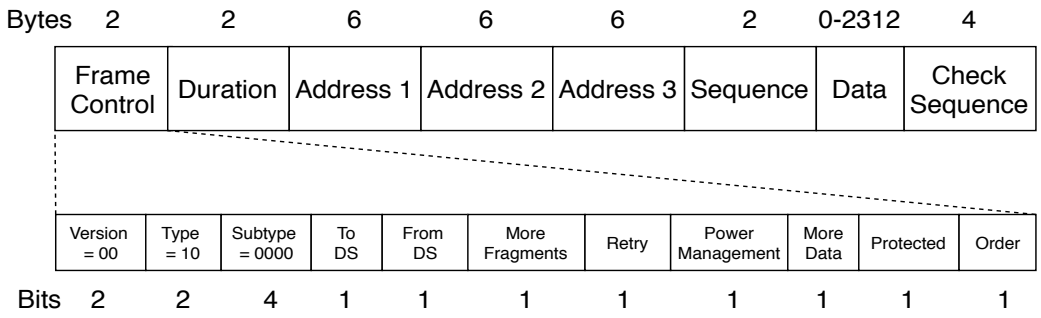

Figure 2.3: IEEE 802.11 Frame Format.

The second field is the frame *Duration* which indicates how long the frame is and how long its acknowledgement will occupy the channel and is central to the Network Allocation Vector (NAV) mechanism described later. The first address field contains the address of the receiver, the second contains the address of the transmitter, and the third provides the address of a relayed destination node. The *Sequence* field provides a reference number for frames being transmitted in order to detect duplicates. The *Data* field contains the frame payload which may be up to 2312 bytes. Finally the *Check Sequence* field contains the result of a 32 cyclic redundancy check of the frame.

2.2.1.4 Related Organisations

The IEEE 802.11 Working Group was established in 1990 with the goal of developing a protocol standard for wireless Ethernet. The Working Group is comprised of multiple committees and for each new version of the standard a separate Study Group is responsible for researching potential amendments and preparing Project Authorization Requests for submission to the IEEE 802 Executive Committee. Once requests are authorised a Task Group is designated by the Working Group as the official author/s of a particular version or amendment. A Topic Interest Group is focused on a particular topic and gathering interested members in order to determine the project technical feasibility and requirements before a formal Study Group is established. There are also various other committees with a determined role but whose function does not involve modifying the IEEE 802.11 standard, for example a Standing Committee or a short-term Ad-Hoc Committee. Note that the 802.11 Working Group is responsible for developing and defining new versions of the standard but that it has no function for testing vendor hardware compliance with the published specifications. Therefore in order to improve vendor interoperability, promote the use of 802.11, and provide a unified certification procedure, the pioneering vendors of 802.11b including Lucent, 3Com, and Nokia, formed the Wireless Ethernet Compatibility Alliance (WECA) and created the Wi-Fi brand name. The WECA was later renamed as the Wi-Fi Alliance in 2002 and has since grown to include over 500 different member companies.

IEEE Standard	**Freq. (GHz)**	**Bandwidth (MHz)**	**Data Rate (Mbps)**	**Description**
802.11-1997	2.4	22	< 2	1st 802.11 standard
11b-1999	2.4	22	< 11	2nd 2.4 GHz standard
11a-1999	5	5/10/20	< 54	1st 5 GHz standard
11g-2003	2.4	5/10/20	< 54	Faster 2.4 GHz Wi-Fi
11i-2004	—	—	—	Security Enhancements
11e-2005	—	—	—	Added Quality of Service
11n-2009	2.4/5	20, 40	< 600	2.4/5 GHz Wi-Fi
11p-2010	5.9	10	< 27	Vehicular Wi-Fi
11s-2011	—	—	—	Wireless backhaul
11ad-2012	60	2160	< 6700	Milli-metre wave Wi-Fi
11ac-2013	5	20, 40, 80, 160	< 1300	Faster 5 GHz Wi-Fi
11af-2014	0.48-0.79	20, 40, 80, 160	< 35.6	Wi-Fi in TVWS
11ah-2017	<1	1, 2, 4, 8, 16	< 347	Wi-Fi for IoT
11ax-2019	1-6	20, 40, 80, 160	< 6970	Current Wi-Fi standard
11be-202X	1-7.125	20, 40, 80, 160, 320	> 10K	Next generation Wi-Fi

Table 2.1: Table of Major IEEE 802.11 Standards

2.2.1.5 802.11 Standards

Since 1997 a great number of specifications have been published by the 802.11 Working Group. Some are fully-fledged, major revisions to the main standard, such as 802.11b and 802.11ac, and some, for example 802.11e and 802.11i, extend 802.11 to include additional features such as Quality of Service (QoS) or improved security. Others make insignificant adjustments or minor corrections to earlier versions. In each major revision of the standard there is usually a significant improvement in performance at the PHY layer, although increasingly more changes are now also starting to appear in the shape of MAC layer efficiency improvements.

2.2.1.5.1 Naming Convention Each new Working Group is assigned a letter of the alphabet and is referred to for example as 802.11b or 802.11g. When letters of the alphabet were exhausted they were recycled by appending an additional latter to the standard name, for example 802.11ac. This approach is cumbersome, not very informative or descriptive, and does not allow differentiation between major revisions and minor amendments. The letter-based convention has network engineers and undergraduates for over two decades since 802.11b famously beat the 802.11a Working Group to publication in September 1999. To increase marketability, the Wi-Fi Alliance introduced a new generational, numbered naming convention in 2019. For example 802.11n is now also referred to as Wi-Fi 4 and 802.11ac as Wi-Fi 5. The recently published 802.11ax is the 6th Generation of Wi-Fi, and the recently announced 802.11be Working Group will now develop the 7th generation. However the new convention is mostly a marketing exercise and the lettered-based convention is still used throughout the technical literature.

2.2.1.5.2 IEEE 802.11-1997 The original 802.11 specification [2] was published in 1997 and designed for use in home and office networks providing data rates of 1 and 2 Mbps in the unlicensed 2.4 GHz ISM band using a 22 MHz signal bandwidth. Three distinct PHY types were prescribed: Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and the now obsolete Infrared (IR). The FHSS PHY works by rapidly switching transmission frequency in a predetermined pseudo-random fashion. Available frequency resources are divided into a series of frequency slots of width 1 MHz and time is also slotted. Both sender and receiver have knowledge of the transmission pattern so that the send and receive frequencies can be synchronised at both ends. The FHSS PHY uses Gaussian Frequency Shift Keying (GFSK) modulation which encodes data in the frequency changes of the carrier. The technique is substantially immune to noise but has a relatively low symbol rate. The DSSS PHY uses spread-spectrum techniques to transmit data over a very-wide frequency range. A chipping sequence is applied that spreads the carrier over the entire frequency band, which helps mitigate the adverse impact from narrow-band interference more effectively. The DSSS PHY uses Differential Phase Shift Keying (DPSK) modulation which encodes data inside the phase changes of the transmitted signal. Two MAC types were also specified: the decentralised DCF and the scheduled but now obsolete Point Coordination Function (PCF). Note that the PCF was never widely adopted. The DCF is dealt with in detail later in the section.

2.2.1.5.3 IEEE 802.11b-1999 The second 802.11 specification was published in September 1999 as IEEE 802.11b [4]. The new iteration extended the achievable data rates up to 11 Mbps in the 2.4 GHz ISM band using a High Rate DSSS (HR-DSSS) PHY by leveraging Complimentary Code Keying (CCK). The technique uses a pair of codes called chipping sequences which are complimentary such that the number of pairs of like elements in one is equal to the number of pairs of unlike elements in the other with both sequences having the same separation. CCK has a shorter chopping sequence of 8 bits compared to 11 bits in the traditional Barker code, which results in less spreading but higher data rates. As a result the signal is more prone to narrow band interference which also reduces the achievable range. The standard was also the first to feature ASR which allows an STA to adapt its current data rate dynamically according to changes in the channel conditions such as signal strength and interference. Note that in this standard MAC sub-layer was left largely unchanged from the first 802.11 standard.

2.2.1.5.4 IEEE 802.11a-1999 The first 5 GHz Wi-Fi standard was introduced in September of 1999 as 802.11a [3] and significantly increased the achievable data rates up to 54 Mbps using a new PHY based on Orthogonal Frequency Division Multiplexing (OFDM). The overhauled PHY used multiple closely-placed and orthogonal sub-carriers with partially overlapping spectra, as illustrated in Figure 2.4 below, over which data is transmitted robustly in parallel. Three different channel bandwidths of 5, 10, and 20 MHz are specified offering maximum data rates of 13, 27, and 54 Mbps respectively. The prescribed modulation types are Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and Quadrature Amplitude Modulation (QAM). A later addition to 802.11a standard was the 802.11h amendment that defined a spectrum sharing mechanism called Dynamic Frequency Selection designed to mitigate the interference of Wi-Fi with certain radar devices that also operate in the 5 GHz band. Again the MAC layer was largely unmodified.

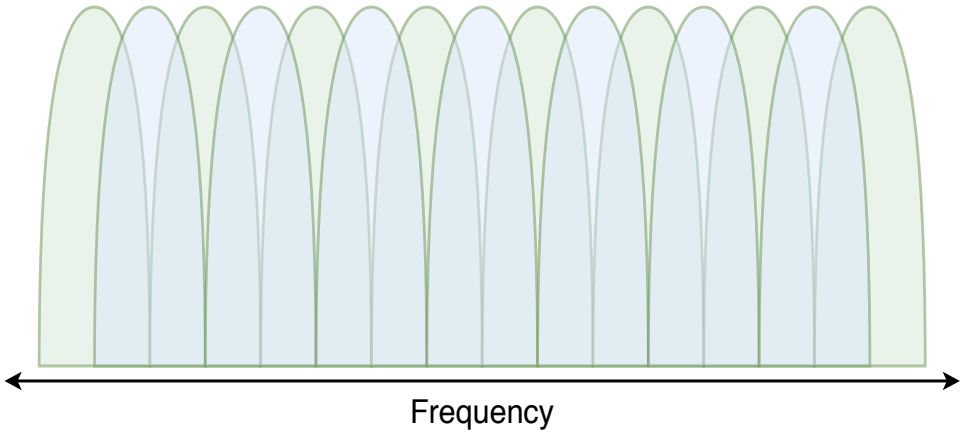

Figure 2.4: Orthogonal Sub-Carriers in 802.11a OFDM

2.2.1.5.5 IEEE 802.11g-2003 IEEE 802.11g was the third 2.4 GHz standard published in 2003 and continued the previous trend of *ever-faster* Wi-Fi. Being fully backwards compatible with 802.11b the new specification could provide data rates up to 54 Mbps using the Extended Rate PHY (ERP) which was similar to the OFDM-based PHY in the 5 GHz 802.11a standard. The 802.11g standard also provided options for ERP-DSSS and ERP-DSSS-CCK PHY types. The prescribed modulation techniques were BPSK, QPSK, 16-QAM and a higher density 64-QAM. The standard made support for the PLCP short preamble mandatory and also introduced the CTS-to-Self feature designed to reduce the number of collisions occurring on the channel as the most significant change at the MAC sub-layer.

2.2.1.5.6 IEEE 802.11i-2004 The IEEE 802.11i standard was published in 2004 with the purpose of addressing the widely known security vulnerabilities in the previous Wired Equivalent Privacy (WEP) security specification. The vulnerabilities included weak key recovery, poor data integrity, and a lack of mutual authentication between STA and AP. The new security standard introduced a more secure four-way handshake (shown in Figure 2.6 below) between wireless stations and access points, improved random number generation with the Pseudo Random Function (PRF), and replaced the vulnerable Rivest Cipher 4 (RC4) encryption with a stronger algorithm based on the Advanced Encryption Standard (AES).

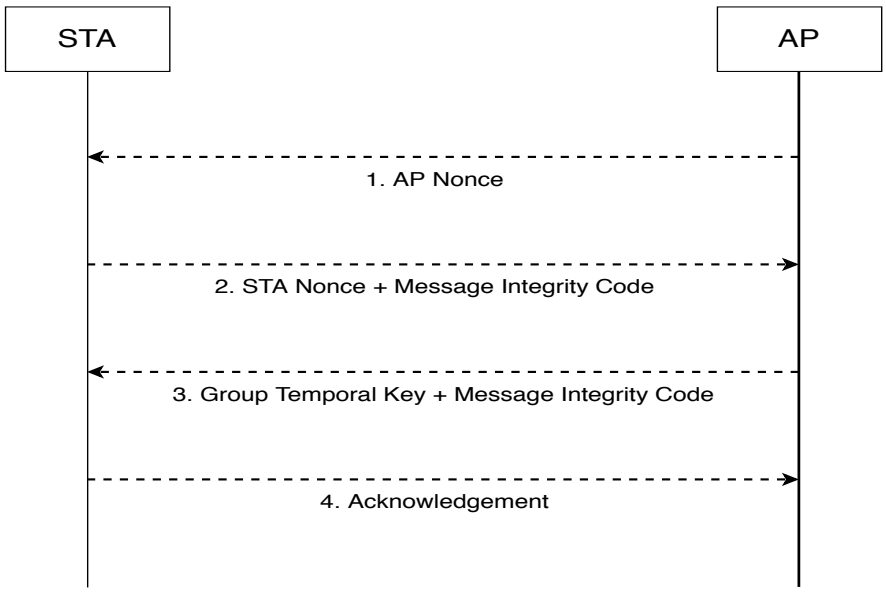

Figure 2.5: IEEE 802.11i 4-Way Authentication Handshake

2.2.1.5.7 IEEE 802.11e-2005 Quality of Service (QoS) was first introduced in the 802.11e amendment published in 2005 [6]. The Enhanced Distributed Channel Access (EDCA) mechanism differentiates between specific types of data traffic, for example, Voice over IP (VoIP) and file downloads, using four different traffic classes: Background, Best Effort, Voice, and Video. High priority packets are assigned a shorter duration of inter-frame spacing known as Arbitration Inter-Frame Spacing. The EDCA provides contention-free channel access for a period called the Transmission Opportunity (TXOP) in which a node is allowed to send as many frames as possible. Traffic from different classes originating from the same node are resolved internally using a novel virtual contention process.

2.2.1.5.8 IEEE 802.11n-2009 802.11n [9] was published in 2009 and combines the previous 2.4 and 5 GHz standards into a single specification. Among the most notable improvements was the introduction of Channel Bonding which increased the link bandwidth size to 40 MHz. The standard was also the first to introduce Multiple Input Multiple Output (MIMO) techniques to reach data rates of 600 Mbps using four spatial streams, and the first to feature a Frame Aggregation mechanism allowing increased throughput by sending multiple data frames in a single transmission. Full backwards compatibility with 802.11a/b/g was provided via Legacy, Mixed or Greenfield modes of operation.

2.2.1.5.9 IEEE 802.11p-2010 The 802.11p [10] specification was published in 2010 and defined a set of enhancements supporting Intelligent Transport Systems (ITS). Known as the Wireless Access in Vehicular Environments (WAVE) standard, 802.11p operates in the 5.9 GHz ITS band using a 10 MHz wide channel to give data rates of up to 27 Mbps between high-speed vehicles and between vehicles and roadside infrastructure. In the US and Europe along with the IEEE 609 family, 802.11p forms part of the suit of protocols used in Dedicated Short Range Communications (DSRC) aimed at automotive communications. The DSRC spectrum is split into seven 10 MHz channels. The control channel (178) is used for safety communications only, and the two channels (172 and 182) at either end of the band are reserved for special use. The remaining channels (174, 176, 180, and 182) are used for both safety and non-safety related applications.

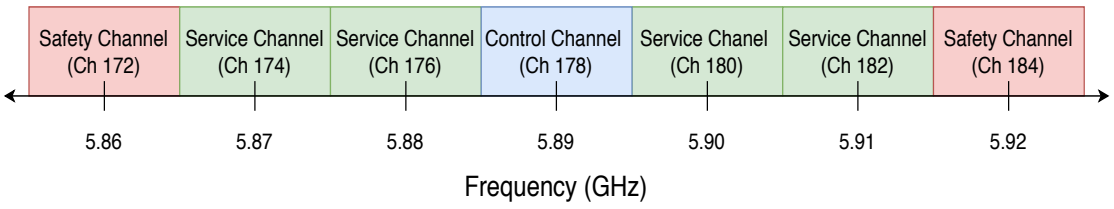

Figure 2.6: IEEE 802.11p Channels for Dedicated Short Range Communications

2.2.1.5.10 IEEE 802.11s-2011 The 802.11s standard [5] was published in 2011 and contains a set of extensions for the construction of multi-hop, wireless mesh networks wherein multiple APs connect together in a Mesh Basic Service Set (MBSS) to form the wireless back-haul of an 802.11 network. The main benefits are increased flexibility, easier installation, and greater range provided by the use of the wireless back-haul compared to the traditional wired system. Mesh APs communicate with traditional non-mesh stations in the Distribution System via a logical component called the Mesh Gate. A Mesh Coordination Function (MC) is used to regulate channel access between the mesh access points using a combination of scheduled and contention-based approaches. The Hybrid Wireless Mesh Protocol provides a scheme for path selection between mesh nodes based on an enhanced version of the Ad Hod On Demand Distance Vector Protocol.

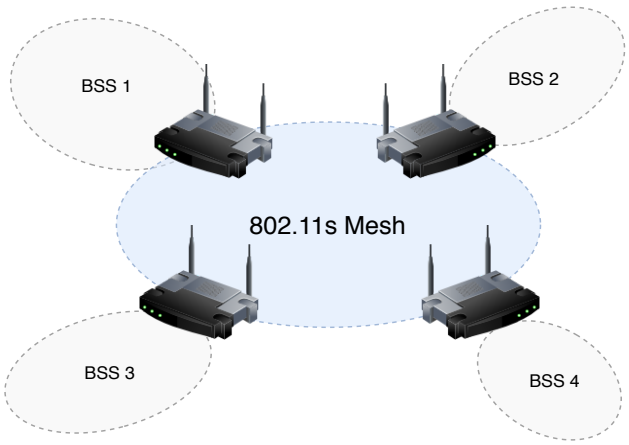

Figure 2.7: IEEE 802.11s Mesh Topology

2.2.1.5.11 IEEE 802.11ad-2012 Published in 2012 the 802.11ad standard was the first to extend Wi-Fi for use in the milli-metre wave range. Known as WiGig the specification operates between 57 GHz and 71 GHz using 6 channels with a channel bandwidth of 2.16 GHz to provide short range communications and data rates up to 8 Gbps. To overcome the increased free space path loss at the milli-metre wave range 11ad introduces directional beam-forming techniques. The Directional Multi-Gigabit (DMG) PHY has three modes of operation: the Control PHY used to send control messages and signalling and OFDM-based and Single Carrier PHY modes used for data transmission. The implementation and adoption of the standard has been slow due to the increased technical challenges and expense of milli-metre wave operation. For example the first commercially available WiGig router the TP-Link AD7200 did not come to market until 2016, nearly four years after the publication of 802.11ad.

2.2.1.5.12 IEEE 802.11ac-2013 A further set of 5 GHz enhancements were made in 802.11ac [12] which was published in 2013 and continued the trend of *ever-faster* Wi-Fi. The 5th generation of Wi-Fi introduced the Very High Throughput OFDM PHY and higher density modulation up to 256-QAM. The MIMO capability was increased up to 8 spatial streams to achieve theoretical data rates in excess of 6 Gbps. The Channel Bonding functionality was also extended to allow aggregated channels up to 160 MHz. The Wi-Fi Alliance split the release of the 802.11ac standard into two phases or waves: Wave 1 provided the basic specification and techniques and Wave 2 provided additional or advanced features such as support for four spatial streams, 160 MHz bandwidths, and additional 5 GHz channels.

2.2.1.5.13 IEEE 802.11af-2014 The 802.11af standard [62] was designed specifically for use in the under-utilised Television White Space (TVWS) located between 480 MHz and 790 MHz in the EU and between 54 MHz and 698 MHz in the USA. The standard was developed after organisations such as the Office of Communications (OFCOM) [138] in the UK and the Federal Communications Commission (FCC) in the USA regulated to allow controlled and limited use of the TVWS frequencies by non-licensed spectrum users. Published in 2014 the specification uses the same OFDM-based PHY as 802.11ac except using a channel bandwidth of 6 or 7 MHz depending on the regulatory domain. Data rates up to 35.6 Mbps are provided and while this was slower than the fastest generation of Wi-Fi at the time (802.11ac) the propagation characteristics of the TVWS band provide far greater range and penetration compared to 2.4 GHz. As well as having an LBT restriction 802.11af has an additional regulatory requirement: before deciding whether to access a particular channel resource all 802.11af nodes must first consult a Geolocation Database managed by regulatory authorities and containing a list of geographic locations and permissible frequencies. If the desired channel is not in use by the primary subscriber then the node may attempt to access the channel but must still use LBT.

2.2.1.5.14 IEEE 802.11ah-2017 Operating in the sub GHz ISM band the 802.11ah standard [13] was designed specifically for the IoT. The standard provides long-range, low-power communications and moderate to high date rates, which is ideal for wireless sensors and IoT due to the battery powered nature of such devices. Known as Wi-Fi "HaLow" 802.11ah provides data rates up to 347 Mbps using a variety of channel bandwidths. The basic PHY is OFDM as per 802.11ac. However, it is a ten-times clocked down version with its symbol duration increased from 4 μs to 40 μs. As a result the resulting bandwidths are ten times smaller: 1, 2, 4, 8, and 16 MHz. Available modulation types include BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM. The 802.11ah standard provides support for an increased total number of 8191 STAs associated with an AP by utilising a hierarchical system of Association Identifiers. To provide energy efficiency for battery-powered IoT devices there are several features such as shorter headers, implicit control frame acknowledgement, and speed frame exchange, which allows bidirectional frame transfer during a specific TXOP. More information on 802.11ah can be found in Section 2.3.3.

2.2.1.5.15 IEEE 802.11ax-2019 The current 6th generation of Wi-Fi has bucked the previous trend of *ever-faster* speeds and instead focused on increasing the overall efficiency. Published in 2019 the standard is known as 802.11ax High-efficiency Wireless [51] and is designed to work in all ISM bands between 1 GHz and 6 GHz in addition to the traditional 2.4 GHz and 5 GHz bands. For a single spatial stream the maximum theoretical data rate is around 2.5 Gbps. The new release marks a significant shift towards a centralised medium access strategy bringing Wi-Fi much closer in operation to a centralised channel access strategy such as LTE. Similar to OFDM the Orthogonal Frequency Division Multiple Access (OFDMA) PHY allocates resources in the frequency domain but also allocates temporal resources reducing the level of contention on the shared channel and placing the 802.11 AP fully in control of the network transmissions. The new random channel access mechanism, called Uplink OFDMA Random Access, is used only in the uplink when STAs wish to send buffer status reports or bandwidth requests to the AP. Ultra-dense 1024-QAM modulation is introduced which has four times the density and speed of 802.11ac. Furthermore each STA now has its own individual Target Wake Time designed to reduce power consumption and extend battery life.

2.2.1.5.16 IEEE 802.11be-202X Although the 6th generation of Wi-Fi technology is only just being brought to market at the time of writing work has already begun on the 7th titled 802.11be Extremely High Throughput. The standard will cover a range of frequency bands from 1 GHz to 7.125 GHz and extend individual channel bandwidths up to 320 MHz. The MIMO capabilities will also be increased to up to 16 spatial streams. In addition to the usual generational increases in raw speed 802.11be will feature several efficiency improvements with respect to cooperation and synchronisation between 802.11 nodes. A main objective of 802.11be is the efficient use of aggregated multi-channels in order to exploit the multi-band connectivity of mobile STAs and APs. A key optimisation being considered is a new requirement for STAs to synchronise the start of their TXOPs across different frequency bands. Other potential improvements in multi-channel aggregation include simultaneous transmission and reception across different frequency bands (multi-channel full-duplex), simultaneous transmission and reception in the same channel (full-duplex) and the separation of data and control plane traffic across different frequency channels. To allow more efficient use of available frequency, time, and spatial resources there are several improvements being considered with respect to the multi-AP coordination. Coordinated OFDM can allow adjacent APs to use orthogonal time and frequency resource allocation and synchronise data transmission in order to reduce contention compared to the use of independent CSMA/CA and also reducing the worst case latency. Also under consideration are Coordinated Null Steering, a method of reducing interference by coordinating with unmanaged adjacent APs, and Distributed MIMO which allows a combined data transmission between multiple STAs while reusing frequency and time resources.

2.2.1.6 MAC Sub-layer

The main role of the MAC sub-layer is to regulate access to the shared physical transmission medium. From a protocol layer perspective the 802.11 MAC receives a MAC Service Data Unit (MSDU) from the LLC and adds a MAC header to form a MAC Protocol Data Unit (MPDU), which is then passed down to the PHY layer for transmission. On receipt of a MAC frame the opposite process is followed with each layer stripping off its header and passing the payload to the next layer. A strict frame formatting scheme is provided with each frame consisting of a MAC header, a variable length frame body, and a 32-bit Cyclic Redundancy Check (CRC). Other functions performed by the MAC include fragmentation, association, and authentication. Due to the principle of encapsulation the PHY and MAC layers have evolved independently but the two layers are not entirely decoupled: DCF parameters such as contention window size and slot duration are tuned to the needs of a given PHY type.

2.2.1.6.1 Distributed Coordination Function For all versions of the 802.11 standard excluding the new 11ax the basic random access mechanism is the DCF which uses a positive acknowledgement scheme and a channel sensing algorithm called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). When a DCF node has a frame to send it senses the medium for aDistributed Inter-Frame Spacing (DIFS) duration. If idle the node starts a random timer with its duration taken randomly from a predefined range called the contention window. The timer is paused when transmissions are ongoing and it resumes after an idle period of length DIFS. When the random timer expires the node transmits its data. In the event of a lost acknowledgement (ACK) a collision is inferred. In order to avoid further potential collisions the node backs off, i.e. the range from which the random waiting time is drawn is doubled. The node keeps track of how many failed transmission attempts have occurred and drops the frame after a specified number. The general aim of the DCF is to minimise the number of frame collisions which occur on the channel and maximise the overall system throughput while providing a linear delay response with increasing load. The DCF frame sequences are illustrated in Figure 2.8 below and the CSMA/CA state diagram is shown in Figure 2.9 on the following page.

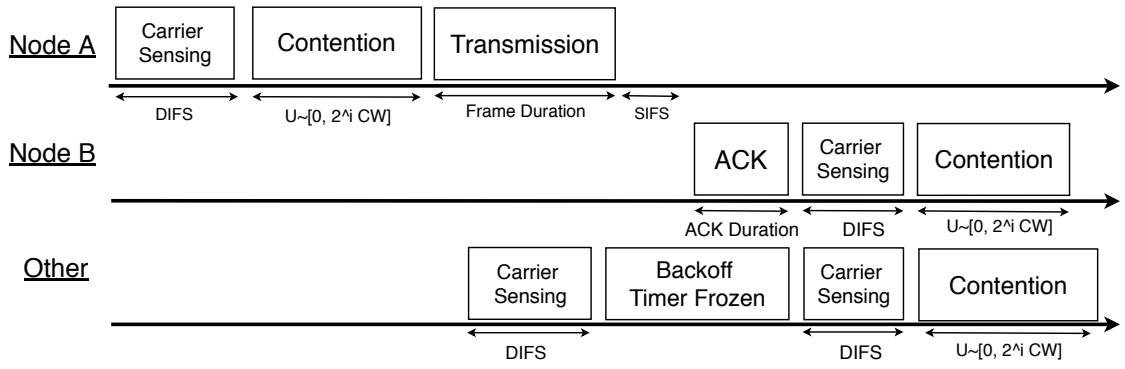

Figure 2.8: The IEEE 802.11 Distributed Coordination Function

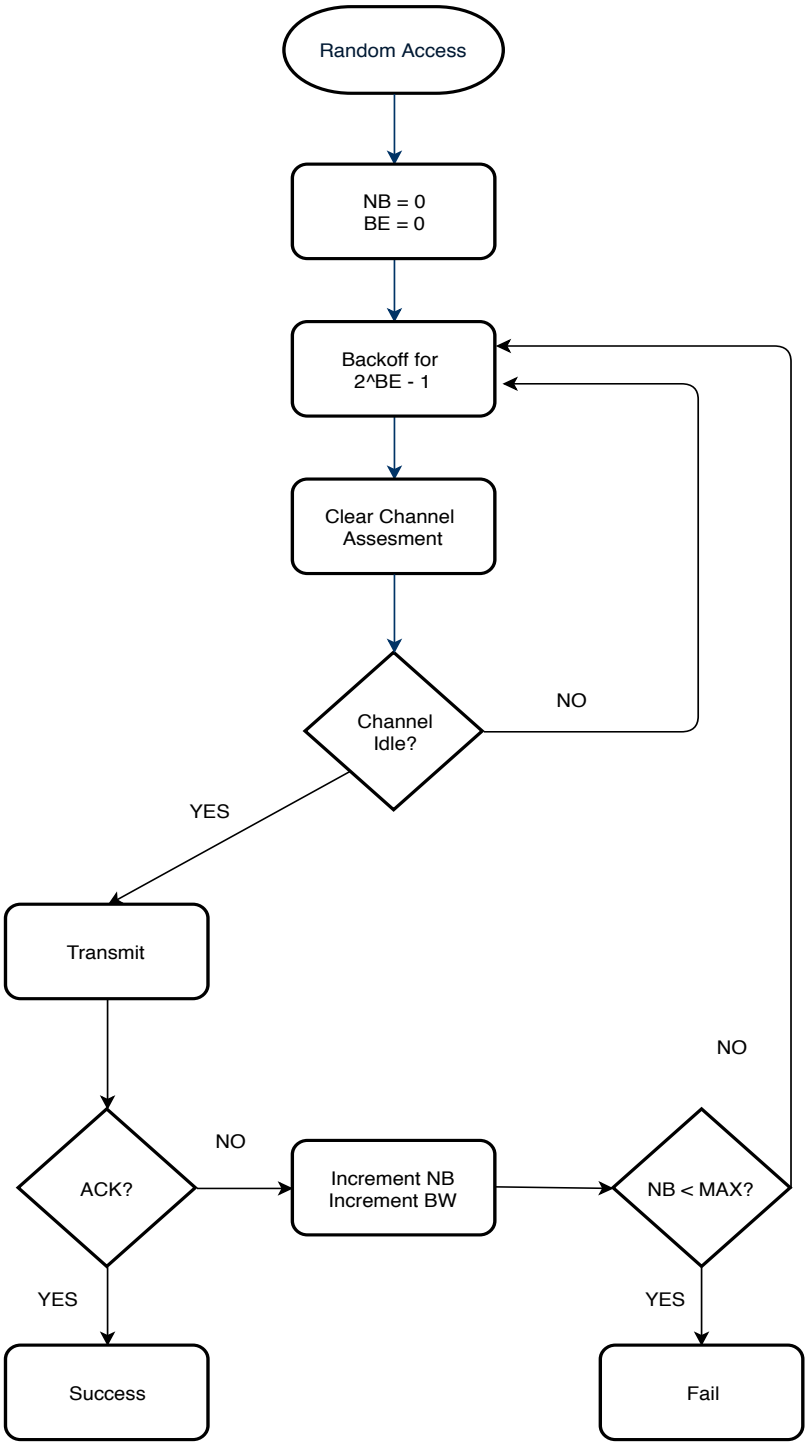

Figure 2.9: The IEEE 802.11 CSMA/CA State Diagram

2.2.1.6.2 RTS-CTS and NAV To mitigate the impact of the hidden node problem in wireless networks the 802.11 MAC defines an optional four-way handshake called RTS-CTS. In the mechanism the source node sends a small packet called an RTS to the destination, asking for permission to transmit. The destination confirms that the channel is clear using a CTS, and the source node is then free to transmit its data. The destination responds with an ACK to complete the handshake. RTS and CTS frames carry the length of the frame waiting for transmission, so other nodes overhearing can make note of it in a Network Allocation Vector (NAV) and maintain radio silence for the duration. Because both frames carry this information, only one of them needs to be received, allowing nodes outside of the carrier sensing range of one another to avoid colliding.

2.2.1.6.3 Uplink OFDMA Random Access As discussed previously the new 802.11ax specification represents a significant shift away from the decentralised MAC strategy of the DCF and CSMA/CA and towards a more centralised approach such as those used by cellular technologies such as LTE and WiMAX. However for 802.11ax both the downlink and the uplink are not yet *fully* scheduled. In the 802.11ax Uplink OFDMA Random Access mechanism the contention period is initiated by a trigger frame sent by the AP. Each node initialises its back-off counter in the range of 0 to the maximum initial contention window size (OCW). After receiving the trigger frame reply containing the resource units allocated for random access by the AP, each node subtracts this number from the counter. When it reaches zero, the STA randomly selects an available resource unit for transmission. When more than one STA chooses the same resource unit, a collision occurs and the contention window grows according to $min(2OCW_{i-1}+1, OCW_{max})$. The OFDMA approach is illustrated in the diagram below at Figure 2.10.

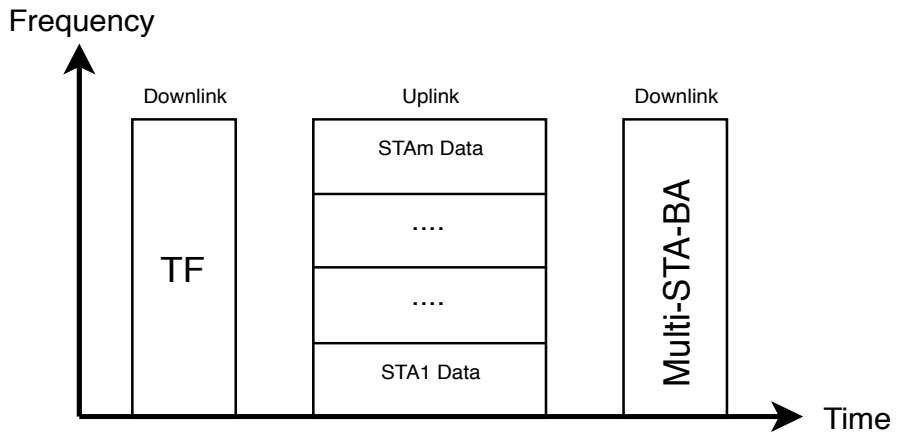

Figure 2.10: The IEEE 802.11 UL-OFDMA-RA Mechanism

2.2.2 Modelling the 802.11 MAC Layer

The numerical analysis presented in Section 2.3 and the DCF simulator validation in Section 2.4 rely of the use of a well-known 802.11 DCF channel model presented by Bianchi and a set of delay extensions proposed by Chatzimisios.

To provide the reader with the background knowledge required to understand the implementation and limitations of the main thesis research methodology this section provides an overview of wireless MAC analysis concepts and reviews published techniques for the mathematical modelling of the 802.11 DCF with a detailed description of the aforementioned works.

2.2.2.1 MAC Modelling Concepts

The analysis of random access MAC protocols involves the construction of mathematical models, the derivation of useful expressions such as the system throughput, and the validation and evaluation using techniques such as Monte Carlo simulation or hardware-based evaluation. One of the main challenges is mathematical tractability, as due to the complex nature of modern MAC protocols expressions can quickly become unwieldy. Therefore concessions are required that limit system functionality to make the formulas easier to manipulate.

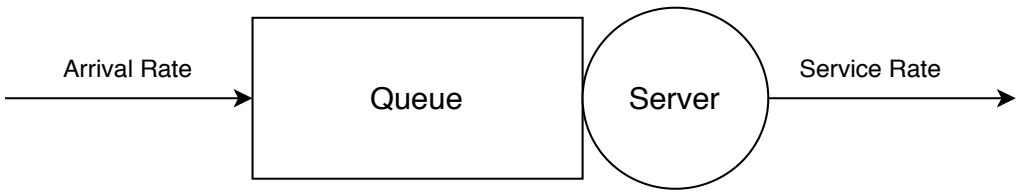

Figure 2.11: Single Server Queuing Model

A simplification assumed in most DCF analysis works found in the literature is a single-hop network topology such that all competing stations are within the transmission range of each other, i.e. there no hidden or exposed nodes present. Modelling more complex multi-hop wireless topologies can be done, for example an 802.11s mesh network, but it involves more complicated algebra. Another commonly used traffic assumption is that of Poisson arrivals in which N number of mutually independent stations share the channel and the expected number of data unit arrivals during a period is a constant denoted by λ. Another common arrival assumption is saturated traffic whereby each node always has a frame in its transmission buffer ready to send.

A simple channel model is that of a single server queue as illustrated below in Figure 2.11.. The system or service rate is given as the average time taken to transmit a single data frame on the channel, and in general the rate is dependent on the size of the frame being transmitted, the raw PHY data rate, and the added latency of the MAC algorithm. The frame size is often given

by a constant or expected value or in some cases even as a probability distribution. In most MAC layer analyses the PHY rate is constant. The channel load is given as the arrival rate divided by the service rate: $\rho = \frac{\lambda}{\mu}$ such that when $\rho < 1$, the system is in a steady state and the throughput is maximised.

2.2.2.2 Modelling Example: The Aloha Protocol

Here we provide an example of wireless MAC protocol modelling by describing the techniques used to analyse one of the first wireless data communications systems - the Aloha protocol [194].

The channel access strategy of Aloha was to permit the competing stations to transmit whenever they had a data frame ready to send. To detect collisions the central network computer re-transmits a received frame by broadcasting it back into the network. In this way the sender can be sure that the data was received successfully at the destination. If a station experiences a collision when transmitting the frame on the shared channel then it cannot immediately attempt to re-transmit the frame, otherwise the same stations would persistently collide together as their frames were resent. Instead each node chooses a random length of time to wait before re-transmitting. This is known as a contention-based channel access approach, as opposed to a collision-free approach in which all uplink and downlink transmissions are fully scheduled by a central control node.

In Aloha arrival traffic is assumed to be modelled by a Poisson distribution with a mean of N frames per time interval. The system throughput is denoted by S and is defined as the offered load, λ, multiplied by, p_s, the probability of a successful transmission, i.e. $S = \lambda p_s$. The probability that k frames are generated during a given time interval in which G number of frames is expected on average, is given by the Poisson distribution as $P_k = \frac{G^K e^- G}{k!}$. This also implies that the probability of no frames being received during the interval is e^{-G}. When the time interval is two frames long the mean number of generated frames is $2G$, and the probability if no frames being transmitted during the entire two-frame period is given by $P_0 = e^{-2G}$. Using the relation $S = GP_0$ the Aloha throughput becomes $S = Ge^{-2G}$.

Later the the use of slotted time was established as a way of reducing collisions and increasing the overall Aloha network throughput. Turning the continuous time of the original Pure Aloha protocol into slotted discrete time had the effect of halving the vulnerable period if a transmission and doubling the overall system throughput to $S = Ge^{-G}$.

However since the original Aloha protocol was developed in the early 1970s wireless MAC protocols have increased significantly not only in efficiency, but also in complexity. As shown for the DCF in the following section modelling the throughput and delay behaviour of modern MAC protocols such as 802.11 is far more complex due to advanced mechanisms such as carrier sensing, back-off, RTS-CTS, and NAV.

2.2.2.3 Bianchi DCF Saturation Throughput Model

In his seminal work [30] Bianchi constructed a bi-dimensional discrete time Markov chain representing a CSMA/CA node from which expressions for the conditional collision probability and saturation throughput of the 802.11 DCF were derived.

The original paper has been applied and cited in several thousand researches with many building on or extending the model in some way, for example, the delay expressions provided by Chatzimisios [1] and the non-saturation model by Felemban [60].

Note that the Bianchi model is central to the validation of the 802.11 DCF simulator presented in Section 2.4 which is an important contribution to the main thesis research work.

2.2.2.3.1 Model Assumptions The main modelling assumptions of the analysis are: (1) Network Topology: A flat, single-hop wireless topology is assumed with no hidden or exposed nodes, (2) Traffic Arrivals: saturated traffic arrivals are assumed such that each node always has a frame in its transmission queue ready to send, (3) Re-transmissions: an infinite number of frame re-transmission attempts are allowed, (4) Collision Decoupling: the collision process is not dependent on past transmission history, and (5) Ideal Channel Conditions: i.e. the only source of frame loss considered are collisions occurring on the shared channel.

2.2.2.3.2 Chain Construction Note that the back-off counter process of the DCF, $b(t)$, is non-Markovian, as the waiting time is dependent on the past transmission history of the node. However, the embedded back-off stage process $s(t)$ is Markovian. Assuming that each frame collides with constant and independent probability, p, a two-dimensional discrete time Markov chain, $\{s(t), b(t)\}$, can be developed. Let the maximum back-off value stage, m, be the value such that $CW_{max} = 2^m W$, where $W = CW_{min}$, and also let $W_i = 2^i W$ for $i \in (0, m)$. The resulting Markov chain is shown in Figure 2.12 below.

2.2.2.3.3 Transition Matrix The only non-null transition probabilities of the Markov chain constructed by Bianchi are given below:

$$P_t = \begin{cases} P\{i,k|i,k+1\} = 1, & k \in (0, W_i - 2) i \in (0, m) \\ P\{0,k|i,0\} = (1-p)/W_0 & k \in (0, W_0 - 1), i \in (0, m) \\ P\{i,k|i-1,0\} = p/W_i & k \in (0, W_i - 1), i \in (1, m) \\ P\{m,k|m,0\} = p/W_m & k \in (0, W_m - 1) \end{cases}$$

2.2.2.3.4 Stationary Distribution The stationary distribution of a Markov chain represents the long-term proportion of time spent by the chain in each of the possible states under stable traffic conditions. The stationary distribution of the chain described in this section has a closed-form solution, which is derived by Bianchi as follows. Let $b_{i,k}$ be the stationary distribution of the chain and note the following:

$$\begin{aligned} b_{i-1,0}p = b_{i,0} \rightarrow b_{i,0} = p^i b_{0,0}, \quad for \;\; 0 < i < m \\ b_{m-1,0}p = (1-p)/b_{m,0} \rightarrow b_m = \frac{p^m}{(1-p)b_{0,0}} \end{aligned} \tag{2.1}$$

We also note that:

$$b_{m-1,0}p = (1-p)/b_{m,0} \rightarrow b_m = \frac{p^m}{(1-p)b_{0,0}}$$

Next, due to the chain regularities this gives:

$$b_{i,k} = \frac{W_i - k}{W_i} = \begin{cases} (1-p)\sum_{j=0}^{m} & i = 0 \\ p b_{i-0,0} b_{j,0} & 0 < i < m \\ p(b_{m-1,0} + b_{m,0}) & i = m \end{cases}$$

Which by means of relation 2.1 and the fact that $\sum_{i=0}^{m} b_{i,0} = b_{0,0}/(1-p)$ can be rewritten as:

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0} \quad i \in (0,m), \quad k \in (0, W_i - 1)$$

The value of $b_{0,0}$ is found by applying the normalisation condition and re-arranging to give:

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW(1-(2(p)^m)} \tag{2.2}$$

2.2.2.3.5 Station Transmission Probability Now that the value $b_{0,0}$ is known it is possible to derive an expression for the probability that a single station transmits in a given time slot. This probability is denoted by τ and given as:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1-p} = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{2.3}$$

In general the transmission probability τ depends on the conditional collision probability, p.

2.2.2.3.6 Conditional Collision Probability The conditional collision probability is the probability that a frame being transmitted experiences a collision due to the simultaneous transmission attempt of another competing station. Using τ this is given as the probability that at least one of the $n-1$ nodes also transmits:

$$p = 1-(1-\tau)^{n-1} \tag{2.4}$$

Equations 2.3 and 2.4 form a system of non-linear equations in two unknowns, τ and p. The value p is central to the throughput equation described in the next section and many other extensions to the Bianchi model including Chatzimisios.

2.2.2.3.7 Saturation Throughput Bianchi expresses the normalised throughput of the DCF as the average number of payload bits transmitted per slot time:

$$S = \frac{E[bits\ tx\ per\ slot]}{E[slot]} \tag{2.5}$$

If $E[P]$ is the expected payload size and a successful transmission occurs with probability $P_{tr}P_s$, then the average number of payload bits transmitted per slot time is given as $P_{tr}P_sE[P]$. With probability $(1-Ptr)$ a slot is empty, with probability $P_{tr}P_s$ it contains a successful transmission, and with probability $Ptr(1-P_s)$ it contains a collision. The resulting expression is given as:

$$S = \frac{P_sP_{tr}E[P]}{(1-P_{tr})\delta + P_{tr}P_sT_s + P_{tr}(1-P_s)T_c} \tag{2.6}$$

where T_s is the average length of time the channel is sensed as busy due to a successful transmission and T_c is the average time the channel is sensed as busy during a collision:

$$\begin{aligned} T_s &= H + E[P] + SIFS + \delta + ACK + DIFS + \delta \\ T_c &= H + E[P] + DIFS + \delta \end{aligned} \tag{2.7}$$

When the RTS/CTS mechanism is employed T_s and T_c become:

$$\begin{aligned} T_s &= RTS + SIFS + \delta + CTS + SIFS + \delta + H + E[P] + SIFS + \delta + ACK + DIFS + \delta \\ T_c &= RTS + DIFS + \delta \end{aligned} \tag{2.8}$$

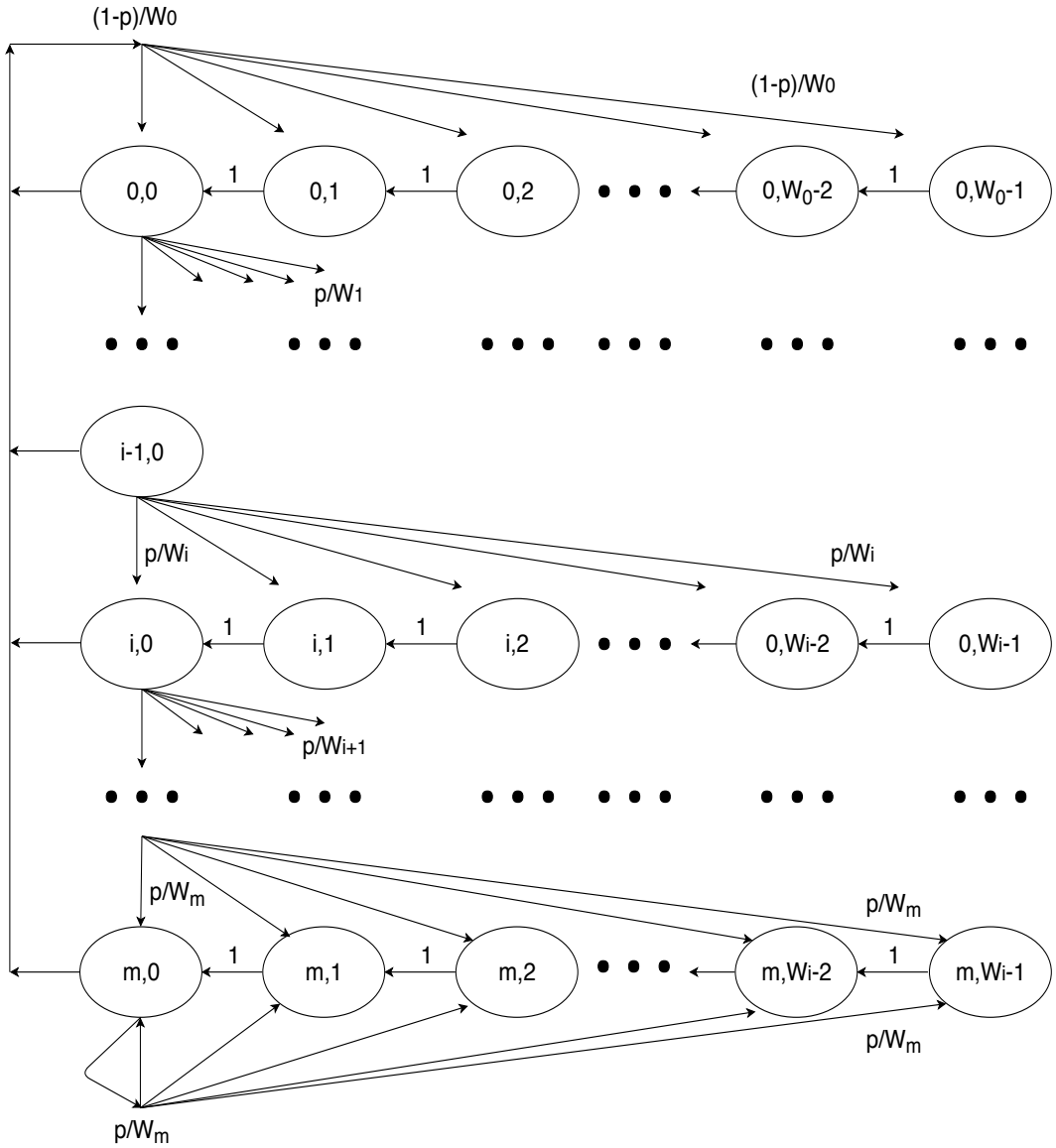

Figure 2.12: Bianchi Bi-dimensional Markov Chain for the DCF Node

2.2.2.4 Chatzimisios DCF Access Delay Model

The access delay is defined by Chatzimisios in [1] as the length of time between a frame arriving at the head of the source node transmission queue and the frame being successfully received at the sink node.

2.2.2.4.1 Infinite Re-transmissions In his basic model Chatzimisios formulates the DCF access delay as the average number of idle slot times multiplied by the average slot duration:

$$E[D] = E[X]E[Slot] \tag{2.9}$$

where $E[Slot]$ is the expected length of a slot time and $E[X]$ is the expected number of slots during a successful transmission and given by:

$$E[X] = \frac{(1-2p)(W+1) + pW(1-(2p)^m)}{2(1-2p)(1-p)} \tag{2.10}$$

2.2.2.4.2 Finite Re-transmissions Chatzimisios later adapted his model to include finite re-transmissions and the packet drop probability. In the case of a finite number of re-transmissions, m', the window size is given by:

$$W_i = \begin{cases} 2^i W, & for\ i <= m' \\ 2^{m'} W, & for\ i > m' \end{cases}$$

As with Bianchi, the DCF is modelled using a bi-dimensional Markov process. Chatzimisios derives a new expression for $b0,0$ as follows:

$$b_{0,0} = \begin{cases} \frac{2(1-2p)(1-p)}{W(1-(2p)^{m+1})(1-p)+(1-2p)(1-p^{m+1}),} & for\ m <= m \\ \frac{2(1-2p)(1-p)}{W(1-(2p)^{m+1})(1-p)+(1-2p)(1-p^{m+1})+W2^{m'}p^{m'+1}(1-2p)(1-p^{m-m'}),} & for\ m > m' \end{cases}$$

In the finite model, if a frame exceeds the maximum number of transmission attempts is it dropped, which occurs with probability:

$$p_{drop} = p^{m+1} \tag{2.11}$$

Which is the probability of $m+1$ consecutive collisions. The average number of slots to drop a packet is given as:

$$E[T_{drop}] = \begin{cases} \frac{W(2^{m+1}-1)+(m+1)}{2}, & for\ m <= m' \\ \frac{W(2^{m+1}-1)+W2^m(m-m')+(m+1)}{2}, & for\ m > m' \end{cases}$$

The average time taken to drop a packet is therefore given as:

$$E[D_{drop}] = E[T_{drop}]E[slot] \tag{2.12}$$

Again the delay is given by Equation 2.9, but in the case of finite re-transmissions $E[X]$ is given by:

$$E[X] = \begin{cases} \frac{W(1-(2p)^{m+1})(1-p)+(1-2p)(1-p^{m+1})}{2(1-2p)(1-p)} - p^{m+1}E[Tdrop], & for\ m <= m' \\ \frac{W(1-(2p)^{m+1})(1-p)+W2^m p^{m+1}(1-p^{m-m})(1-2p)+(1-2p)(1-p^{m+1})}{2(1-2p)(1-p)} - p^{m+1}E[Tdrop], & for\ m > m' \end{cases}$$

and $E[slot]$ is given by:

$$E[slot] = (1-P_{tr})\delta + P_{tr}P_s T_s + P_{tr}(1-p_{tr})T_c \tag{2.13}$$

2.2.2.5 Other DCF Analyses

This subsection presents a brief literature review of other related DCF modelling works. Note that the analytical modelling of the DCF represents a substantial body of research in its own right and several thousand works have been published over the past two decades dealing with many different aspects of DCF functionality of which the following works represent only a small portion.

Related analytical works can be classified according to three distinct categories:

1. Improvements: Such works seek to improve previous analyses by making the model more realistic and truer to the 802.11 specification. A good example of this sort of work is seen in the papers by Chatzimisios in [1] and [38]. Here the author extends his first work to accommodate finite transmission attempts in order to more accurately reflect the standardised behaviour.

2. Extensions: These works extend previous the original Bianchi model to include a new feature or aspect of a new standard or a performance metric that was not previously considered. A good example is the 802.11e model for QoS presented by the authors in [196].

3. Evaluations: These works numerically evaluate different 802.11 technologies or specific networking and traffic scenarios. For example the performance evaluation of the 802.11ah standard for IoT networks by Banos-Gonzalez et al. in [27] and the analysis of 802.11p in vehicular networks by the author Song in [181].

Publication	Topology	Traffic	Metrics	Retries	QoS	Chan.
Bianchi [30]	Single-Hop	Saturated	Coll., T.put	Infinite	No	Ideal
Carvalho [35]	Single-Hop	Saturated	Delay	Infinite	No	Ideal
Chatzimisios [1]	Single-Hop	Saturated	Delay	Infinite	No	Ideal
Chatzimisios [38]	Single-Hop	Saturated	Delay	Finite	No	Ideal
Bianchi [31]	Single-Hop	Saturated	T.put, Delay	Infinite	No	Ideal
Bianchi [68]	Single-Hop	Saturated	T.put, Delay	Infinite	No	Ideal
Tantra [196]	Single-Hop	Saturated	T.put, Delay	Finite	Yes	Ideal
Engelstad [58]	Single-Hop	Near-sat.	Delay	Finite	Yes	Ideal
Vu [208]	Single-Hop	Saturated	Delay	Finite	No	Ideal
Vardakas [204]	Single-Hop	Saturated	Delay	Finite	No	Ideal
Sharma [168]	Single-Hop	Saturated	Coll.	Finite	Yes	Ideal
Xu [216]	Single-Hop	Saturated	Delay	Finite	Yes	Ideal
Felemban [60]	Single-Hop	Non-sat.	T.put, Delay	Finite	No	Ideal
Kumar [107]	Multi-Hop	Non-sat.	T.put	Finite	No	Error
Alsbou [22]	Single-Hop	Non-sat.	T.put	Finite	No	Error
Alabady [20]	Multi-Hop	Saturated	T.put, Delay	Finite	No	Ideal
Zeeshan [226]	Multi-Hop	Saturated	T.put, Delay	Finite	No	Ideal
Wang [210]	Multi-Hop	Saturated	T.put, Delay	Finite	No	Ideal
Song [183]	Multi-Hop	Saturated	Coll., T.put	Finite	Yes	Ideal

Table 2.2: Comparison of Related DCF Modelling Works

2.2.2.5.1 Non-saturated Traffic Arrivals One of the central assumptions of the Bianchi model is saturated traffic arrivals, i.e. every competing node in the network always has a frame waiting to be sent in its transmission queue. This assumption is perhaps one of the most unrealistic as in a real-life network most application traffic is extremely bursty in nature with peak traffic to mean traffic ratios of up to 1000:1 [194]. Several research works have therefore extended Bianchi to accommodate alternate traffic assumptions such as Poisson-based arrivals. For example the work by Felemban and Ekici in [60] wherein the authors derive an expression for the of the number of active nodes used to calculate the non-saturation collision probability, delay, and throughput of the DCF. In order to model the non-saturation load an iterative algorithm is developed which computes the probability that a node does not have any packets to send using the basic saturation model. For a Poisson arrival rate per slot time equal to λ this probability is given as:

$$P_0 = 1 = \frac{\lambda}{\mu}, \tag{2.14}$$

Where μ is the average channel service rate. The PMF of the number of active stations with at least one frame ready to send in their transmission queues is given as:

$$\beta_i = \binom{N}{i}(1-P_0)^i P_0^{(N-i)} \tag{2.15}$$

The non-saturation conditional collision probability is then given as:

$$P_\lambda = \frac{\sum_{i=1}^{N} P_i \beta_i}{1-\beta_0} \tag{2.16}$$

Where P_i denotes the saturated collision probability also provided in the work. The non-saturation channel throughput is given as:

$$U_\lambda = \sum_{i=1}^{N} U_i \beta_i \tag{2.17}$$

Where U_i denotes the saturated throughput of the DCF also provided in the work. Finally the non-saturation expected delay is given as:

$$T_\lambda = \frac{\sum_{i=1}^{N} T_i \beta_i}{1-\beta_0} \tag{2.18}$$

Where T_i denotes the saturated expected delay as provided in the work.

Other notable non-saturation analyses of the DCF include Garetto and Chiasserini [71], which uses a multi-dimensional Markov chain to estimate the number of active nodes present and the number of frames per transmission queue, Engelstad [58] and Malone [119], which both employ an additional node state to model the case when a DCF node not have any data in its transmission queue, and Zhai [229] and Tickoo [199], which both scale the saturated, single-node transmission probability by the probability of that node having at least a single packet in its transmission queue.

2.2.2.5.2 Multi-hop Wireless Topologies The original Bianchi analysis assumed a single-hop, 802.11 WLAN topology, but more recently multiple research works have emerged which extend the model to accommodate multi-hop IEEE 802.11s [5] style mesh topologies. In a single-hop network when all nodes are within carrier sensing range of each other the DCF is able to co-ordinate and synchronise the transmissions of all competing stations. However this cannot be assumed in multi-hop networks and as such node transmissions may not be effectively synchronised which can lead to severe throughput degradation and even complete starvation for some nodes. Zeeshan et al. [226] use Bianchi to model the packet loss probability and busy time in multi-hop 802.11 WLANs. The authors combine the analytical formulas for the single node transmission probability and expected delay of the DCF with a novel geometric model for interference and derive an expression for the conditional collision probability for the two-flow topology in Figure 2.13.

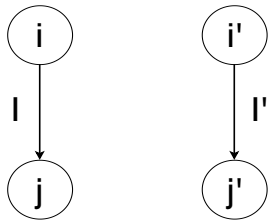

Figure 2.13: Two Flow Topology used in Zeeshan et al. [226]

This is the probability that a link $l(i,j)$ is interfered with by station i' due to a simultaneous transmission, and is calculated as the conditional probability that station i' attempts to transmit at the same time station i is already transmitting:

$$c(i'/i) = \frac{Q(\theta)}{\sum_{D(i)} Q(D)} \tag{2.19}$$

Where Q(D) represents the probability that the stations in a region are both in the active state, Q(θ) is the empty set represent the situation where no station in any region is active, and A(i) is the set of regions where station i can become active. The values are obtained via simulation and computation of the average busy time using a method similar to Garetto et al. in [72].

In another related work Alabady et al. [20] constructs a multi-hop 802.11 model and evaluates the DCF performance in the presence of hidden nodes in order to demonstrate the impact of network size, the maximum back-off stage, and collision probability on the overall system throughput and delay. Wang [210] also analyses the throughput and end-to-end delay of the 802.11 MAC over a 2-hop wireless path by differentiating between the DCF access delay and the queuing delay experienced at the intermediate nodes using M/M/1 ad M/G/1 queuing models. Dong [54] also models the intermediate delay using queuing theory, and utilises a metric called

the hidden area of a transmitting station to estimate the number of hidden nodes present in the network. Other examples of multi-hop analysis of the 802.11 MAC include [72], Khalaf [100], Sirus [176], Siripongwutikorn [175], and Abdullah et al. [14].

2.2.2.5.3 Alternative Delay Models Several other authors including Chatzimisios have developed access delay extensions to the core Bianchi model. Vu [208] derives the generating function of the DCF access delay using a numerical transform inversion. Sakurai provides an expression for the access delay and derives explicit expressions for the generating function and the first two moments. The authors Carvalho et al. [35] analyse the DCF delay in a single-hop wireless topology under saturation conditions. Other delay works include Vardakas [204], Wang [210], and even Bianchi [68], who later developed an alternate model based on elementary probability theory including the DCF access delay.

2.2.2.5.4 Quality of Service Several works have extended the Bianchi analysis to accommodate 802.11e style QoS. Such works include Tantra [196], Engelstad [58], Xu [216], Sharma [168], and Song [183]. The aforementioned works vary considerably in terms of their consideration of other aspects of the DCF such as the finite vs infinite re-transmissions and saturated vs non-saturated traffic arrivals.

2.2.2.5.5 Hidden Nodes Hidden nodes are an inherent problem in wireless networks and 802.11 is no exception. Multiple works target numerical DCF performance evaluation in the presence of hidden nodes using the RTS-CTS throughput equation provided by Bianchi, including Pal et al. [140], Kumar [107], Alsbou [22], Chang [37], Alabady [20], and Song [183].

2.2.2.5.6 Kalman Filter Estimation Bianchi [31] applies Kalman filter techniques to estimate the number of nodes present in an 802.11 network based on the measured collision probability. Bianchi show that Auto Regressive Moving Average filter estimation leads to a biased estimation and has low performance in terms of the accuracy-tracking trade off. The Kalman Filter estimation technique provides more accurate results and react more promptly to changes in network state. The technique is also shown to work effectively under non-saturation conditions. Tinnerello [200] applies a similar technique to help select the most appropriate modulation and coding schemes based on the current channel load.

2.2.2.5.7 Modelling Criticisms Analytical techniques can be accurate and revealing but a mathematical model may never reflect the idiosyncrasies of a real-life hardware-based implementation. In addition modelling assumptions made in order to simplify the construction of such models can influence the resulting behaviour. Furthermore some assumptions may not hold true for all possible conditions.

Huang et al. question the validity of the DCF modelling hypothesis in [88]. The authors test two key assumptions used by models such as Bianchi and also for other CSMA types such as p-persistent CSMA and CSMA/CA: (1) the sequence of attempted frame transmissions forms a random and mutually independent sequence, and (2) the sequence of outcomes are identically distributed random variables and do not depend on past transmission history. These assumptions are sometimes referred to collectively in the literature as the decoupling assumption. Huang considers three types of 802.11 network: (1) saturated/unsaturated infrastructure networks, (2) saturated 802.11e networks, and (3) saturated/unsaturated 802.11s mesh networks. The test results published in the work show that the above assumptions are true for saturated and unsaturated networks with small buffers but do not hold for 802.11e and 802.11s networks with saturated traffic.

2.3 Analysing the 802.11 MAC Layer using Bianchi

In order to evaluate the performance of the individual 802.11 links used in the interface bonding tests in Chapters 3 and 4 of the thesis, in this section we analyse and compare the performance of the MAC sub-layer channel access mechanism, between the 802.11ac and 802.11ah versions of the standard.

2.3.1 Example Bianchi Throughput Evaluation

First, to illustrate the numerical output provided by the Markov-based analytical model by Bianchi, the main modelling expressions are numerically evaluated and discussed for the FHSS PHY as per the original 802.11 standard used by the author in [30].

2.3.1.1 Input Parameters

The DCF parameters used in the Bianchi evaluation are summarised in Table 2.3 below. The modulation type assumed is 2-level Gaussian Frequency Shift Keying (GFSK), which provides a raw PHY data rate of 1 Mbps. The payload size of 8184 bits matches Bianchi and is set to 1 quarter of the maximum MAC Protocol Data Unit (MPDU) for the FHSS PHY. All MAC and PHY parameters were chosen to exactly match the original Bianchi publication and were also verified against the 802.11 specification in [2].

Parameter	**Value**
Number of Nodes (N)	100
Slot Time (μs)	50
CWmin (slots)	16
CWmax (slots)	1023
Maximum Backoffs	5
Payload Size	8184
MAC Header Size (bits)	272
PHY Header Size (bits)	128
ACK Size (bits)	112
SIFS Duration (μs)	28
DIFS Duration (μs)	128
Propagation Time (μs)	1
Data Rate (Mbps)	1

Table 2.3: Parameters used in Bianchi Evaluation of 802.11 FHSS

2.3.1.2 Applied Load Tests

The scenario used for the FHSS evaluation is shown in Figure 2.14 below. The applied channel load is incremented from n = 1 to n = 100 competing nodes, and at each loop iteration the DCF channel saturation throughput and conditional collision probability are calculated (using the native computational MatLab functions for solving systems of linear equations) as per the analytical expressions provided by Bianchi described in Section 2.2.2.3 above. The test output is therefore an N-length vector of the DCF throughput as a function of the applied load under the assumption of saturated traffic.

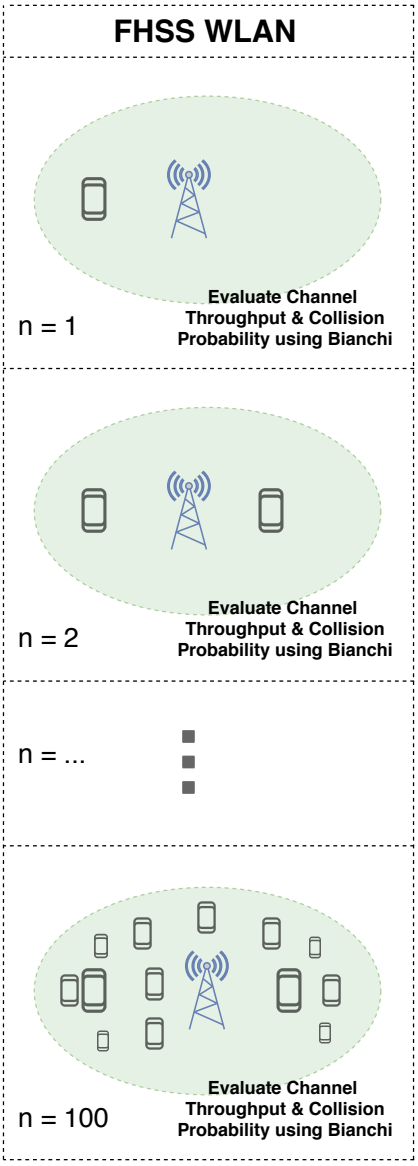

Figure 2.14: FHSS Test Applied Channel Load

2.3.1.3 FHSS Saturation Throughput Results

As per Equation 2.6 of the Bianchi model above, the saturation throughput of the 802.11 DCF is defined as the fraction of time in the long-term that the shared channel spends in the successfully transmitting state under the assumption of saturated arrival traffic.

The saturation throughput is evaluated for the FHSS PHY in Figure 2.15 below as a function of the applied load using the input parameters defined in Section 2.3.1.1 above. The overall mean DCF channel throughput for all applied loads considered was 0.5649 Mbps.

Notice that at the initial applied load of n = 1 the achieved channel throughput of 0.8761 Mbps was already much less than the raw PHY data rate of 1 Mbps due to the additional latency of the CSMA/CA back-off mechanism as well as the PHY/MAC headers and ACK frames. The respective PHY/MAC header sizes of 112 bits and 272 bits remain constant throughout the test in contrast to the delay which increases non-linearly with the applied load, as seen for the DSSS PHY in the Chatzimisios evaluation in Figure 2.21 of the following section. The increase in latency has the knock-on effect of reducing the achieved channel throughput, which is seen below where the output of the Bianchi expression decreases non-linearly to a minimum of 0.4558 Mbps at the heaviest test load of n = 100.

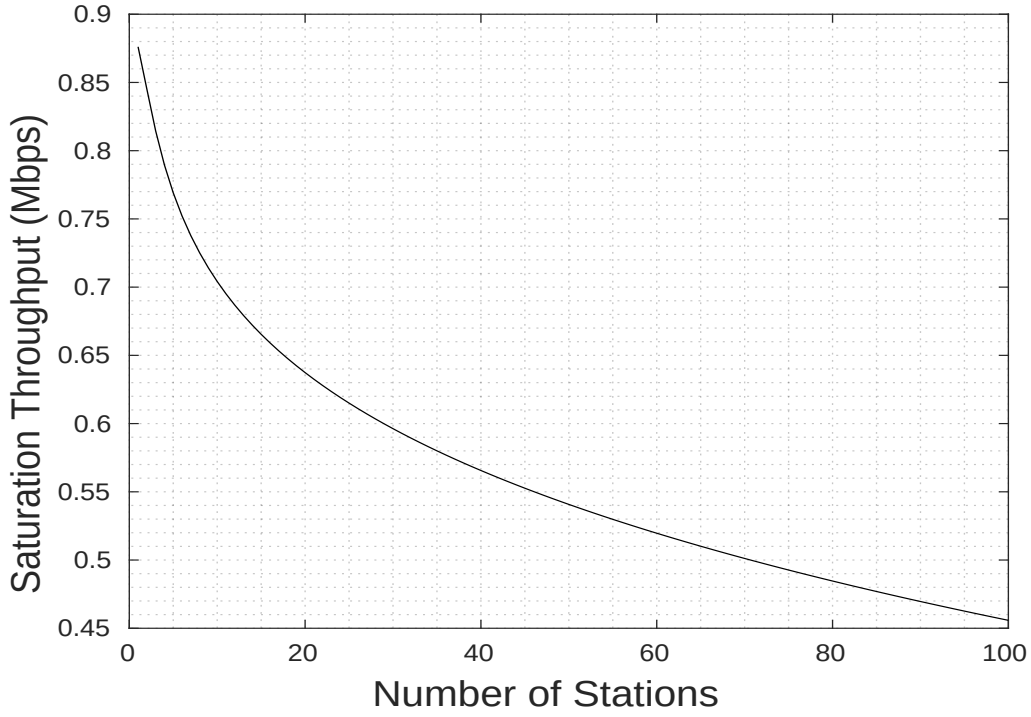

Figure 2.15: Bianchi Saturation Throughput Results 802.11 FHSS

2.3.1.4 FHSS Conditional Collision Probability

The conditional collision probability formula in Equation 2.4 above gives the probability that a frame being currently transmitted onto the DCF channel by a given STA experiences a collision with one of the other $n-1$ competing STAs due to a simultaneous transmission attempt.

The collision probability is one of the two key formulas[1] of the Bianchi model and is used to derive all the main performance metrics of the DCF including the saturation throughput and expected delay. It is evaluated for the FHSS PHY in Figure 2.16 below, where as seen the model output increases non-linearly from a minimum of zero at n = 1 (because a node cannot collide with itself) to a maximum of 0.7232 at the heaviest considered load of n = 100 with an overall mean value of 0.5845.

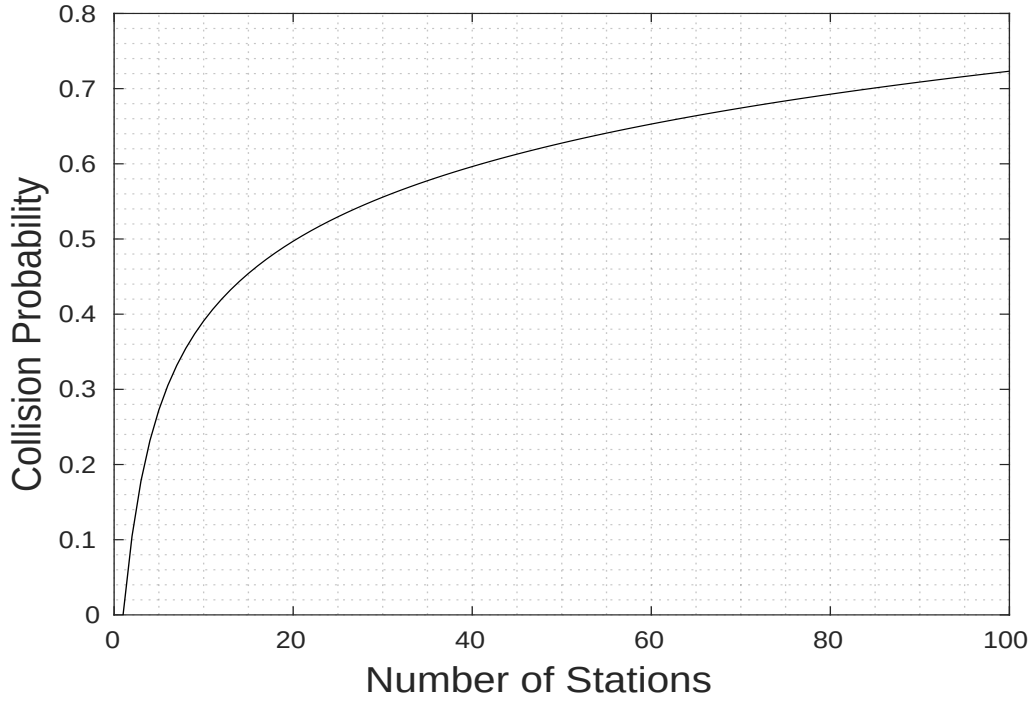

Figure 2.16: Bianchi Conditional Collision Probability Results 802.11 FHSS

[1]The other key formula is the expression for Tau as described in Section 2.2.2.3 and evaluated in Figure 2.17

Related to the collision probability is the value represented by τ shown in Equation 2.3 above used in the Bianchi Markov model to derive the stationary distribution of the chain by providing the probability of the chain being in the initial or starting state $b_{0,0}$.

Along with the conditional collision probability P_c, the expression τ forms the system of non-linear equations that must be solved when performing the numerical evaluation of a specific 802.11 standard. The overall mean value of τ for all applied loads considered was 0.5987, and as seen from the graph the model output increases non-linearly from a minimum of 0.1176 at n = 1 to a maximum of 0.7267 at the heaviest considered channel load of n = 100.

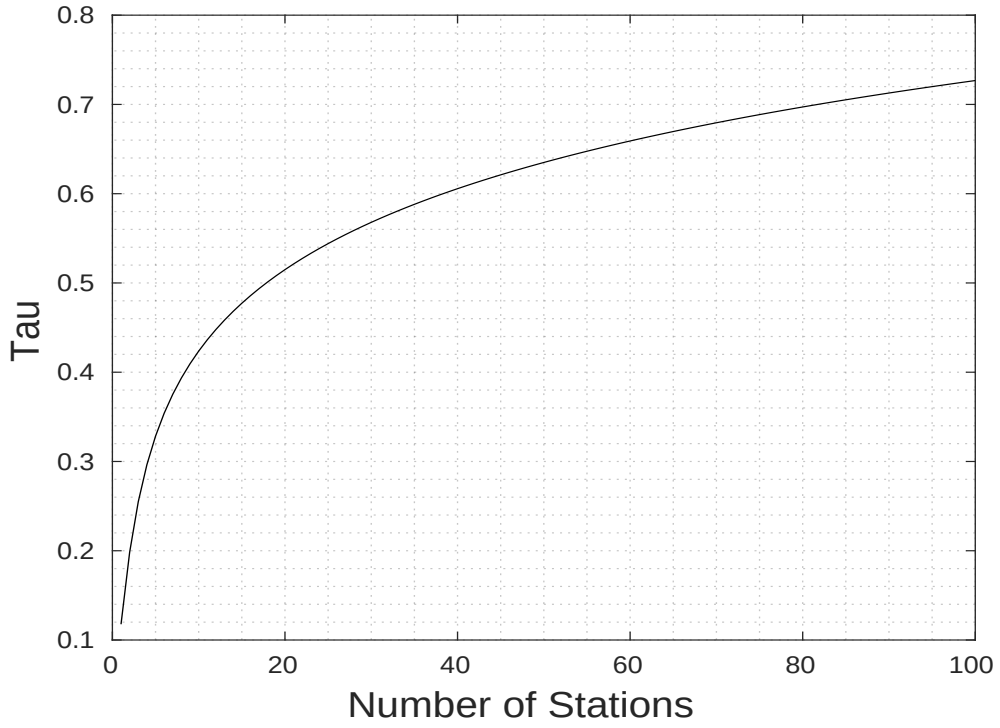

Figure 2.17: Bianchi Tau Probability Results 802.11 FHSS

In the Bianchi model the parameter P_{tr} gives the probability that at least one of the n competing stations present in the network will attempt to transmit a data frame during a given time slot. The expression is used in both the numerator and denominator of Equation 2.6 to derive the saturation throughput of the DCF. Shown below, the expression looks almost identical to the conditional collision probability, but instead of raising to the power of $n-1$ the bracket contents are raised to the power of n:

$$p_{tr} = 1-(1-\tau)^n \tag{2.20}$$

As such its evaluated value is close to that of P_c throughout the results, as shown in Figure 2.18 and Figure 2.16. For the initial load of n = 1 the transmission probability is 0.1176 and the output then increases non-linearly to a maximum of 0.7267 at the heaviest considered load of n = 100. The overall mean transmission probability for all applied loads is 0.5987.

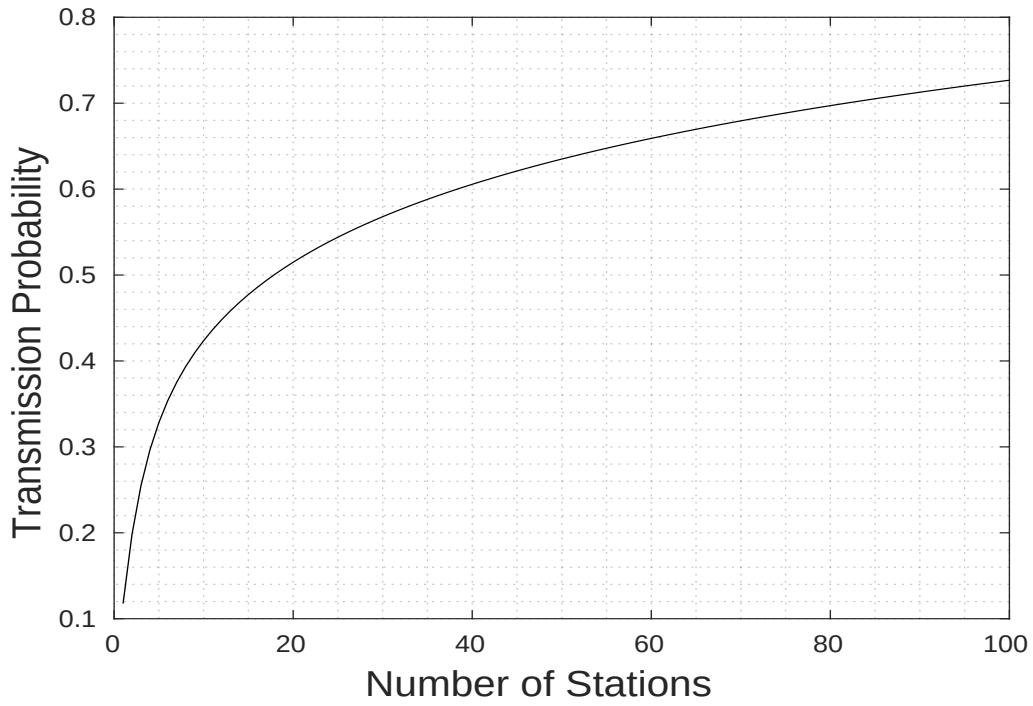

Figure 2.18: Bianchi Station Transmission Probability Results 802.11 FHSS

The parameter P_s provides the probability that a transmitted DCF frame will be successfully received at the sink node and is defined as:

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} \tag{2.21}$$

The probability is used in both numerator and denominator of Equation 2.6 above to calculate the saturation throughput of the DCF and is evaluated for the FHSS PHY in Figure 2.19.

As expected the model output decreases non-linearly as the number of competing stations present in the network is increased. At n = 1 the probability is equal to 1 as there are no other nodes in the network for the transmitted frame to collide with. As the applied load is gradually increased towards the maximum of n = 100, the output decreases non-linearly to a minimum value of 0.491. The overall mean success probability for all applied loads considered in the evaluation is 0.6136.

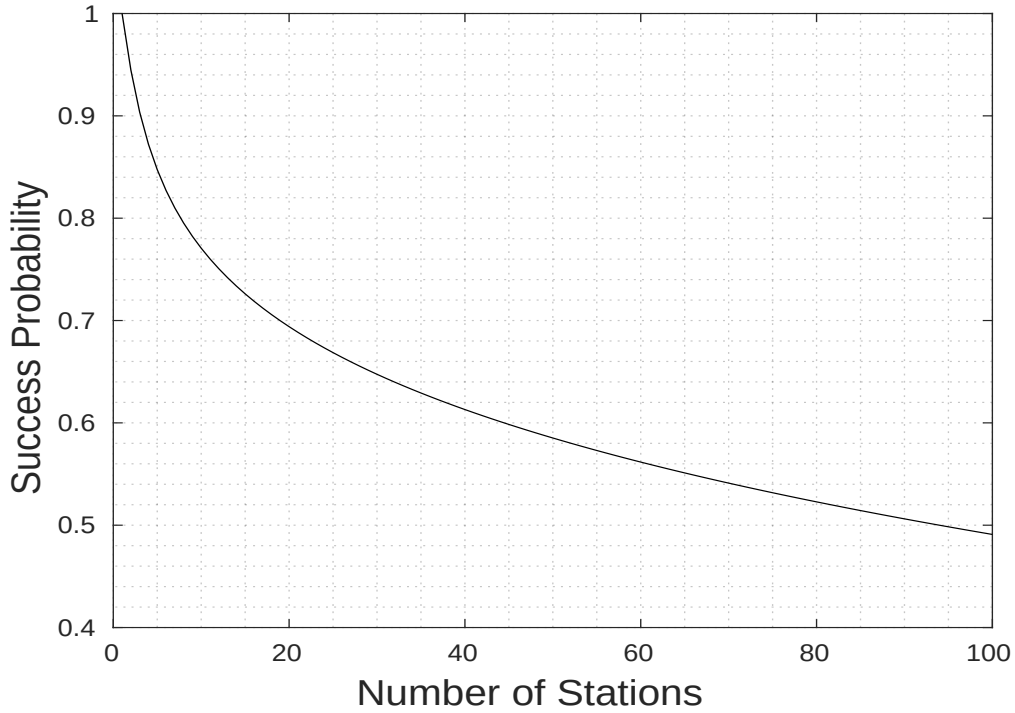

Figure 2.19: Bianchi Success Probability Results 802.11 FHSS

2.3.2 Example Chatzimisios Delay Evaluation

Next, in order to illustrate the output of the delay extensions to the Bianchi model provided by Chatzimisios, the main modelling expressions are evaluated for the DSSS PHY as per the work of the author in [1].

2.3.2.1 Input Parameters

The DCF parameters used in the Chatzimisios evaluation are summarised in Table 2.4 and are taken directly from the IEEE 802.11b standard in [4]. All PHY settings were chosen to exactly match that used by Chatzimisios in [1] in order to allow a direct comparison with the results printed. These parameters include for example the payload size of 8184 bits and the PHY transmission rate of 1 Mbps, which assumes the use of BPSK modulation.

Parameter	**Value**
Number of Nodes (N)	100
Slot Time (μs)	20
CWmin (slots)	32
CWmax (slots)	1023
Maximum Backoffs	5
Payload Size	8184
MAC Header Size (bits)	224
PHY Header Size (bits)	192
ACK Size (bits)	112
SIFS Duration (μs)	10
DIFS Duration (μs)	50
Propagation Time (μs)	1
Data Rate (Mbps)	1

Table 2.4: Parameters used in Chatzimisios Evaluation of 802.11b DSSS

2.3.2.2 Applied Load Tests

The applied load scenario used for the evaluation of the DSSS PHY is illustrated in Figure 2.20 below. As per the Bianchi FHSS test the applied channel load is incremented from n = 1 to n = 100 competing STAs and at every loop iteration the expected access delay of the DCF is numerically evaluated using the Chatzimisios expression. Again the model output from the test is therefore an N-length vector providing the DCF access delay as a function of the applied channel load under the assumption of saturated arrival traffic.

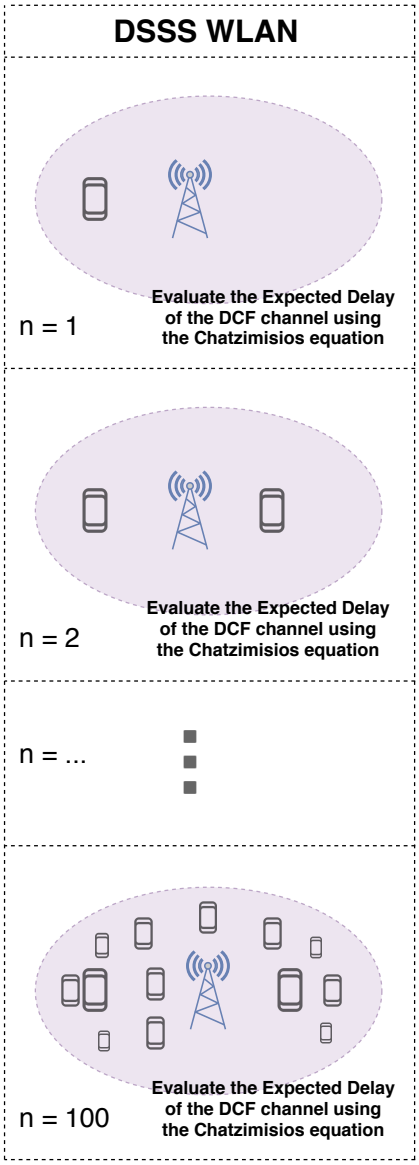

Figure 2.20: DSSS Test Applied Channel Load

2.3.2.3 Expected Access Delay

The expected DSSS channel access delay as per the Chatzimisios formula in Equation 2.9 is shown in Figure 2.21 below. Defined as the length of time between a frame arriving at the head of the source node transmission queue and the frame being successfully received at the sink, the overall mean delay for the evaluation was 810380 μs. As seen from the graph the output of the Chatzimisios delay model increased *almost* linearly from a minimum of value of 9341 μs at n = 1 to a maximum of 1486000 μs at the heaviest considered load of n = 100.

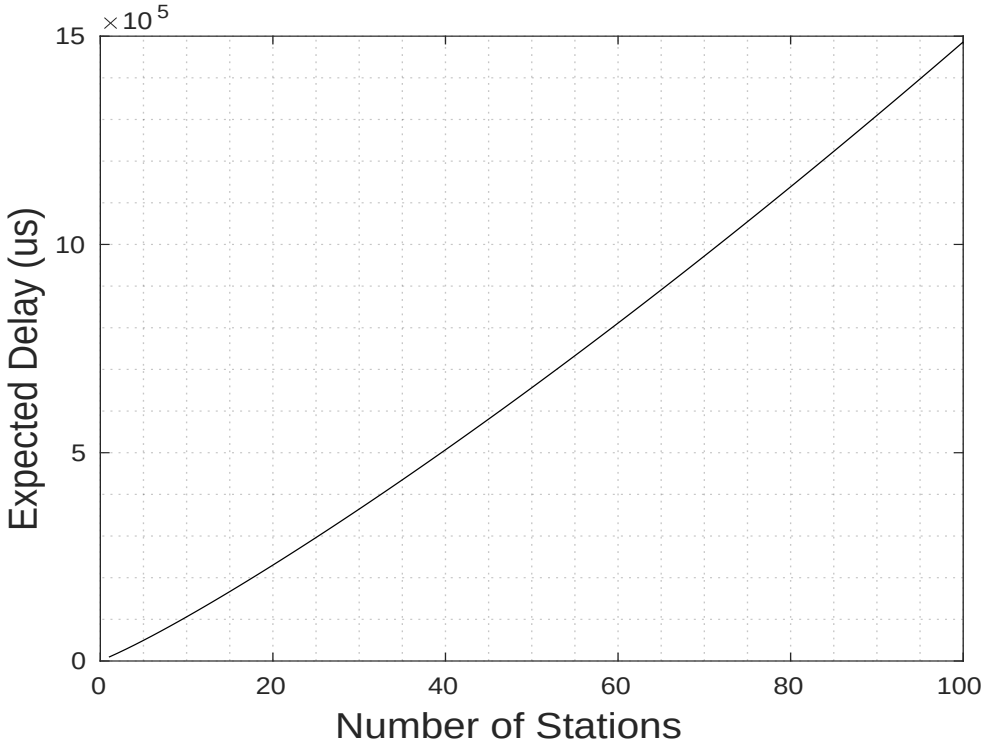

Figure 2.21: Chatzimisios Expected Delay Results 802.11b DSSS

Also of interest are the constituent terms of the Chatzimisios delay expression $E[S]$ and $E[X]$ shown below and on the following page. The parameter $E[X]$ represents the total number of time slots taken on average to send a successfully received frame and is evaluated for the DSSS PHY in Figure 2.22 below. As seen the model output increases linearly from a minimum of 10.5 slots at n = 1 to a maximum of 155.2 slots at the heaviest considered load of n = 100 with an overall mean value of 81.691 slots.

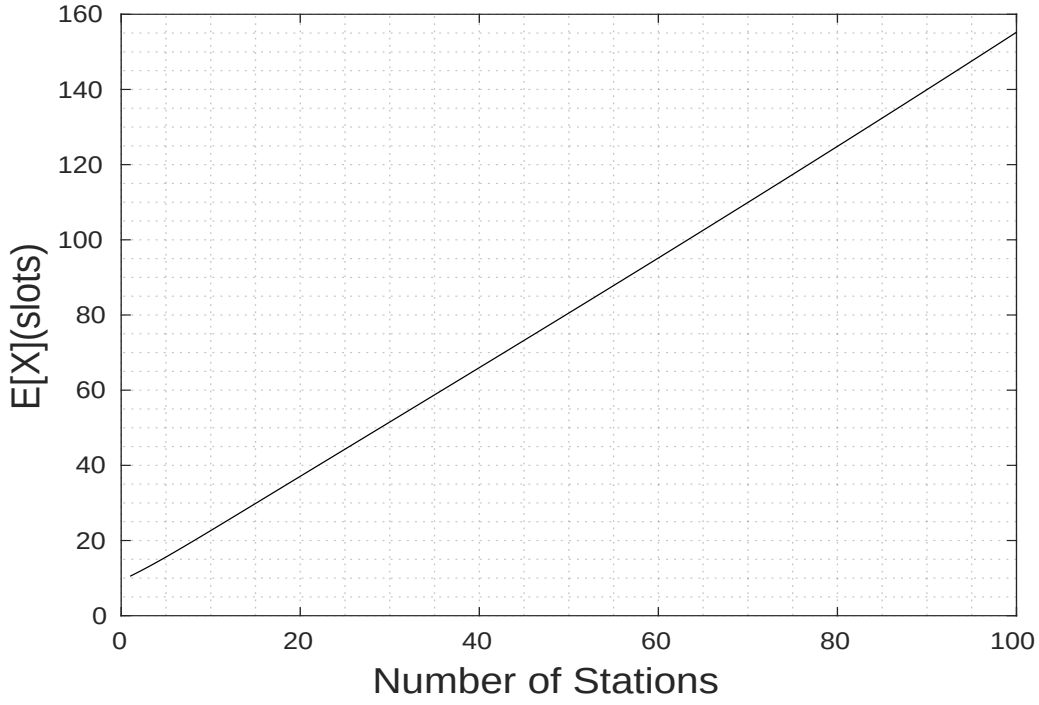

Figure 2.22: Chatzimisios E[X] Results 802.11b DSSS

The parameter $E[S]$ represents the average length of a slot time and is the same value as used in the Bianchi saturation throughput formula. The expression is evaluated for the DSSS PHY in Figure 2.23 below. As seen the model output increases exponentially from a minimum of 517 μs at n = 1 to a maximum value of 5500 at the heaviest considered load of n = 100. The overall mean value of $E[S]$ for all applied loads considered in the evaluation was 4331.81 μs.

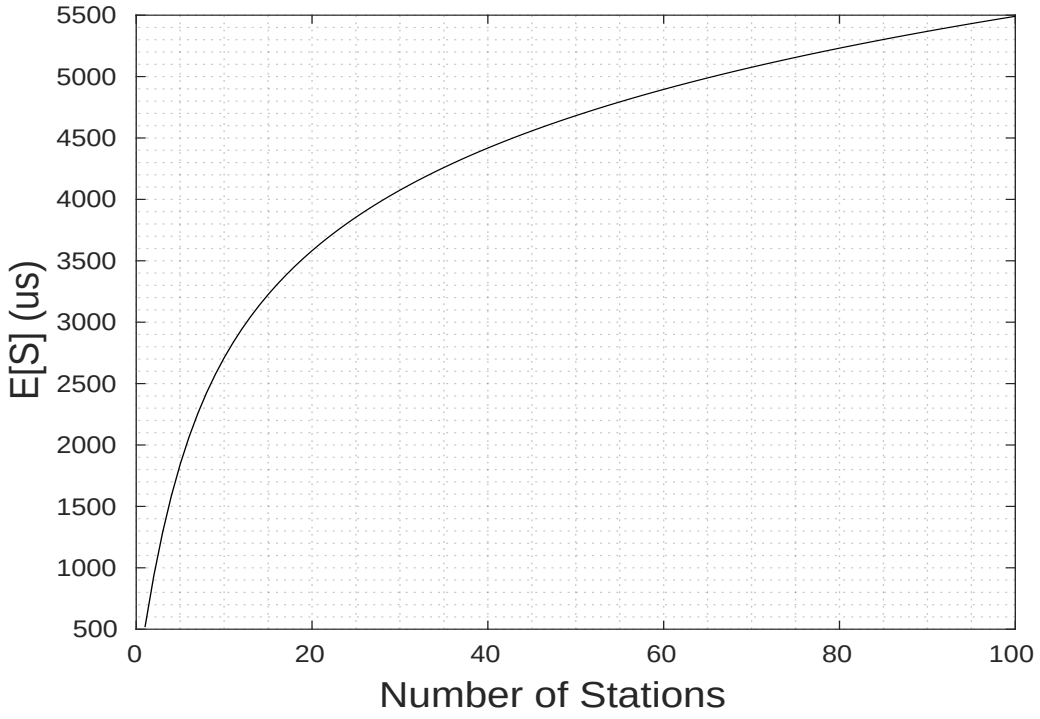

Figure 2.23: Chatzimisios E[S] Results 802.11b DSSS

2.3.3 Comparison of 802.11ac vs 802.11ah DCF Performance

This section analyses and compares the MAC layer performance between traditional 2.4 GHz 802.11ac and sub-1 GHz 802.11ah designed for wireless sensor networks and the Internet of Things. More specifically the DCF channel throughput, delay, and conditional collision probability are numerically evaluated and discussed for both standards using the analytical modelling provided by Bianchi and Chatzimisios.

Note that the analysis presented is intended as an illustration of the performance of the two 802.11 types used in the interface bonding study in Chapter 3 and Chapter 4, and not as a full root and branch performance comparison in its own right.

2.3.3.1 802.11ah Use Cases

Published over two decades ago, the 2.4 GHz 802.11 standard was originally designed for home and office to connect laptops and desktop with the killer application being Internet access. By contrast [27], the recent IEEE 802.11ah amendment was designed specifically for the following use cases:

- **Smart Sensors & Meters:** The main goal of the 11ac amendment is to enable 802.11 to accommodate IoT devices covering a range of user applications including both indoor and outdoor use in urban, suburban, and rural environments.
- **Backhaul Aggregation:** This is where a 802.11ah router or gateway aggregates the operational data collected from sensor devices and forwards it to a control application or database via a backhaul connection in order to improve efficiency.
- **Cellular Offloading & Extending Hotspot Range:** The high throughput and long range of the sub-1 GHz bands used by the 11ah standard are ideal for extending Wi-Fi hotspot range and offloading data and voice traffic from cellular networks such as LTE.

2.3.3.2 Technical Challenges for 802.11 in IoT

The main technical challenges faced by 802.11ah when extending the standard to accommodate IoT application requirements include:

- **Increased Range of Coverage:** Certain long-range IoT applications require greater than 1 km of wireless coverage. In addition to the increase in range afforded by the use of longer wavelengths in the sub-1 GHz bands, in order to help meet this requirement 1 MHz channel bandwidths and a new MCS scheme (MCS10) are introduced. Narrow bandwidths provide a larger symbol duration that is most robust towards inter-symbol interference as seen in most outdoor and long-range scenarios. Multi-hop operation with mesh nodes and relays is also supported.

- **Large Numbers of Connected Devices:** IoT and wireless sensor networks are characterised by a large number of connected devices which leads to increased channel contention and a high number of frame collisions. To mitigate this effect 11ah defines an optional channel access mechanism called Restricted Access Window (RAW). In simple terms the mechanism reduces collisions by dividing stations into different groups and restricting access during a particular period of time. Furthermore 11ah supports up to 8191 active stations per Access Point (AP) using a novel hierarchical structure for the Association Identifiers representing each connection.

- **Spectral & Temporal Resources:** Most medium range wireless technologies such as 802.11 operate in the crowded 2.4 GHz ISM band. With the introduction of billions of IoT devices this portion of the spectrum is now at risk of becoming exhausted and therefore the 802.11ah standard turns to the less contentious sub-1 GHz band where the detrimental effects of co-channel interference are significantly reduced.

- **Low Power Consumption Requirement:** Many IoT devices are battery powered and require weeks, months, or even years of continuous use. Therefore the efficiency of the power consumption becomes a critical consideration with the embedded NIC representing a large portion of that use. The original 802.11 version supports an idle time of up to 18.64 hours, but 11ah extends this to allow multiple idle periods up to one year in length. Several other new features of 11ah also help to increase battery life, for example shorter frame headers and implicit frame acknowledgements in certain cases.

2.3.3.3 802.11ah Improvements at the MAC Layer

In addition to those mentioned above, the other new MAC layer features of note in the 802.11ah standard are described below:

- **Bi-directional Transmission Opportunities:** In order to extend device battery life the standard provides a method of bi-directional TXOPs allowing the transmission and receipt of multiple MAC layer frames in a single TXOP.
- **Block Frame Acknowledgements:** A block acknowledgement procedure is introduced wherein fragments obtained from partitioning a MAC Service Data Unit (MSDU) can be acknowledged either immediately using an NDP Block ACK frame, or en masse using the regular block ACK mechanism.
- **Sub-Channel Selective Transmission:** The Sub-Channel Selective Transmission feature allows wireless stations to rapidly switch through different channels from one frame transmission to the next in order to avoid the fading that occurs over the individual narrow-band sub-channels.
- **Delivery Traffic Indication Map:** The Delivery Traffic Indication Map introduced in 802.11ah informs sleeping wireless stations when multicast traffic destined for the node is buffered at the AP by means of a bit map sent via the MAC beacon.

2.3.3.4 Input Parameters

The DCF and PHY parameters used in the following evaluation were taken directly from the relevant standards in [12] and [13]. The 11ac DCF parameters are shown in Table 2.5 and the 11ac parameters in Table 2.6. The respective Modulation and Coding Scheme (MCS) settings for each type were chosen to provide an identical raw PHY data and therefore highlight the difference at the MAC layer. The MCS assumed for 802.11ac was Index = 6, which uses a 20 MHz channel bandwidth, a single spatial stream, 64-QAM modulation, a 3/4 coding rate, and a 400 ns guard interval, to give a raw PHY transmission rate of 65 Mbps. The MCS for 802.11ah was Index = 7, which uses a 9 MHz bandwidth, two spatial streams, 64-QAM modulation, a 3/4 coding rate, and a 4 μs guard interval to also produce a raw PHY transmission rate of 65 Mbps.

DCF/PHY Parameter	**Parameter Value**
Number of Nodes (N)	100
Slot Time (μs)	9
CWmin (slots)	15
CWmax (slots)	1023
Payload Size	8184
MAC Header Size (bits)	288
PHY Header Size (bits)	192
ACK Size (bits)	112
SIFS Duration (μs)	16
DIFS Duration (μs)	34
Propagation Time (μs)	1
Data Rate (Mbps)	65

Table 2.5: DCF Parameters used in Simulation Validation for 802.11ac

DCF/PHY Parameter	**Parameter Value**
Number of Nodes (N)	100
Slot Time (μs)	52
CWmin (slots)	15
CWmax (slots)	1023
Payload Size	8184
MAC Header Size (bits)	288
PHY Header Size (bits)	192
ACK Size (bits)	112
SIFS Duration (μs)	160
DIFS Duration (μs)	264
Propagation Time (μs)	6
Data Rate (Mbps)	65

Table 2.6: DCF Parameters used in Simulation Validation for 802.11ah

2.3.3.5 Saturation Throughput Comparison

The achieved saturation throughputs for 11ac and 11ah are compared using the Bianchi model in Figure 2.24 below. The 11ac throughput is plotted as the dotted and dashed black line and the 11ah throughput as the dashed red line. As seen the 802.11ac channel throughput is considerably higher than 11ah for all applied loads considered with an overall mean percentage difference of 23.74 %.

For 802.11ac the achieved channel throughput increased from a minimum value of 42.75 Mbps at the initial load of n = 1 to a maximum of 43.69 Mbps at n = 2 and then decreased gradually to an overall test minimum of 21.5 Mbps as the applied load was raised towards the heaviest considered level of n = 100. The overall mean channel throughput for all applied loads considered in the test was 29.9211 Mbps.

By comparison for 11ah the achieved channel throughput was considerably lower and increased from a minimum of 15.48 Mbps at the initial load of n = 1 to a maximum of 18 Mbps at n = 4. The output then decreased non-linearly to an overall minimum of 10.93 Mbps as the applied load was raised towards the heaviest considered level of n = 100. The overall mean 11ah throughput for all applied loads considered was 13.8745 Mbps.

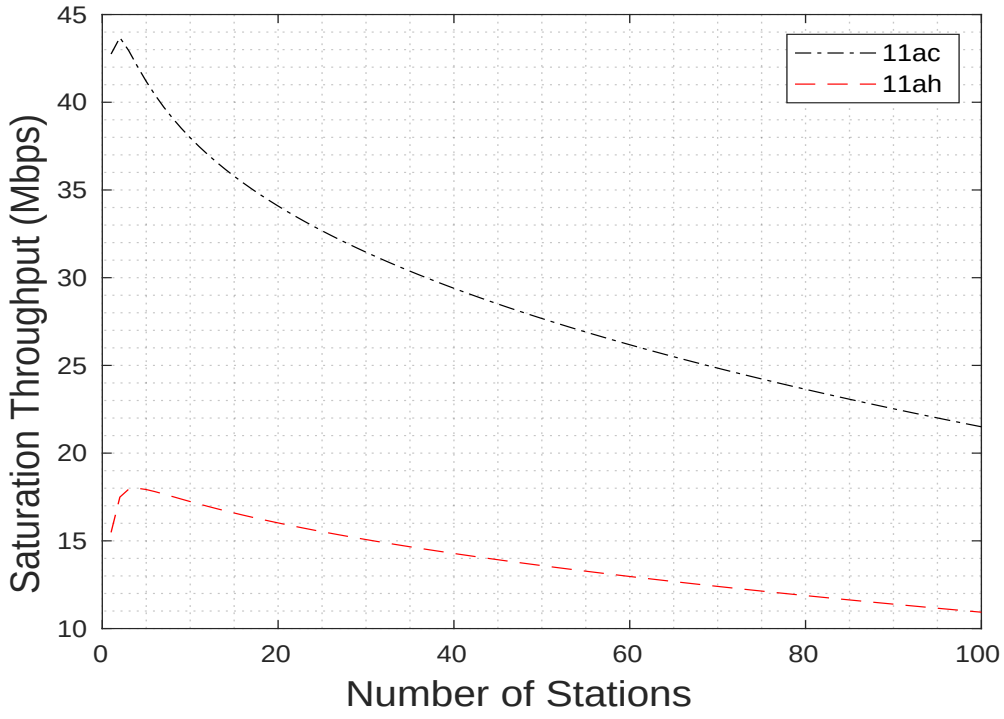

Figure 2.24: 802.11ac vs 802.11ah Saturation Throughput Comparison

2.3.3.6 Expected Access Delay Comparison

The expected channel access delays for 11ac and 11ah are presented in Figure 2.25 below as a function of the applied channel load. The 11ac delay is plotted as the dotted and dashed black line and the 11ah delay as the dashed red line. As seen the achieved 11ah access delay was considerably higher throughout when compared to 11ac with an overall mean percentage difference of 67.14%.

For 11ac the delay increased non-linearly from 383.2 μs at the initial load of n = 1 to a maximum of 76200 μs at the heaviest considered load of n = 100, with an overal mean value of 32735 μs. By comparison for 11ah the delay output increased from 1058 μs at n = 1 to a maximum of 149900 μs at the heaviest test load of n = 100, with a mean value of 65822 μs.

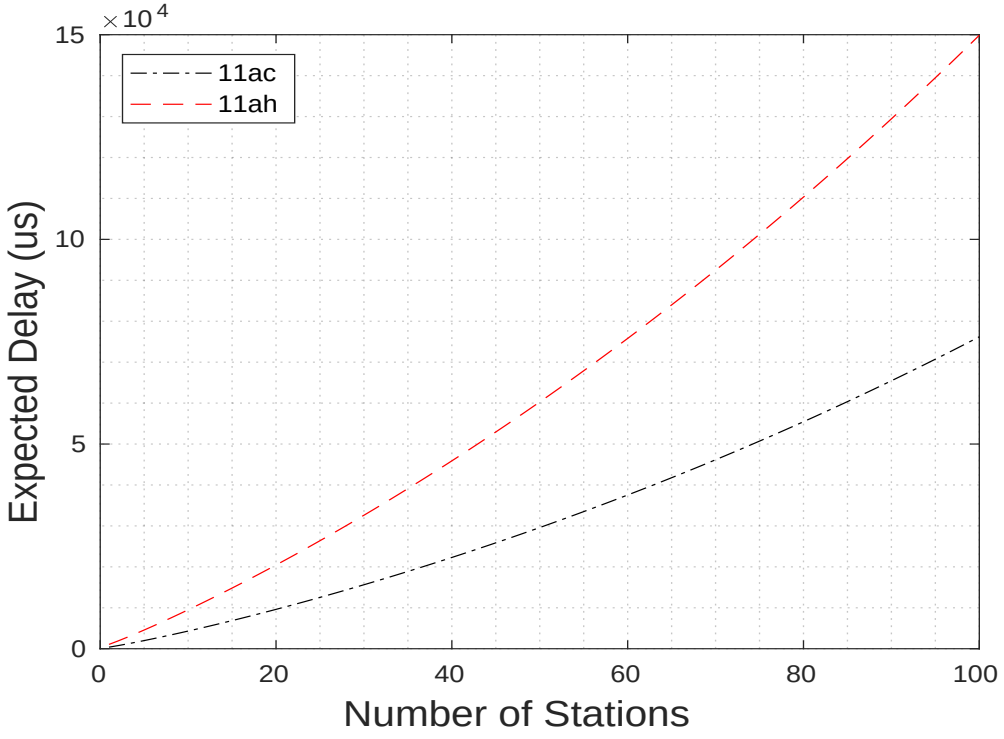

Figure 2.25: 802.11ac vs 802.11ah Expected Access Delay Comparison

2.3.3.7 Conditional Collision Probability Comparison

The conditional collision probabilities for 11ac and 11ah are compared in Figure 2.26 below. The 11ac collision probability is plotted as the dotted and dashed black line and the 11ah collision probability as the dashed red line. Notably for both 11ac and 11ah the collision probability was identical throughout the test. Both outputs increased exponentially from a value of zero at the initial load of n = 1 to a maximum of 0.7962 at the heaviest test load of n = 100. The overall mean collision probability for all applied loads considered was 0.6561.

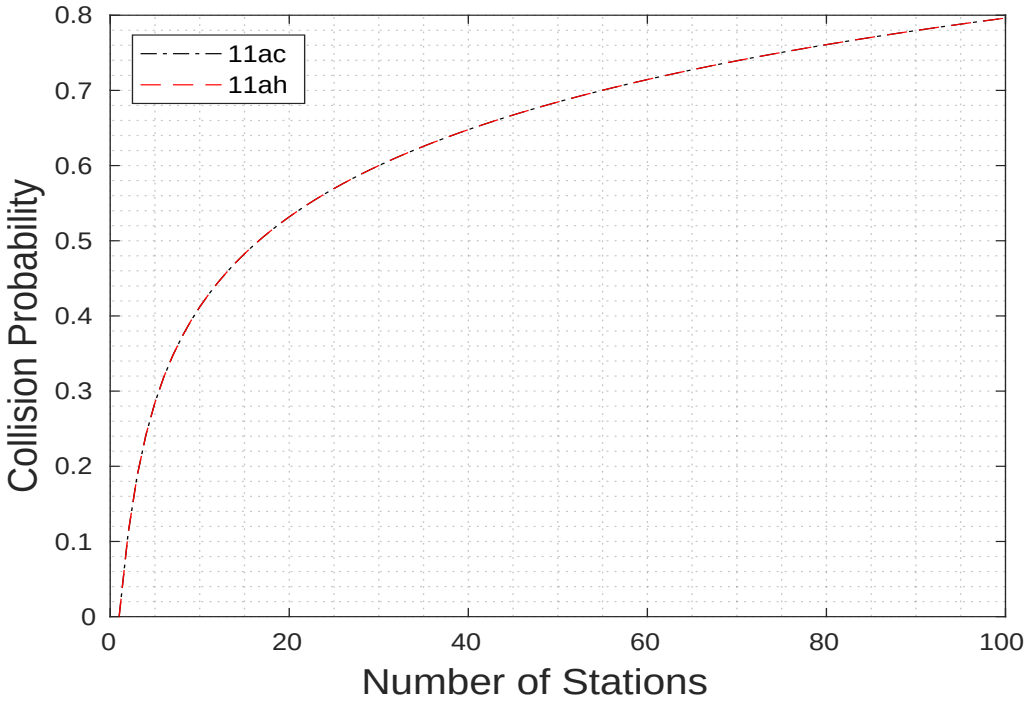

Figure 2.26: 802.11ac vs 802.11ah Conditional Collision Probability Comparison

2.4 Simulating the 802.11 MAC Layer in MatLab

Towards the overarching thesis research goal this section implements a novel evaluation framework for the 802.11 DCF using the MatLab programming language and verifies the correct operation by comparison against the well established and widely accepted analytical models by the authors Bianchi and Chatzimisios.

The simulator output provides the DCF channel throughput, expected access delay, and conditional collision probability as a function of the number of competing wireless stations under the assumption of saturated arrival traffic.

The implemented framework plays an important role in research presented later in the thesis and is extended in Chapter 3 to evaluate and compare the aggregated throughput, link bandwidth utilisation, and delay of statically configured homogeneous 11ac-11ac and heterogeneous 11ac-11ah interface bonds under different combinations of applied link load. To help answer the main research questions of the thesis the framework is then further extended in Chapter 4 to allow a rigorous performance analysis of state-of-the-art *dynamic* link selection algorithms.

2.4.0.1 Benefits of Software Simulation

The benefits of the implemented framework verses a hardware-based evaluation testbed are:

1. Flexibility & Control: The framework provides a flexible method of evaluating the DCF performance that can be adapted to accommodate any 802.11 technology by swapping DCF input parameters (such as slot length and DIFS) as per the relevant specification. Other parameters such as the PHY data rate and payload size can easily be changed to suit the desired experimental requirements and the framework also gives the ability to adopt non 802.11 compliant behaviour for research purposes without impacting adjacent protocol layers or other software components.

2. Speed & Cost: To experimentally evaluate the DCF performance for the IoT network population sizes considered in the main thesis research in Chapter 4 would be prohibitively expensive in hardware. The software simulation also provides quicker execution, allowing multiple tests to be run and re-run in a fraction of the time required for hardware.

3. MAC Sub-Layer Focus: The framework only simulates the 802.11 MAC sub-layer and PHY layer considerations such as signal strength and propagation time are not evaluated, and neither is the adjacent Logical Link Control Layer. By measuring the link throughput directly at the MAC layer we obtain the best-case scenario for the 802.11 performance based solely on the channel access procedure. Later in Chapter 3 this clear focus helps reveal the impact of the DCF mechanism on the 802.11 interface bond throughput performance under a range of applied load conditions.

2.4.1 Simulation Implementation

This section presents the main assumptions, algorithms, and input-output data used in the MatLab-based DCF evaluation framework implemented as part of the thesis research.

The interpreted MatLab language was chosen for the implementation because of its expressive power and wide range of available mathematical functions. A compiled language such as C++ could have provided greater speed with the potential for real-time execution and integration with other simulations/emulators such as NS3 and Mininet, but for the primarily analytical research presented in this thesis MatLab was considered as the appropriate choice.

2.4.1.1 Simulation Assumptions

The list of modelling assumptions made during the implementation of the DCF simulator is provided below. These assumptions were made in order to closely align with those of the Markov-based models presented by Bianchi and Chatzimisios in [30] and [1] to allow a direct comparison and verification between the simulator results and those of the established and widely accepted 802.11 analyses.

2.4.1.1.1 802.11 Network Topology In the implemented simulator a flat single-hop 802.11 wireless topology is assumed such that all competing DCF stations are within carrier sensing range of one another with no hidden or exposed nodes present. In addition the RTS-CTS and NAV mechanisms are not used. This fully connected topology is illustrated in the diagram in Figure 2.27 below.

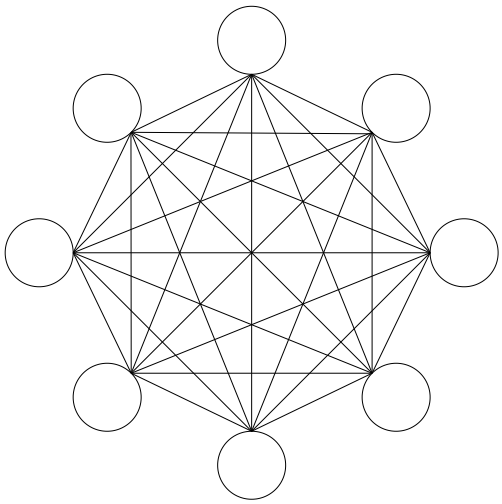

Figure 2.27: 802.11 WLAN Topology Used in DCF Framework

2.4.1.1.2 Traffic Arrival Discipline As per the original Bianchi work in [30] saturated arrival traffic conditions are assumed such that all nodes in the network always have a frame ready to send in their transmit buffer. While this approach cannot model more realistic per-user traffic arrival rates it provides a good approximation of very heavily loaded traffic conditions while greatly simplifying the implementation.

2.4.1.1.3 Frame Re-transmissions As per Bianchi an infinite number of frame re-transmission attempts are assumed, i.e. a node does not drop a frame after a certain number of failed transmission attempts set to $k = 5$ in most versions of the 802.11 standard. This is a fundamental and well-documented limitation of the single-link Bianchi model and the Chatzimisios delay extensions which the implemented simulator uses as a reference.

2.4.1.1.4 Ideal Radio Conditions Again as per the Bianchi model an important assumption is that the only source of frame loss considered in the simulations are due to collisions occurring on the DCF channel, i.e. other losses due to radio frequency factors such as slow and fast fading are ignored.

2.4.1.1.5 Collision Decoupling Following Bianchi the frame transmission-collision process is *not* dependent on the past transmission history of the node. Intuitively this assumption becomes more accurate as the contention window size and number of competing stations become larger.

2.4.1.1.6 Aggregation and Fragmentation The techniques of aggregating and fragmenting multiple individual frames are not considered in the proposed simulation. All packets received from the upper protocol layer are assumed to be sent as is directly over the DCF channel.

2.4.1.1.7 Quality of Service The individual traffic priorities and Quality of Service (QoS) mechanisms including virtual collisions as per 802.11e are not modelled in the implemented simulation, i.e. all test traffic belongs to a single undifferentiated class.

2.4.1.2 Simulation Input Parameters

The implemented DCF simulator takes as input the various parameters summarised in Table 2.7 below. Values such as the slot length and inter-frame spacing (DIFS/SIFS) are specific to a particular version of the standard, for example 802.11ac or 802.11ah. The global simulation parameters include the total number of competing stations denoted by N, the grand total number of frames successfully transmitted by all competing stations denoted by M, and the DCF frame payload size in bits which is denoted by P.

Parameter	**Description**
N	Number of contending DCF stations present in the network
M	Group total number of DCF frames transmitted by all nodes
Slot	DCF slot length given in units of micro seconds
CWmin	Minimum contention window size given in numbers of slots
DIFS	Duration of the Distributed Inter-Frame Spacing in microseconds
SIFS	Duration of the Short Inter-Frame Spacing in microseconds
MaxBackoff	Maximum number of back-offs or increments of the binary exponent
Payload	DCF frame payload size in bits
MacH	Size of the 802.11 MAC header given in bits
PhyH	Size of the 802.11 PHY header given in bits
Ack	Size of the DCF ACK frame given in bits
DataRate	Raw PHY transmission rate given in Mbps
Prop	PHY layer propagation time given in micro seconds

Table 2.7: MatLab 802.11 DCF Simulator Input Parameters

2.4.1.3 Simulation Output Data

The main simulation output data consists of the frame collision likelihood, expected access delay, and the saturated throughput of the DCF channel for a given 802.11 standard. In addition the same statistics are also provided for a single 'tagged' DCF node. All output vectors are of size N and provide the simulation results for each performance metric as a function of the applied load given as the number of competing DCF stations present in the network under the assumption of saturated arrival traffic. In addition to the above vector outputs their overall mean values for n = 1 to n = N are also given for both the DCF channel and the tagged DCF node.

The simulation output data and notation used is summarised in Table 2.8 below. Note the use of the underline to differentiate between the DCF channel statistics and those for the tagged DCF node. Also note the use of the bar symbol to distinguish the overall mean values calculated from the above vector quantities.

Parameter	**Description**
P_c	Collision probability of the DCF channel (vector)
D	Access delay for the DCF channel (vector)
S_n	Saturation throughput for the DCF channel (vector)
$\underline{P_c}$	Collision probability for the tagged DCF node (vector)
$\underline{D}$	Access delay for the tagged DCF node (vector)
$\underline{S_n}$	Saturation throughput for the tagged DCF node (vector)
$\bar{P_c}$	Collision probability of the DCF channel (mean)
$\bar{D}$	Access delay for the DCF channel (mean)
$\bar{S_n}$	Saturation throughput for the DCF channel (mean)
$\underline{\bar{P_c}}$	Collision probability for the tagged DCF node (mean)
$\underline{\bar{D}}$	Access delay for the tagged DCF node (mean)
$\underline{\bar{S_n}}$	Saturation throughput for the tagged DCF node (mean)

Table 2.8: MatLab 802.11 DCF Simulator Output Data

2.4.1.4 DCF Algorithm Implementation

Next the implementation of the DCF algorithm used in the MatLab-based simulation is described in detail. The pseudocode for the implementation is shown in Algorithm 1 on the page opposite in a verbose and easy to read format.

The central DCF algorithm is contained within an outer while loop which keeps track of the number of transmitted frames and exits after a total of M frames are successfully transmitted onto the channel. The transmission attempt times for each of the N competing stations present are maintained in an N-sized vector with another vector used to keep track of the current back-off stage for each node. First the back-off counter for each node is initialised to a value of zero as per the DCF algorithm defined in the 802.11 specification. Next the time of the first transmission attempt for each node is calculated using the following formula: $T_1 = DIFS + (CWmin * rand) + SLOT$, where *rand* is the output of a native MatLab function that generates a pseudo random number between 0 and 1.

Once the vector of initial transmission attempt times for all competing stations has been calculated the node/s with the earliest transmission time is found. If only a single node is found with the shortest transmission attempt time then the frame is automatically deemed to have been successfully transmitted onto the channel. A note is made of the delay of the winning node, its back-off counter reset to zero, and the node then calculates a new random waiting time used when sending the next frame in its transmission queue. Note that all other non-winning nodes resume contention using their existing waiting times as per the DCF algorithm.

If more than one node happens to choose the exact same randomly generated transmission time then both/all nodes will attempt to access the shared channel simultaneously and a collision will occur. When this happens each node experiencing a collision increments its back-off counter and then re-contends for channel access using a doubled contention window size.

After a total of M frames are successfully transmitted on the channel the number of competing nodes in the network is incremented and the entire outer while loop is repeated. The end result is multiple output vectors of length N providing the collision probability, mean delay, and saturation throughput of the DCF channel over the range of contending stations considered from 1 to N.

Algorithm 1: Main DCF Algorithm Simulation Pseudocode

```
Function SimulateDCF()
    Initialise all counters/trackers to zero
    For all contending stations in the network
        Initialise node back-off stage counter to zero
        Calculate initial random waiting time for node
        While the number of successful frame transmissions is less than the maximum
            Find the node/s with the shortest random waiting time
            Check whether more than one node has the shortest random waiting time
                If no, the frame transmission of the winning node is successful
                        Increment the number of successful transmissions counter
                        For the winning node
                                Calculate the delay incurred due to the contention process
                                Reset node back-off counter to zero
                                Calculate the new node random waiting time
                        If total successful frame transmissions is equal to the maximum
                                Calculate the total simulation time
                        For all other nodes
                                Wait until current tx completes and resume contention
                If yes then a collision has occurred
                        For each colliding node
                                Increment the number of failed transmissions
                                Calculate a new random waiting time
        Calculate collision likelihood, saturation throughput, and expected delay
End
```

2.4.1.5 DCF Simulation Operation

The diagram in Figure 2.28 below illustrates the operation of the implemented DCF simulation for a single node and the successful transmission of a single frame.

At time zero the tagged node already has a frame in its transmission queue ready to send due to the saturated traffic assumption and it calculates an initial random waiting time before making its first attempt to access the shared physical medium. The initial waiting time is calculated as above using $DIFS + (2^0 * CWmin * rand) + SLOT$.

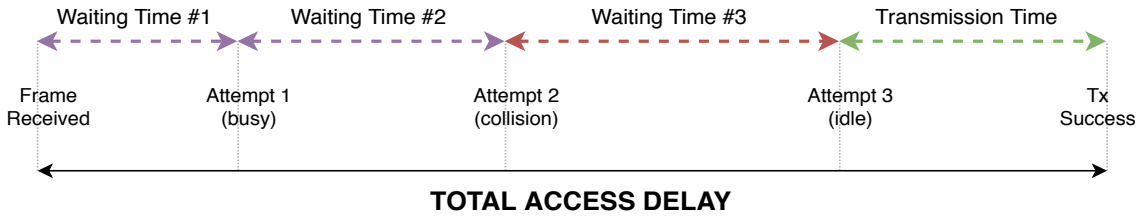

Figure 2.28: Illustration of MatLab DCF Simulation

On the first attempt once the random time timer has elapsed the tagged node senses the medium to be busy and it reacts by choosing a new random waiting timer from the same sized range and then restarts the contention process. At the second transmission attempt the node senses that the medium is idle and, believing that it has won sole access to the channel, it attempts to transmit its frame only for it to collide with the simultaneous transmission attempt of another station. Inferring the collision by lack of ACK from the sink the tagged node then backs off by incrementing its binary exponent and re-contends for the channel using a waiting time drawn uniformly from the now doubled range of $DIFS + (2^i * CWmin * rand) + SLOT$, where i denotes the current back-off stage of the tagged node. On the third attempt the node wins contention outright and successfully transmits its frame.

The length of time taken is calculated as: $MACH + PHYH + P + SIFS + ACK + DIFS$, where MACH, PHYH, and ACK are the respective times in μs taken to transmit the required control and data frames. The total access delay experienced by the frame is therefore equal to the sum of the waiting times between transmission attempts and the time taken to transmit the frame once the node has gained channel access. In this simple example the collision likelihood of the transmitted frame is calculated as $\frac{1}{3}$.

The single frame contention process illustrated is repeated until the specified M number of frames have been successfully transmitted onto the DCF channel. This entire process is then repeated N number of times for the total number of competing stations present from 1 to N to produce the N-sized simulation output vectors described in Table 2.8 above.

2.4.1.6 Calculation of DCF Performance Metrics

In the implemented simulator the saturation throughput, expected access delay, and conditional collision probability are calculated for the overall DCF channel and tagged STA node as follows.

2.4.1.6.1 Channel Saturation Throughput The saturation throughput of the overall DCF channel is calculated for n by counting the total number of successfully transmitted payload bits and then dividing this number by the total simulation time for all M successful transmitted frames:

$$S_n = \frac{Total\,Payload\,Bits\,Transmitted}{TotalSimulationTime} \tag{2.22}$$

2.4.1.6.2 Expected Access Delay The expected access delay of the DCF channel is calculated by counting the delay experienced by each successfully transmitted frame and summing over the total number of M successful frames sent over the shared DCF channel. The summed value is then divided by M to provide the overall mean delay for the simulation as shown below:

$$D_n = \frac{\sum_{m=1}^{M} D_m}{M} \tag{2.23}$$

2.4.1.6.3 Conditional Collision Probability The conditional collision probability is calculated for n by counting the number of successful and failed frame transmissions events and then dividing the number of failed transmissions by the sum of the total number of both successful and failed transmissions:

$$P_n = \frac{Total\,Frame\,Collisions}{Total\,Frame\,Collisions + Total\,Successful\,Frames} \tag{2.24}$$

2.4.1.6.4 Tagged Node Performance To calculate the collision probability, expected delay, and saturation throughput of an individual DCF node the first in the N-length array of stations is chosen as the 'tagged' STA under study. Separate counters are implemented for the number of failed frame transmissions and successful frame transmissions for the tagged node which are then used to calculate the collision probability using the same formula as for the channel collision probability described above. Similarly for the delay of the tagged node the individual frame delays experienced by that node are summed and then divided by the total number of successful frame transmissions for the node. The saturation throughput is similarly calculated by counting the number of bits transmitted by the individual node and then dividing that number by the total simulation time in μs

2.4.2 Simulation Validation

In the following sections we verify the implemented simulator by comparison against the widely accepted analytical models of the 802.11 DCF by the authors Bianchi and Chatzimisios.

2.4.2.1 Validation Methodology

The principle technique used to ensure the operational validity of the simulator is a direct comparison against two existing and proven analytical models of the 802.11 DCF. The reference models used were the Bianchi and Chatzimisios works [1, 30] described in Section 2.3 previously. The validation methods used include a combination of objective confidence intervals and subjective graphical data comparison.

First we validate the implemented DCF simulator using a qualitative graphical comparison between the output and the analytical modeling provided by Bianchi and Chatzimisios. For the quantitative analysis, based on the technique presented in [145] and also used in [19] and [15], the different stages of the validation process are summarised below:

1. From the available simuland data, the model response variables selected for the validation were the DCF channel throughput, expected access delay, and conditional collision probability.

2. A total number of eight iterations was chosen for the validation, which was considered sufficient due to the high number of 2M frames sent per sub-test.

3. The model was then executed eight times and the above response variables were measured to provide the required samples.

4. The sample mean and standard deviation were then calculated for each of the above response variables.

5. Using the z-distribution the 99.9% confidence intervals were then calculated for each response variable to provide a high level of statistical certainty.

6. Finally, using the known values for each response variable from the proven Bianchi and Chatzimisios models, it was then determined whether these fell within the desired 99.9% confidence interval.

2.4.2.2 Input Parameters

The simulation setup and input parameters used in the following validation are identical to those used for the previous 11ac vs 11ah performance comparison and are presented in Table 2.5 and Table 2.5 of Section 2.3.3.4 above.

2.4.2.3 Applied Load Tests

The applied loads used in the validation tests are described below and on the following pages.

2.4.2.3.1 802.11ac Validation Test For the 11ac validation the applied load is incremented from n = 1 to n = 100 competing stations. At each iteration a total M number of frames are transmitted and the long-term DCF channel and tagged node throughput, delay, and conditional collision probability are calculated and compared against the Bianchi and Chatzimisios output.

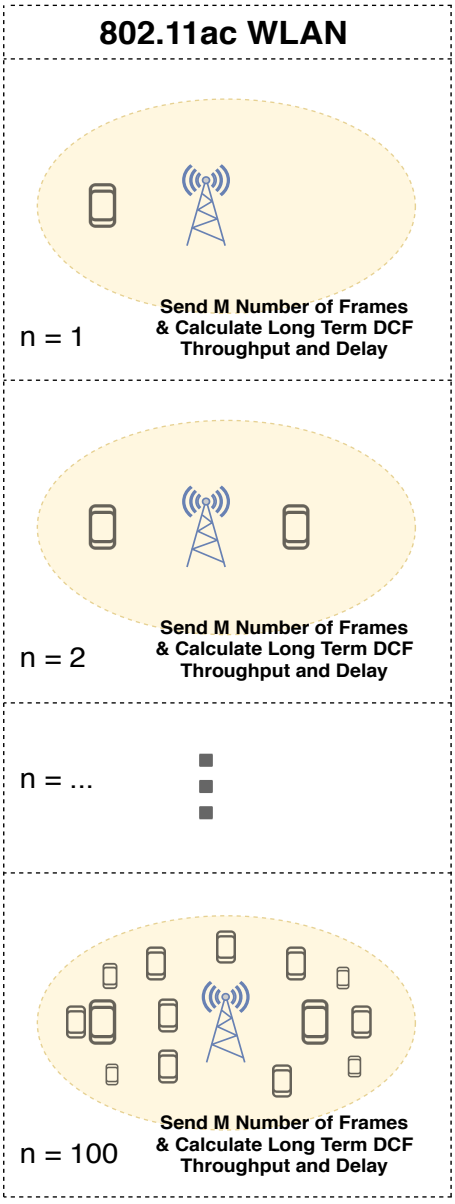

Figure 2.29: Applied Channel Load used for 802.11ac Validation

2.4.2.3.2 802.11ah Validation Test The test scenario used for the 11ah validation is shown in Figure 2.30 below. Similar to the 11ac test above the applied load on the channel is increased from n = 1 competing nodes to n = 100 and at each step a group total M number of frames are transmitted and the long-term channel and tagged STA throughput, delay, and conditional collision likelihood are calculated and appended to output vectors. In both tests the output is an N-length vector giving the simulated DCF statistics as a function of the applied channel load.

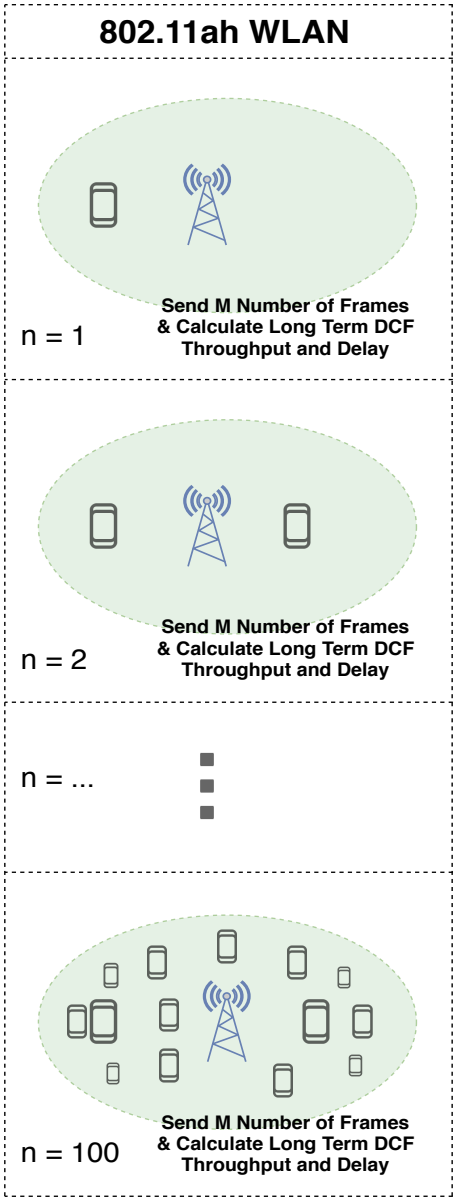

Figure 2.30: Applied Channel Load used for 802.11ah Validation

2.4.2.4 Verification against Bianchi and Chatzimisios

Next we verify the output of the implemented DCF simulator against the widely accepted analytical modelling by the authors Bianchi and Chatzimisios.

2.4.2.4.1 802.11ac Subjective Verification The validation results for the 11ac DCF simulation are presented in the graphs below and on the opposite page. The simulated DCF channel throughput and conditional collision probability are plotted against the same output from the Bianchi model in Figure 2.31 and Figure 2.33 respectively, and the simulated channel access delay is plotted against the Chatzimisios model in Figure 2.32.

Note from Figure 2.33 opposite that the analytical collision probability from the Bianchi model is slightly lower compared to its simulated output value. This has the knock-on effect of slightly decreasing the delay and increasing the channel throughput in relation to the output of the model, as also seen in both Figure 2.32 and Figure 2.31 respectively. Furthermore the resonance at n = 2 is not quite as pronounced in the simulations compared to Bianchi, however from the visual analysis off all three outputs, the implemented simulator was found to provide a reasonable representation of the DCF throughput performance.

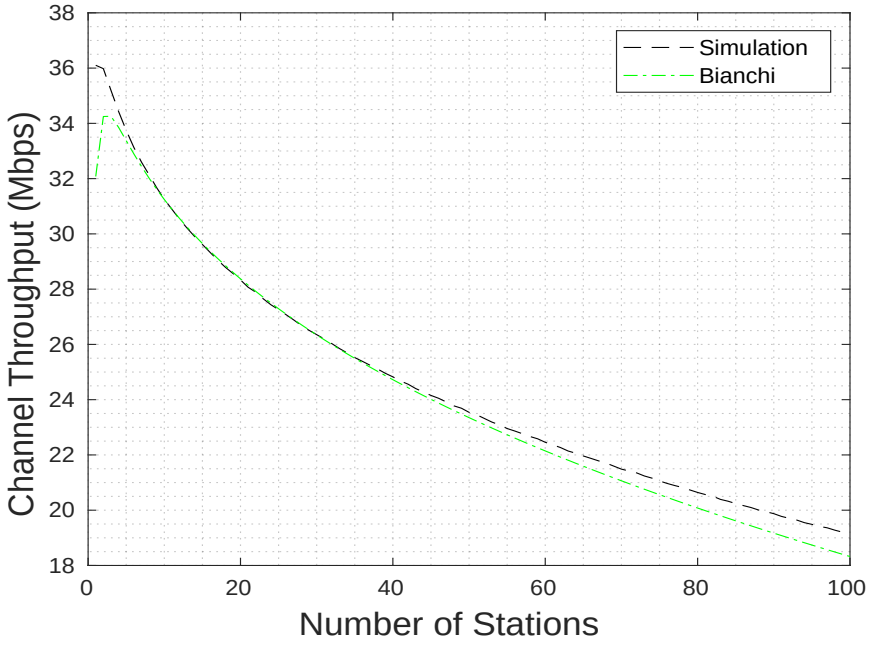

Figure 2.31: 802.11ac Saturation Throughput Simulation Validation Results

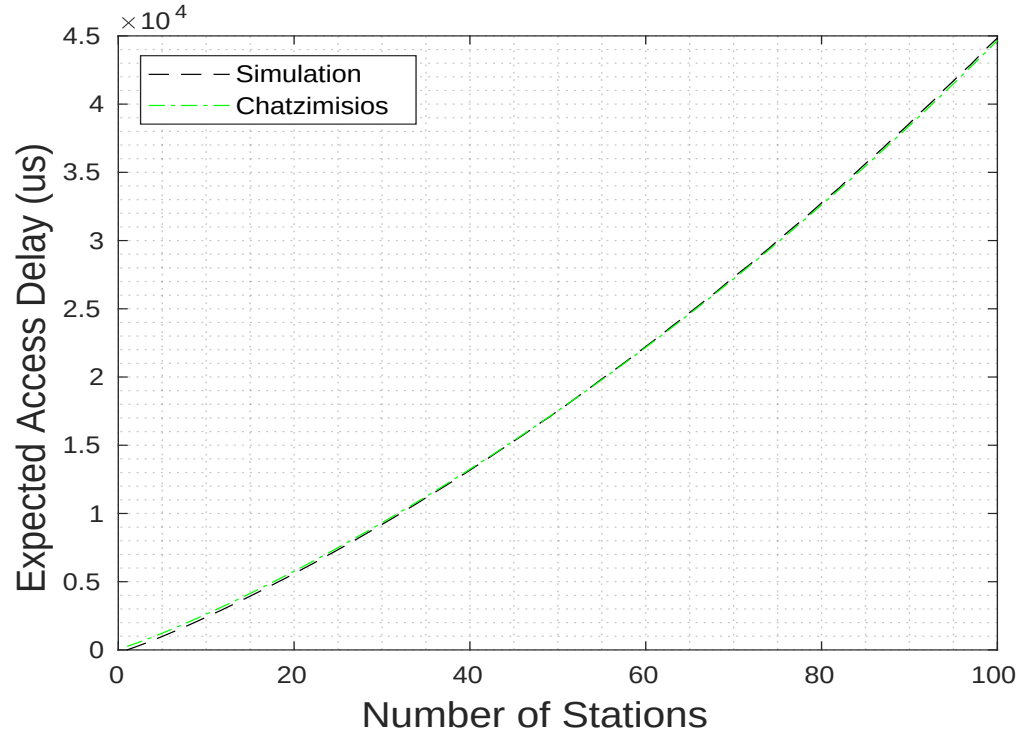

Figure 2.32: 802.11ac Expected Access Delay Simulation Validation Results

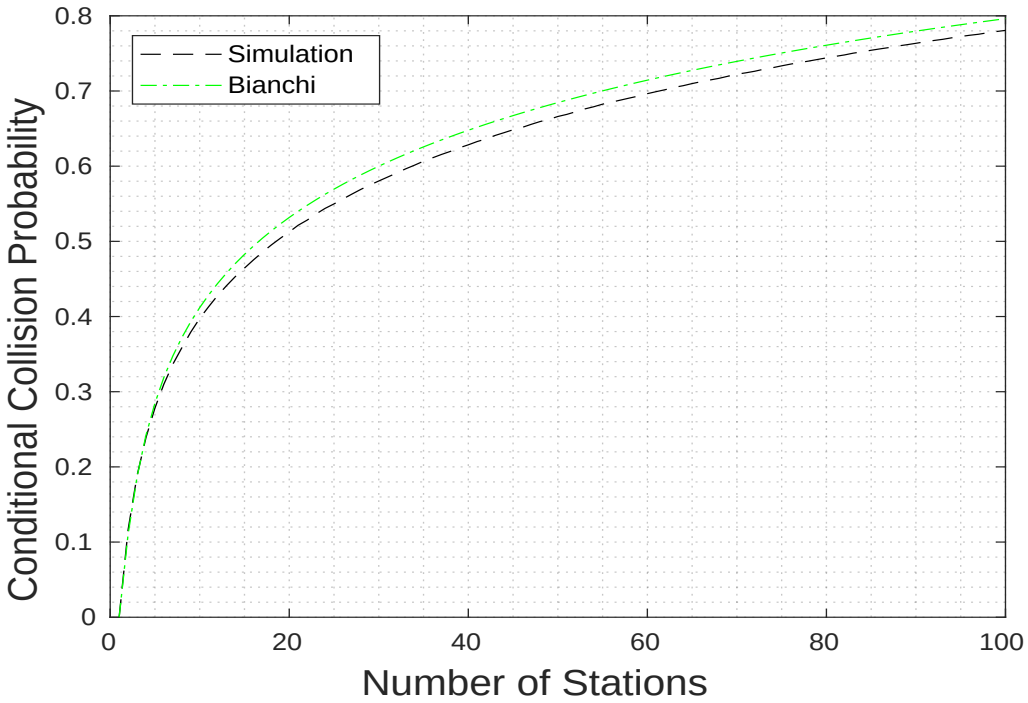

Figure 2.33: 802.11ac Conditional Collision Probability Simulation Validation Results

2.4.2.4.2 802.11ac Objective Verification Next we present the results from the objective verification testing for the implemented 11ac DCF simulator using statistical confidence intervals and the Bianchi and Chatzimisios reference models.

Compared to the evaluated values from the expressions provided by Bianchi and Chatzimisios, the implemented DCF simulator framework was found to model the saturation throughput, expected access delay, and conditional collision probability of the 802.11ac DCF channel with an accuracy of within ±1.15%. This value was calculated by taking an average of the percentage difference between the simulation output and the Bianchi-based reference models over each of the $n = 100$ data points in each test.

A stronger verification is obtained via the use of statistical confidence intervals, following the technique used in multiple works by [145] and [19].

The mean values for the simulation outputs are shown in Table 2.9, alongside their 99.9% confidence intervals, the evaluated values from the analytical modelling, as well as the difference between the simulator outputs and the analysis.

The main finding was that for the three key DCF metrics, the output of the analytical modelling was well within the desired 99.9% confidence intervals calculated from the DCF simulation output, therefore verifying the implemented simulator with a high degree of certainty.

	Throughput	**Delay**	**Coll. Prob.**
Simulation Mean Values	24.4768	18896	0.6217
99.9% Confidence Interval	±0.330492	±720.0875	±0.012814
Model Computed Value	24.196	19510	0.6321
Sim-Analysis Difference	-0.2808	+614	+0.0104

Table 2.9: 802.11ac Validation Results and Confidence Intervals

2.4.2.4.3 802.11ah Subjective Verification The validation results for the 802.11ah DCF simulation are shown in the figures below and on the opposite page.

The DCF channel throughput and conditional collision probability from the simulator are plotted alongside the output of the Bianchi modelling in Figure 2.34 and Figure 2.36 respectively, and the expected channel access delay is compared against the Chatzimisios model in Figure 2.35.

Similar to the 11ac verification, in Figure 2.36 the collision probability output from the simulation was slightly lower compared to the evaluated modelling value which slightly reduced the delay and increased the achieved throughput in relation to the model output as seen in Figure 2.35 and Figure 2.34 respectively.

Again, the resonance at n = 2 was not as pronounced compared to Bianchi, but as seen from the plotted data the implemented simulator provides a reasonable representation of the 802.11ah DCF throughput performance as given by the Bianchi and Chatzimisios modelling.

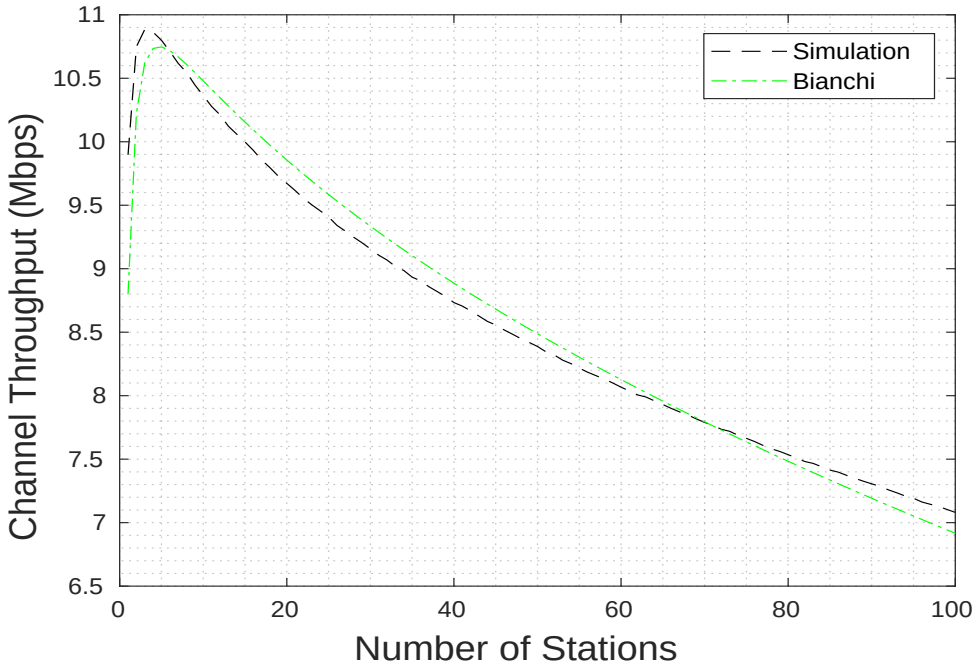

Figure 2.34: 802.11ah Saturation Throughput Simulation Validation Results

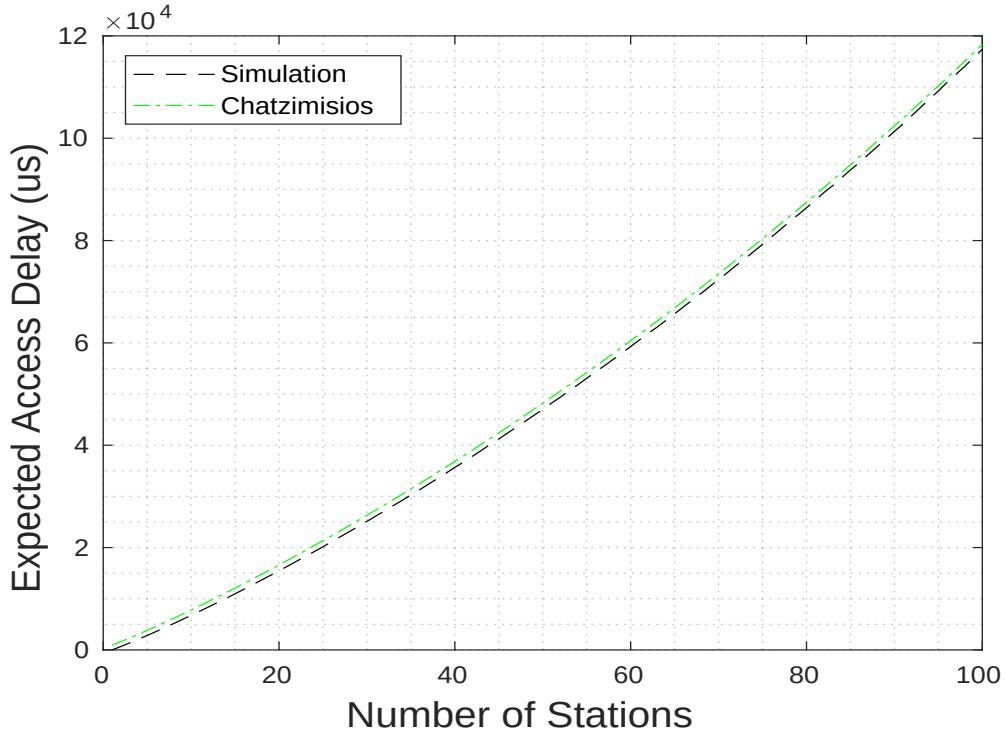

Figure 2.35: 802.11ah Expected Access Delay Simulation Validation Results

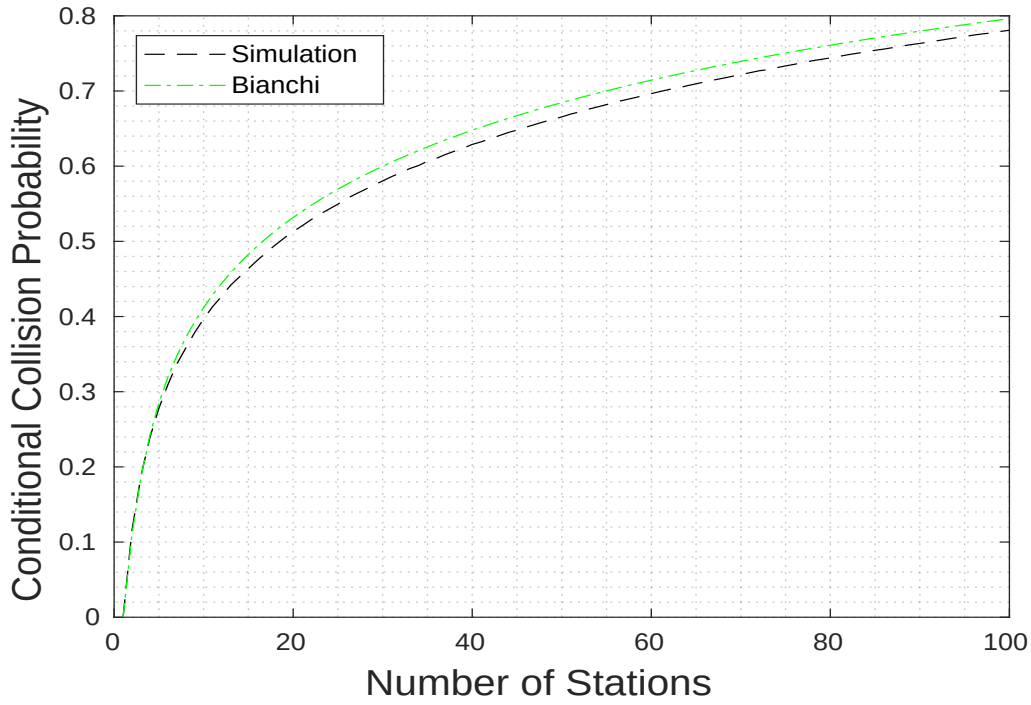

Figure 2.36: 802.11ah Conditional Collision Probability Validation Results

2.4.2.4.4 802.11ah Objective Verification Next we present the objective validation test results for the implemented 11ah simulator using multiple confidence intervals and the Bianchi and Chatzimisios models.

Compared to the values from the modelling provided by Bianchi and Chatzimisios, the implemented DCF simulator was seen to model the saturation throughput, expected access delay, and conditional collision probability of the 802.11ac DCF channel with an average accuracy of within ±0.599%, calculated by taking an average of the percentage difference between the simulation output and the analytical modelling over each of the $n = 100$ data points.

Again, a stronger verification is provided by the use of confidence intervals. The mean values for the three main simulation outputs for the DCF channel are given in Table 2.10 alongside the 99.9% confidence intervals, the outputs of the Bianchi and Chatzimisios modelling, and the difference between the modelling and simulation outputs.

The main finding was again that for the three main DCF metrics, the outputs of the analytical modelling were well within the desired 99.9% confidence intervals, therefore verifying the correctness of the implemented simulator with a high degree of certainty.

	Throughput	**Delay**	**Coll. Prob.**
Simulation Mean Values	8.5549	49942	0.6215
99.99% Confidence Interval	±0.235682	±1072.077	±0.008819
Model Computed Value	8.754	52397	0.6288
Sim-Analysis Difference	+0.1991	+2544	+0.0073

Table 2.10: 802.11ah Validation Results and Confidence Intervals

2.4.2.5 Demonstrating Long Term Fairness

In order to provide the single-link basis for the 802.11 interface bonding framework in Chapter 3, in this section we extend and further verify the presented DCF channel simulator to provide the saturation throughput, expected access delay, and conditional collision probability for a *single* competing wireless station. We then verify the correct operation of the extended single-node simulator using the long-term fairness of the 802.11 MAC and a simple modification to the Bianchi channel throughput model. The long term fairness of the DCF is defined by Duda [56] as the principle that the probability of successful channel access converges to 1/N as the total number of frames M transmitted tends to infinity. The modelling extensions used in the following validation are therefore achieved trivially by dividing the Bianchi channel throughput by n, which the total number of competing stations present in the network.

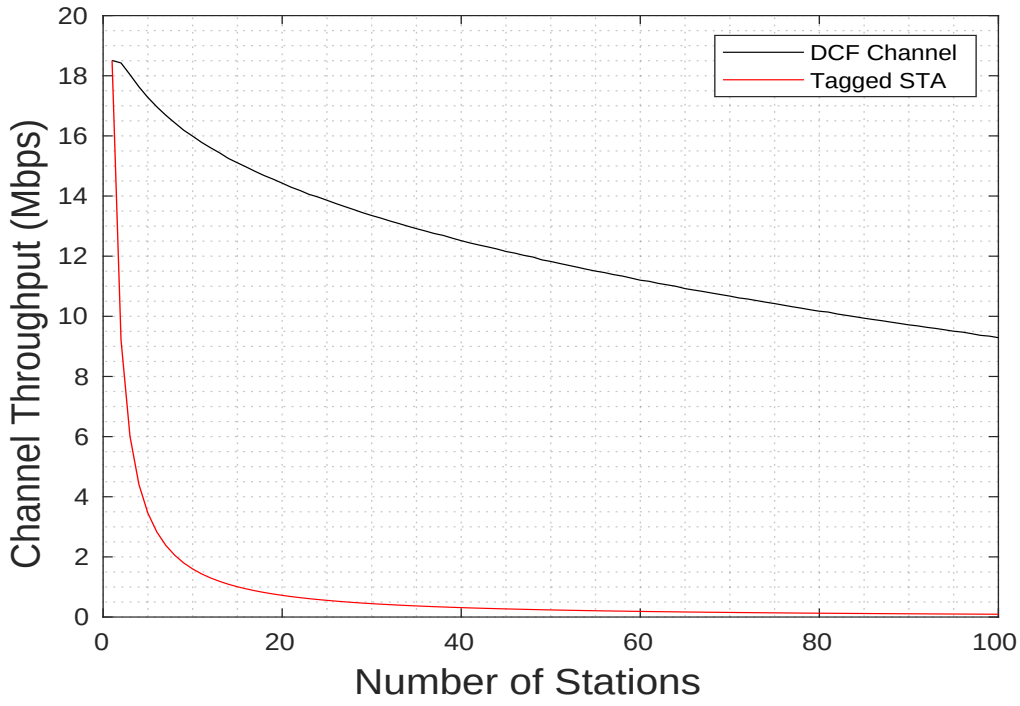

Figure 2.37: 802.11ac Simulated Tagged Node Saturation Throughput

2.4.2.5.1 Comparison of DCF Channel and STA Performance First we compare the simulated saturation throughput, conditional collision probability, and expected delay of the individual tagged STA against the same statistics for the overall shared DCF channel. The achieved throughput for the tagged DCF node is plotted in Figure 2.37 below as the solid red alongside the DCF channel throughput as the solid black line. As seen the output for the tagged-node reduces exponentially while the channel throughput displays a more gradual non-linear decrease.

In the channel saturation throughput we see an overall reduction as the applied load is increased due to the additional latency of the CSMA/CA back-off algorithm employed. The channel throughput decreases from a maximum of 18.51 Mbps at n = 1 to a minimum of 9.296 Mbps at the heaviest test load of n = 100. By contrast the tagged node output decreased exponentially from 18.51 Mbps at n = 1 to a minimum of 0.0926 Mbps. At n = 1 the tagged node had sole occupancy of the channel and therefore its achieved throughput was roughly equal to the overall channel throughput at 18.51 Mbps. Then as the load was incremented to n = 2 the available channel resources became divided equally such that both nodes received 1/N of the total 18.44 Mbps capacity. As more nodes were added to the network the tagged node received a diminishing share of the total channel resources until at the heaviest considered load of n = 100 the node throughput was reduced to one hundredth of the total channel capacity.

Likewise the expected delay for the tagged STA is shown in Figure 2.38 as the dashed red line alongside the overall DCF channel delay as the dotted and dashed black line.

As expected due to the inherent fairness of the DCF, as the applied channel load is raised the tagged STA delay converges to the average channel access delay for a large enough number of transmitted frames. During the test the overall mean DCF channel access delay was 36757 μs while the mean delay of the individual STA was 37638 μs. Both outputs increased non-linearly from a minimum of approximately 0.0003 μs at n = 1 to a maximum of 85740 μs for the channel delay and 87620 μs for the tagged STA delay at the heaviest considered load of n = 100.

Finally the conditional collision probability for the tagged node is plotted in Figure 2.38 as the solid red line with the collision probability for the overall DCF channel as the solid black line.

Again as expected due to the long term fairness of the 802.11 DCF the collision probability of the tagged STA node converges to that of the overall channel for a high enough number of transmitted frames. Both outputs decreased non-linearly from a minimum of 0 at n = 1 to a maximum of 0.781 for the channel and a maximum of 0.7943 for the tagged STA at the heaviest considered load of n = 100. The overall mean collision probability for all applied loads considered was 0.6194 for the DCF channel and 0.6261 for the tagged STA node.

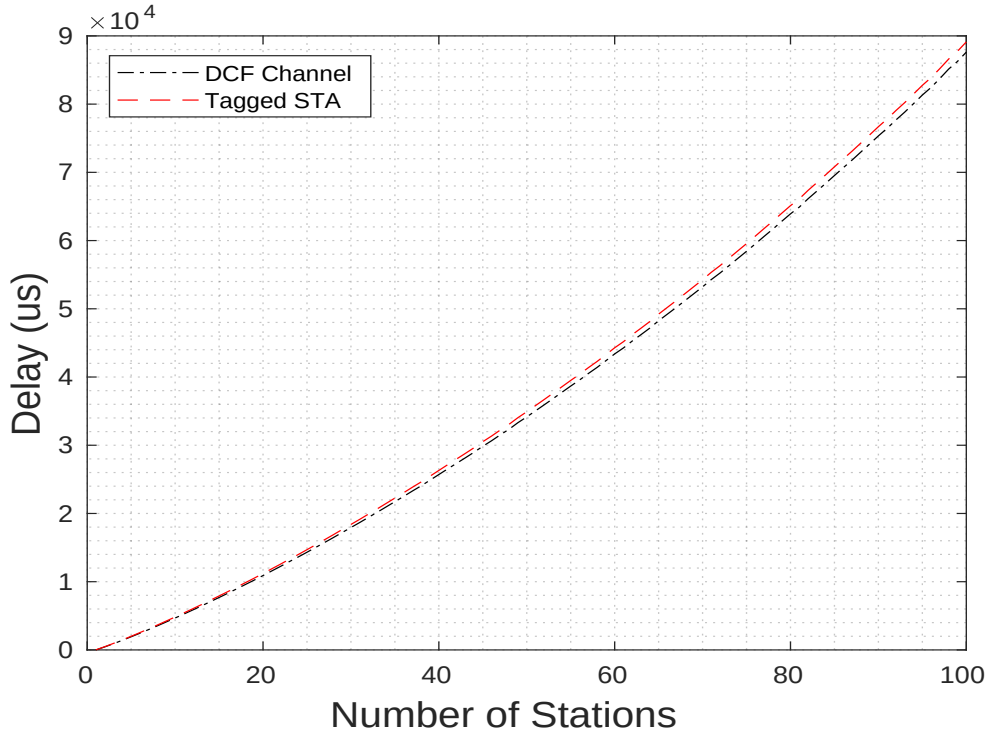

Figure 2.38: 802.11ac Simulated Tagged Node Expected Access Delay

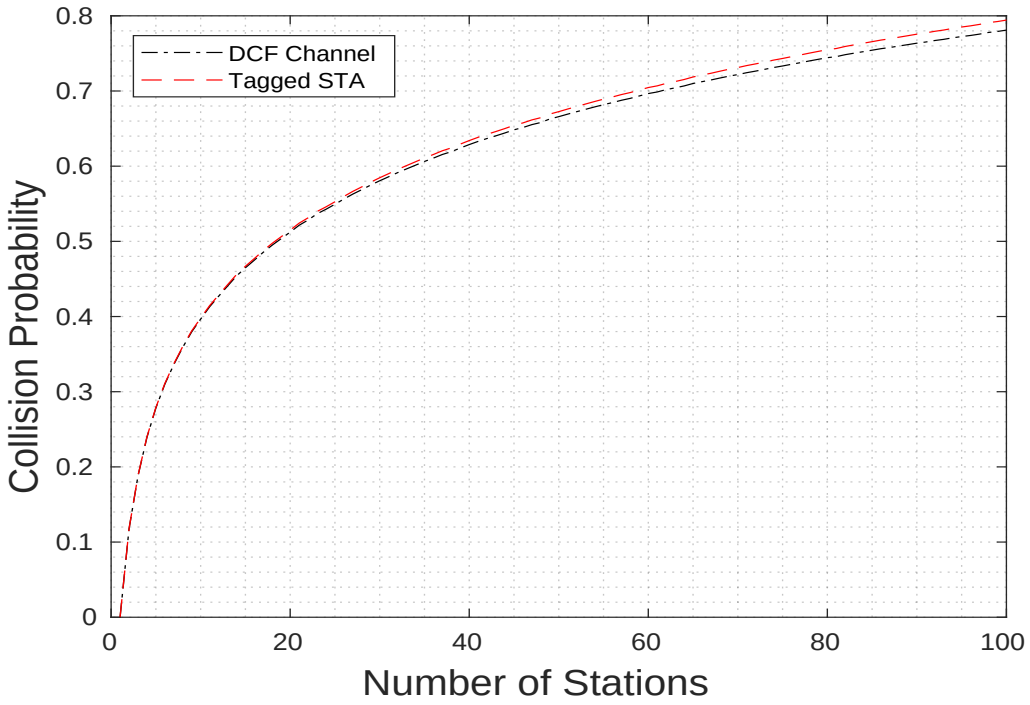

Figure 2.39: 802.11ac Simulated Tagged Node Conditional Collision Probability

2.4.2.5.2 Verifying the Tagged STA Simulator using Bianchi Using the previous observations, it is straightforward to adapt the Bianchi model for the DCF channel saturation throughput to provide an expression for the throughput an individual 802.11 STA that can be used to validate the implemented single node simulation. Under the assumption of saturated traffic conditions the throughput of a single STA is given as the Bianchi channel throughput Sn divided by the number of competing stations n present in the network:

$$S_{STA} = S_n/n \tag{2.25}$$

We saw earlier that the collision probability and access delay of the single tagged node converge to that of the overall DCF channel for sufficiently large values of transmitted frames and therefore no modification is required for these expressions.

The single node throughput expression is validated against the STA throughput from the implemented MatLab simulation in Figure 2.40. As seen below the presented formula provides a good representation of the single STA throughput as the applied channel load is increased with an average accuracy of ±2.017%. The overall mean simulated STA throughput was 0.8161 Mbps while the mean throughput from the modelling expression above was 0.7998.

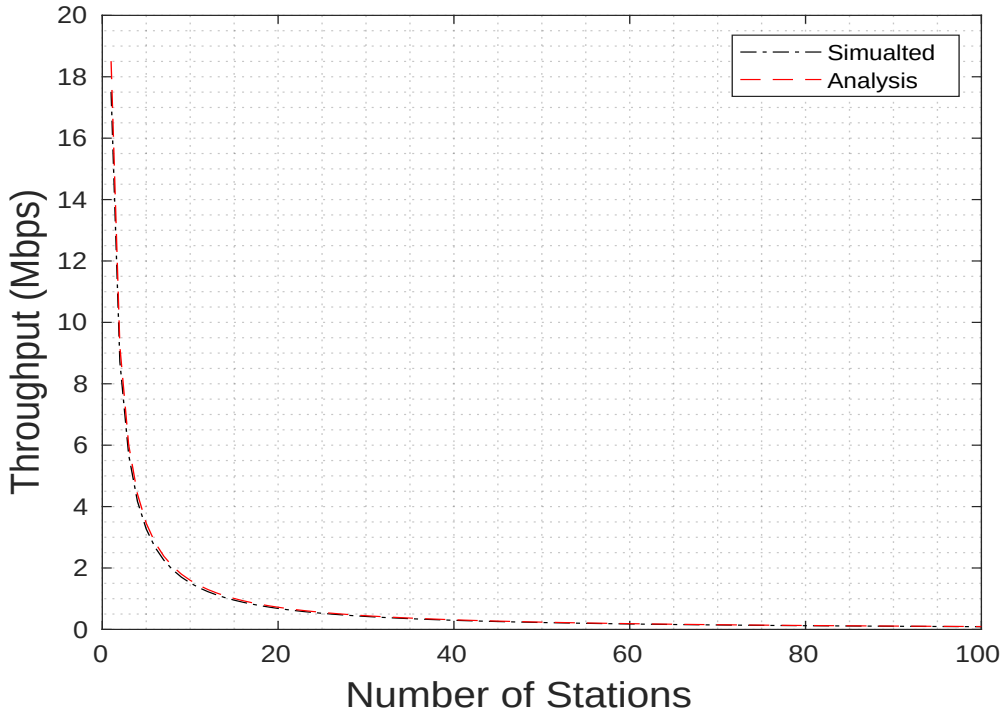

Figure 2.40: Validation Results for 802.11 STA Throughput Extension to Bianchi

2.5 Chapter Conclusion

This initial chapter laid the foundation for the following main thesis research by analysing, evaluating, and simulating the contention-based 802.11 MAC layer channel access mechanism.

The main contribution of the chapter work was the implementation and validation of a novel MatLab-based 802.11 DCF evaluation framework. We saw in Section 2.4.2 that the framework was able to model the overall DCF channel performance for both 802.11 types considered with a high level of accuracy. The DCF channel saturation throughput, expected access delay, and conditional collision probability from the Bianchi modelling were found to fall within the 99.9% confidence interval of their respective simulation outputs, and on average the simulated DCF channel throughput was within ±1% of the numerical output of the Bianchi and Chatzimisios modelling.

The validation testing performed plays an important role in the context of the main thesis research, as the developed DCF channel and STA simulations are subsequently extended and used to evaluate the performance of statically configured homogeneous 11ac-11ac and heterogeneous 11ac-11ah interface bonds in Chapter 3, and is further extended in Chapter 4 to evaluate the performance of state of the art dynamic interface selection algorithms.

The chapter also described the differences between the traditional 2.4 GHz 802.11ac and the recently published sub-1 GHz 802.11ah standard designed specifically for long-range and low-power operation in wireless sensor networks and the Internet of Things. In particular from the DCF performance comparison in Section 2.3.3, the difference in saturated channel throughput and expected access delay between the two protocol versions was 23.74% and 67.14% respectably, despite both using the same raw PHY data rate of 65 Mbps. Although the theoretical PHY layer data rates were identical, the differences in the DCF parameters at the MAC layer led to a significantly lower throughput performance for the HaLow version of the Wi-Fi standard. The results of the performance comparison provided useful insight into the relevant performance between the 11ac and 11ah protocol versions, of which a clear understanding is essential in order to correctly interpret the heterogeneous 11ac-11ah interface bond performance as studied in Chapters 3 and 4, to which we next turn our attention.

CHAPTER 3

EVALUATING STATIC 802.11 INTERFACE BONDING PERFORMANCE

To provide a benchmark against which to evaluate the dynamic interface bonding methods researched later in the thesis, in this chapter we analyse the performance of a statically configured dual-802.11 interface bond at the MAC layer under varying conditions of applied link load and the assumption of saturated arrival traffic. The evaluation is performed using a custom bonding layer implemented in MatLab and based on the single-link DCF simulations from the previous chapter. During the experiments, the aggregate throughput, delay, and link bandwidth utilisation of the homogeneous 11ac-11ac and heterogeneous 11ac-11ah interface bonds considered are analysed and compared at similar levels of applied link loads.

3.1 Chapter Research Overview

Personal mobile devices such as smart phones and tablets have long been equipped with multiple wireless network interfaces for WWAN, WLAN, and WPAN connectivity, typically consisting of LTE, Wi-Fi, and Bluetooth/NFC technologies respectively.

For as long as this has been the case there has also been an interest in exploiting the fact using various link aggregation or interface bonding techniques, in order to improve the throughput, delay, and reliability of both end-to-end and point-to-point communication links, as well as increasing overall network throughput using various traffic load balancing approaches. Multiple technologies exist that provide this capability in some form at all layers of the protocol stack. For example Linux bonding, 802.11 channel bonding, LTE carrier aggregation and LTE-WLAN Aggregation (LWA) at the Link Layer, the Multi-Path Transmission Control Protocol (MPTCP) at the Transport Layer, and multiple experimental approaches at the Application Layer.

Of particular interest for the current research work is the software-based approach to bonding

as used in the open source Linux kernel. Several scheduling options such as round robin and broadcast mode offer an entirely link technology agnostic method of interface bonding without requiring direct device driver intervention, similar to the packet based bonding approach used by MPTCP only at the Link Layer over multiple *point-to-point* links.

It is well-known that using Linux bonding (or any other packet-based Layer 2 approach) over wireless interfaces with a static round robin configuration is problematic for upper layer protocols such as TCP, due to short-term variations in the individual link access delays. As shown in multiple experimental works, these effects are so severe that the aggregate bond throughput can actually be lower than the throughput of the fastest available single link, therefore rendering the application of the technique counter-productive to the goal of increasing the overall throughput.

While the effects on adjacent layers are well-documented, the theoretical long-term performance of 802.11 interface bonding at the MAC layer is largely overlooked by researchers. In the results of the bonding simulations presented later in the chapter, we show that the counter productive scenario described above occurs commonly for an 802.11 bond at clearly defined combinations of applied links loads, even with a lossless physical channel and without considering the adverse effects on upper protocol layers.

This degradation is caused by the slow-down effect that is inherent to the process of interface bonding itself, i.e. that all data frames or packets must be correctly received over all bond links before the structured data can be passed to the adjacent protocol layer. For bonds with heterogeneous link performance, this creates a lag as the faster slave link is forced to wait for the slower bond link to finish transmitting, thereby reducing the achieved bond throughput.

The diversification of the 802.11 standard into non-traditional Wi-Fi applications such as vehicular networks and the Internet of Things (IoT) provides motivation to evaluate the interface bonding performance using heterogeneous 802.11 devices. As shown by the presented simulation results, there is a significant degradation in the heterogeneous case compared to the homogeneous 11ac-11ac bond, due the difference in DCF parameters, such as the slot duration and interface spacing, between the 11ac and 11ah standards, notwithstanding the fact that both links were configured with an identical raw PHY data rate and had the same level of applied link load.

3.1.1 Chapter Research Objectives

Towards the overarching thesis objective, the primary research goal of the chapter work is to evaluate and analyse the achieved throughput performance of the traditional static interface bond configuration under varying conditions of applied link loads, in order to provide a performance benchmark against which to compare the state of the art dynamic interface selection algorithms in Chapter 4. The secondary research aims of the chapter are given as follows:

1. **Analysing MAC Layer Impact on the Throughput Performance of Point-to-Point 802.11 Interface Bonding:** The severe degradation effects on Transport Layer protocols such as TCP and UDP caused by short-term variations in the stochastic DCF channel access delay, are well studied in the research. However, with respect to the long-term impact of the DCF on the aggregate 802.11 interface bond throughput under heavily unbalanced conditions, there is a significant lack of relevant and informative research. Therefore one of the secondary aims of the chapter is to closely evaluate the impact of the DCF MAC layer on the overall 802.11 interface bonding performance using the purpose-built MatLab evaluation framework.

2. **Performance Comparison of Homogeneous 11ac-11ac and Heterogeneous 11ac-11ah Bonds at the MAC Layer:** To help exploit the availability of multiple options for WLAN connectivity in current consumer wireless devices, another secondary aim of the chapter research is to analyse and compare the achieved throughput performance between a homogeneous 11ac-11ac interface bond and heterogeneous 11ac-11ah bond under identical conditions of applied link load. In particular, to evaluate the impact of the difference in DCF MAC layer parameters, such as the inter-frame spacing and slow duration, for the same level of applied load and identical raw PHY data rates.

3.1.2 Chapter Research Contributions

The key research contributions of the work presented in Chapter 3 are given as follows:

1. **Implementation of a novel MatLab-based evaluation framework for heterogeneous 802.11 interface bonding at the MAC layer:** In order to enable to the main research work of the chapter, the validated single-link DCF simulator presented in Chapter 2 is further extended to model the performance of static round robin 802.11 interface bonding at the MAC layer between point-to-point links. The implemented extensions also enables the evaluation of different heterogeneous 802.11 link technologies and raw PHY data rates.

2. **Rigorous analysis of the static 802.11 bond performance and inherent slow-down effect due to the in-order delivery requirement:** To provide a benchmark against which to test the dynamic link selection algorithms in the following chapters, a rigorous evaluation of the point-to-point 802.11 interface bond performance is presented. Using the implemented framework a number of balanced and unbalanced applied bond load tests are performed in which the mean bond throughput, delay, and link bandwidth, are measured, analysed, and discussed.

3. **Comparison of the homogeneous 11ac-11ac and heterogeneous 11ac-11ah bond performance under multiple applied load scenarios:** To understand the difference in behaviour between homogeneous and heterogeneous 802.11 interface bond types, we present a detailed throughput performance comparison between a dual-11ac and a 11ac-11ah interface bond, under a similar set of applied load scenarios and identical 65Mbps-65Mbps raw PHY data rates.

3.1.3 Summary of Chapter Findings

The key findings of the chapter research are as follows:

- **Quantifying the inherent bond throughput slow-down effect:** The main finding from the evaluation of the static 802.11 round robin bond configuration was that, when calculated over all eight tests for both the homogeneous 11ac-11ac and heterogeneous 11ac-11ah interface bonds, the overall mean link bandwidth wastage due to the inherent bond throughput slow-down effect was equal to approximately 18.31% of the total available capacity.

- **Problematic applied link loads for in-order delivery:** Notably, in 7 out of the 8 individual tests performed, at certain levels of applied load the impact of the inherent slow-down effect was so severe that it reduced the aggregate throughput of the static round robin bond to below that of the single fastest available individual link, therefore rendering the use of the interface bonding technique counter productive to the aim of increasing the overall link speed. The surprising aspect was that the above condition was not an uncommon occurrence for an 802.11 interface bond with heterogeneous applied loads, occurring for each individual applied load test with the exception of the balanced benchmark case for the homogeneous 11ac-11ac interface bond in Test 1A.

- **Comparison of the 11ac-11ac and 11ac-11ah interface bonds:** The performance of the heterogeneous 11ac-11ah interface bond was found to be significantly lower when compared to that of the homogeneous 11ac-11ac bond, despite the identical balanced applied load and 65Mbps raw PHY date used by both 802.11 slave links. Due to the difference in DCF parameters such as slot duration and DIFS between the two standards, the mean 11ac-11ac bond throughput achieved by the use of static round robin was 6.085 Mbps compared to just 4.1 Mbps for the heterogeneous 11ac-11ah bond. Furthermore, the mean link bandwidth was also substantially lower at 78.75% compared to 84.62% for the homogeneous bond.

3.1.4 Chapter Organisation

The remainder of the chapter is structured as follows. All required background information is presented in Section 3.2, which gives an overview of interface bonding concepts, techniques, and related research including both wired and wireless applications. The implemented 802.11 bonding MatLab framework is described in Section 3.3. The input parameters and applied load tests used in the static round robin 802.11 bonding evaluation are given in Section 3.4. The quantitative results and qualitative discussion are given in Section 3.5, and a detailed performance evaluation and comparison between the 11ac-11ac and heterogeneous 11ac-11ah bonds is provided in Section 3.6.

3.2 Chapter Background Information

This section presents an overview of interface bonding concepts, algorithms, research, technical challenges, and open research questions at all layers of the protocol stack, including both wired and wireless applications of the technique.

3.2.1 Definition of Interface Bonding

Interface bonding, also known as link aggregation, multi-path streaming, trunking, bundling, teaming, or striping, is the technique of combining multiple network interfaces to present a single unified resource to the upper protocol layers. The technique can be used over both wired and wireless device types to increase dates rates and provide frame-level or link-level redundancy.

The act of bonding can be performed at all layers of the stack but is most commonly used at the Link and Transport layers. In most implementations such as the Multi-Path Transmission Control Protocol (MPTCP) and Linux, the interface bonding is performed agnostically within the operating system kernel networking stack, on a packet-by-packet basis using only Layer 2/3/4 header information.

A important distinction is to be made between wireless interface bonding, which occurs over physical Network Interface Cards (NICs)[1] at the Link Layer and above, and channel bonding or carrier aggregation which occurs over different radio frequency channels at the Physical Layer as first introduced in 802.11n published in 2009.

The research work presented in this thesis is primarily concerned with 802.11 interface bonding at the Link Layer using directly connected point-to-point wireless links. Compared to the device/driver integrated channel bonding and carrier aggregation solutions in 802.11n and LTE-U, the type of kernel-software-based interface bonding used in Linux and MPTCP is cheaper, more flexible, and far simpler in implementation. However, as shown in [171] the use of the traditional static round robin approach with heterogeneous link throughput performance leads to a significant throughput reduction and high number of out-of-order TCP segments.

Note that in the research literature, the term link aggregation is sometimes also used to refer to load balancing, which is the technique of dividing the overall network traffic stream across multiple available links to balance the load across redundant services or network devices on a connection-by-connection basis, usually by hashing the source-destination IP and MAC addresses.

[1] The NICs themselves may be on-board or external using PCI or USB input/output

3.2.2 Benefits of Interface Bonding

The use of interface bonding has the following benefits that apply equally to both wired and wireless applications of the technique:

- **Faster Data Rates:** Multiple individual interfaces may be grouped together to form a single, higher capacity link, which is often much cheaper than a single high speed link. In overlapping wireless networks this allows the user to take advantage of multiple available access points when using data hungry applications. Typically round robin scheduling is used, as per MPTCP [63], Open vSwitch (OVS) [74, 106], and Linux bonding [46]. Note that this particular application is the sole focus of the work presented in the thesis.

- **Improved Reliability:** By transmitting all outgoing frames over all available links of the bond, the destination node has two streams of data from which to select frames for decoding. This provides a benefit in terms of the reliability of the link, which is particularly useful in unpredictable wireless networks, but can also provide a benefit in terms of the overall link delay by selecting the frame with the shortest delay. A example of this type of bonding is broadcast mode in Linux Bonding and Red Hat Teaming [41]. However, this type of bonding is not considered in the research work presented in the thesis.

- **Automatic Fail-over:** In this configuration interface bonding is used to provide automatic fail-over in the event of single link failure. One slave is selected as the active, primary slave, and another is kept as a hot-standby. Most bonding solutions offer this mode of operation. In Linux bonding and Red Hat Teaming this configuration is known as active-standby mode. Please note that although this is perhaps the most common use of interface bonding in most practical deployments, this case is not considered in the thesis research work.

3.2.3 Functional Components of Bonding Systems

In the survey in [79], the authors Habak et al. define the four main functional components of an interface bonding solution as:

1. **Link Monitoring Unit:** The link monitoring unit is responsible for monitoring the state of individual network links (i.e. whether the link is operationally up or down) as well as the instantaneous link performance. Monitoring approaches may be active, for example by direct probing, in which case a specific device driver interface is required for example, the ethtool application as used in the Linux kernel bonding module, or passive, for example using various statistics collected in-line from the actual data stream, for example, as used by the authors So et al. in [180].

2. **Re-sequencer:** The re-sequencer is responsible for the re-ordering of data frames at the receiver, which is necessary before passing the structured data to the upper layer, thereby presenting individual bonded links as a single, unified resource. In current implementations such as MPTCP and Linux bonding re-sequencing is performed using a second-tier frame sequencing scheme which makes extensive use of receiver side packet buffers.

3. **Scheduler:** The main responsibility of the scheduling component in an interface bonding solution is the allocation of individual data frames from a single application traffic stream amongst the available network devices. This is traditionally done in a static and fixed fashion using well-known algorithms such as round robin, weighted round robin, broadcast, or minimum round trip time. The scheduler is therefore one of the key aspects of an interface bonding system contributing to the overall performance.

4. **Interface Selector:** The main role of the interface selection unit is the creation and configuration of interface bonds, as well as the selection of an appropriate set of links to be used as enslaved devices. In most current interface bonding implementations such as Linux bonding and MPTCP, the number of interfaces used is static and bonds are manually configured by a skilled and experienced network engineer, with or without the help of the Link Aggregation Control Protocol (LACP). Dynamic interface selection, as used in many state of the art interface bonding researches such as [130], is the main focus of the research work presented in this thesis.

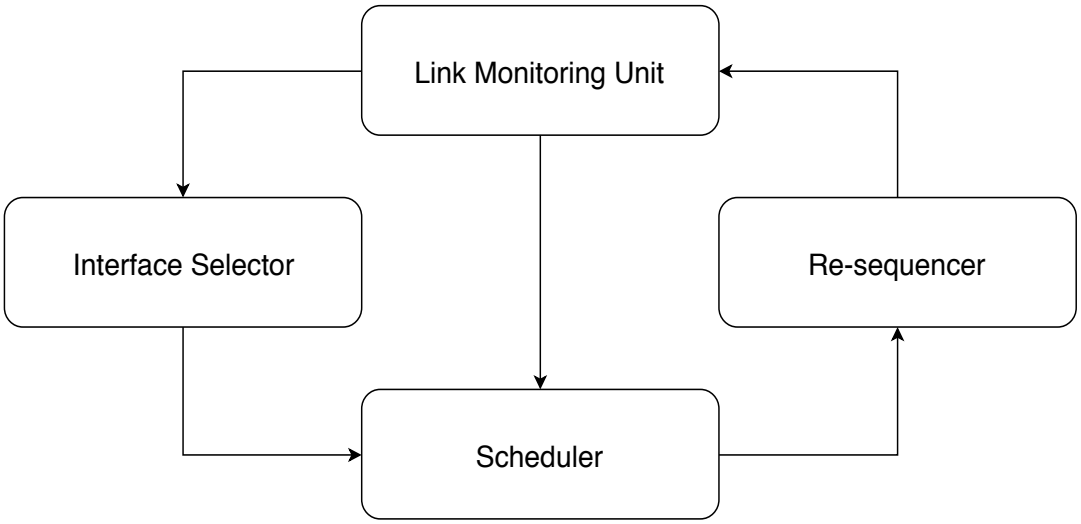

Figure 3.1: Components of an Interface Bonding System defined by Habak et al.

3.2.4 Categorisation of Interface Bonding Systems

This section describes the different categorisations of current link aggregation technology, standards, and research. These categories include the commonly used layered categorisation based on the TCP/IP model, an alternative inter-layered categorisation, and a comparison of end-to-end vs point-to-point bonding.

3.2.4.1 Layered Categorisation

A form of link aggregation may be performed at all layers of the TCP/IP protocol stack: (1) at the Physical Layer as per 802.11n channel bonding and LTE-U, (2) at the Link Layer as per LACP and Linux bonding, (3) at the Network Layer for example Multi-path routing and some of the pre-MPTCP research surveyed by Habak et al. in [79], (4) at the Transport Layer, i.e. MPTCP, and (5) at the Application Layer, for example the Android-based smart phone implementation in [193].

The main objective is the same at all layers: to exploit the availability of redundant communication links. If bonding to increase link speed the main task is also the same at all layers: to transmit a single data stream over multiple communication links and reassemble the data at the receiver, be these (1) bits at the Physical Layer, (2) frames at the Link Layer, (3) packets at the Network Layer, (4) TCP segments at the Transport Layer, or (5) data chunks of any size and type at the Application Layer. Heterogeneous link access delay and throughout pose a challenge at all protocol layers and when these change dynamically, it effects the overall aggregate bond throughput at each layer in a similar way.

3.2.4.2 Alternative Layered Categorisation

It is often useful to think of bonding as occurring *between* the different protocol layers. For example round-robin mode in Linux may be considered as a Layer 2.5 approach as it does not need a specialist interface between the bond software and device driver. This technology agnostic approach at Layer 2.5 is the main focus of the bonding simulations and research presented later in the thesis. This categorisation is useful when evaluating proposals prior to the publication of MPTCP in 2013. Several earlier Transport Layer works such as those described in [79] are better thought of as Layer 3.5 approaches due to their use of a single TCP stack. However this approach was found to limit the aggregated bond throughput and subsequent researches demonstated the use of independent congestion control. This was the method eventually adopted by MPTCP and is arguably described as a Layer 4.5 approach.

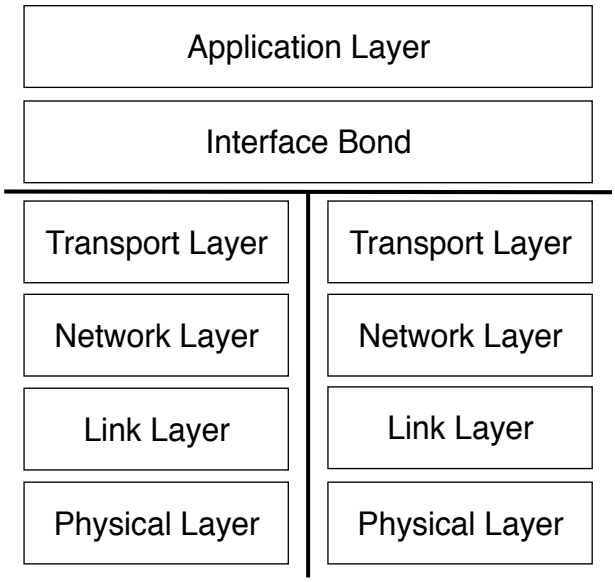

Figure 3.2: Bonding at Layer 4.5 of the Protocol Stack (e.g. MPTCP)

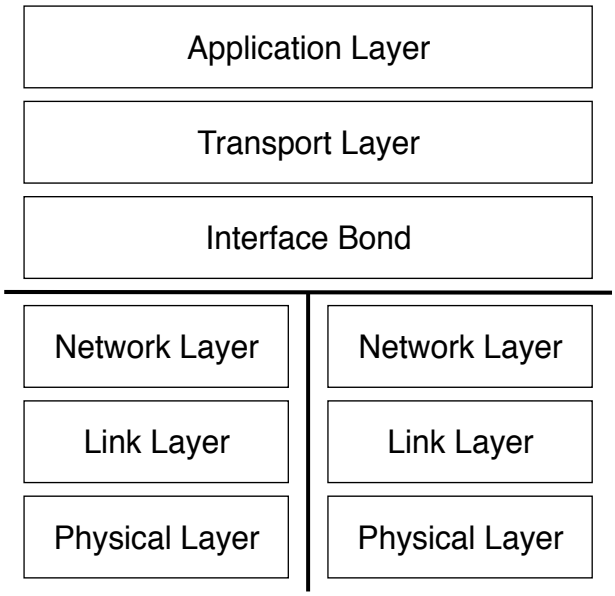

Figure 3.3: Bonding at Layer 3.5 of the Protocol Stack (e.g. pre MPTCP works)

3.2.4.3 Point-to-Point vs. End-to-End Interface Bonding

In addition to the layered categorisation above, interface bonding solutions can be broadly categorised as either point-to-point or end-to-end. The research contained in this thesis is primarily concerned with *point-to-point* link aggregation using dual 802.11 wireless links. Point-to-point link aggregation occurs at the Link Layer between two network devices directly connected via a single hop by at least two individual links, for example as between two adjacent Layer 2 switches as per the LACP Ethernet bonding diagram in Figure 3.6, or as between a multi-homing mobile device and a multi-RAT Wi-Fi access point using Linux Bonding, as illustrated in the diagram in Figure 3.4 below.

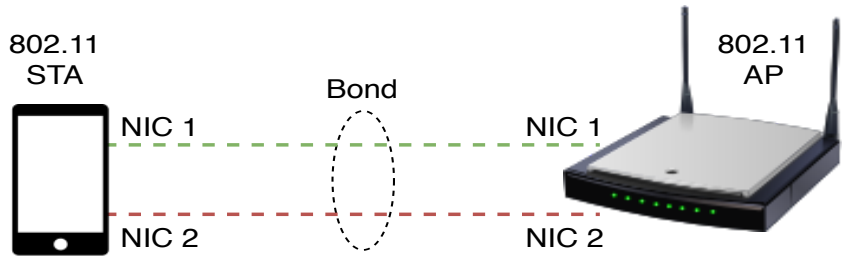

Figure 3.4: Point-to-Point Interface Bonding Example

By contrast end-to-end bonding occurs at Layer 3 and above (commonly at the Transport and Application layers) between two nodes connected via multiple hops across a series of intermediate switches and routers, for example using MPTCP between a cloud-based data server and a mobile client device, as illustrated in Figure 3.5 below. As shown the end-to-end path may consist of both wired and wireless intermediate links. Aggregating proxy servers may also be used by client or server devices which do not have multi-path capabilities, for example an MPTCP proxy.

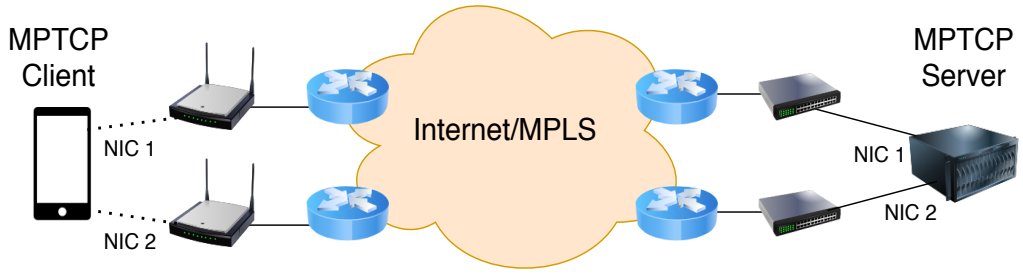

Figure 3.5: End-to-End Interface Bonding Example

3.2.5 Existing Technology and Standards

This subsection provides an overview of the state of the art in interface bonding technologies and standards at all layers of the protocol stack including both wired and wireless types.

3.2.5.1 Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) was originally defined in 802.3ad [8] and provides a mechanism for automatically creating and managing interface bonds between network devices, which are known as Link Aggregation Groups (LAGs) in the protocol nomenclature. The protocol operates at the Link Layer or Layer 2.5 of the stack and targets wired Ethernet links. LACP does not modify the Layer 2 frame format, does not require the use of additional buffers, and does not provide a mechanism for aggregating the individual link capacity - only a redundant configuration for automatic link fail-over is prescribed. It restricts the number of slave links attached to 8 devices and assumes that all links are full-duplex, point-to-point, and share the same raw PHY data rate. A significant aspect of the protocol is the automatic method of bond creation. To setup a LAG between two LACP nodes, for example between two switch devices or between a switch and a server, the individual bond links transmit a Link Aggregation Control Protocol Data Unit (LACPDU). The LACPDU contains a field called the LACP Status, which contains the important bond negotiation information, including an Aggregation flag indicating the link is willing to be used as part of a interface bond. In terms of scheduling, the standard does not define a specific algorithm but states that the method used shall not cause the unnecessary mis-ordering of frames which are part of a single conversation or cause unnecessary frame duplication.

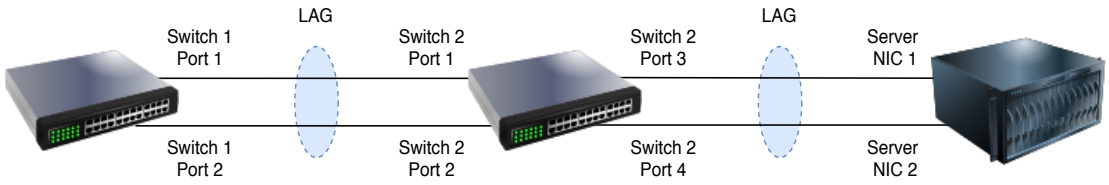

Figure 3.6: Example LACP Bonding Topology

3.2.5.2 Multi Link Point-to-Point Protocol

The Multi Link Point-to-Point Protocol [178] is an extension of the Point-to-Point Protocol [174] that provides a specification for the aggregation of multiple WAN links. MLPPP differs from LACP in that it is designed specifically for WANs and technologies such as ASDL and SONET. Routers or switches with MLPPP enabled advertise their capability using the multilink option during an initial negotiation via the Link Control Protocol. The protocol can be configured in static or dynamic modes: in static mode the number slaves assigned to a bundle remains the same for the life of the bundle, while in dynamic mode links may be dynamically added and removed as and when required, in contrast to LACP.

3.2.5.3 Interface Bonding in Linux

The Linux bonding kernel module [46] provides a partially link technology agnostic method of link aggregation at Layer 2/2.5 of the protocol stack.

The created bond has the name *bond0* and may be configured using *ifconfig* as for any regular interface. Each slave device takes the MAC and IP address of *bond0* to avoid potentially confusing attached switches or routers.

Multiple transmission modes are offered, including round robin, active-backup, 802.3ad (LACP), and broadcast. The default static round robin mode allocates packets between devices in traditional round robin fashion. Active backup mode uses a single slave for transmission and keeps one or more secondary slaves on standby in case of failure. Balanced XOR mode schedules packets according to a hash function to provide load balancing, the default being a simple XOR of source and destination addresses. Broadcast mode transmits all packets on all slaves for frame-level redundancy. The 802.3ad mode is a straightforward implementation of LACP. Adaptive Transmit Load Balancing distributes packets according to current load but requires the use of ethtool [180], and therefore only works effectively for Ethernet. The same applies to the similar Adaptive Load-Balancing mode.

Red Hat introduced a more refined version of the Linux bonding module called the Network Teaming Driver [41] in Red Hat Linux Version 7. The driver has a more modular design compared to regular Linux bonding, with a small kernel based driver handling fast packet flows and user-space applications that perform all other bonding related functionality. Custom scheduling is implemented using additional sections of customisable code called runners. The available transmission modes in the Red Hat teaming driver are broadcast, random, round-robin, active-backup, LACP, and load-balance, for which the functionality corresponds directly to the similarly named modes in Linux bonding. A notable addition is the random runner, which selects an interface at random for each new packet received. As with the Linux bonding most transmission modes in Red Hat Teaming require the use of ethtool to obtain up-to-date link state information, and is therefore largely incompatible with 802.11 devices.

3.2.5.4 Broadband Bonding Services

Broadband bonding services are offered by several operators wherein multiple WAN interfaces are bonded together using a single VPN tunnel and managed at the network level. Communicating end-points appear as if they are on the same LAN segment despite being located on separate physical networks. Speedify is an encrypted mobile VPN capable of bonding multiple interfaces to increase capacity and improve link reliability. Packets are distributed according to the offline capacity of each interface and there is no dynamic load balancing functionality. Mushroom Networks offer a similar Broadband Bonding Service [135] that allows bonding in both the uplink and downlink. Other examples of currently available VPN-based broadband bonding include Peplink SpeedFusion Bonding Technology [143] and Rugged VPN by Viprinet [206].

3.2.5.5 LTE Unlicensed

LTE Unlicensed [83, 93] is a group of 3GPP cellular-based WWAN technologies that extend the use of LTE into the unlicensed ISM bands at 2.4 GHz and 5 GHz. LTE-U uses carrier aggregation to bond together multiple physical channels in both traditional LTE bands and the ISM bands. However, by necessity each band is still subject to its own MAC layer entity and associated processes. LTE-WLAN Aggregation (LWA) differs from LTE-U in that it directly integrates Wi-Fi technology within the LTE infrastructure by bonding links at the Packet Data Convergence Protocol (PDCP) Layer. In this respect LWA is more similar to the interface bonding methods described above. Both of these LTE-based technologies share the common aim of incorporating free-to-use spectral resources into the LTE infrastructure, but differ in the mechanisms used and level of politeness when accessing the shared transmission medium.

3.2.5.6 Channel Bonding in 802.11

The IEEE 802.11n standard [9] defines a method for link aggregation called channel bonding that combines multiple, contiguous 20 MHz channels into a single 40 MHz channel at the PHY layer in the 2.4/5 GHz bands for the purposes of increasing the overall data transfer rate. The IEEE 802.11ac [12] is designed specifically for 5 GHz Wi-Fi devices and extends the channel bonding capabilities of 802.11n to allow bonded channel bandwidths of 80 MHz and 160 MHz. Proprietary technologies include Super G by Atheros, which bonds together two 20 MHz 54 Mbps 802.11g channels to create a single 40 MHz channel up to 108 Mbps. Newer 802.11 standards such as 802.11ax [51] and 802.11ay [234] also include similar extended channel bonding mechanisms.

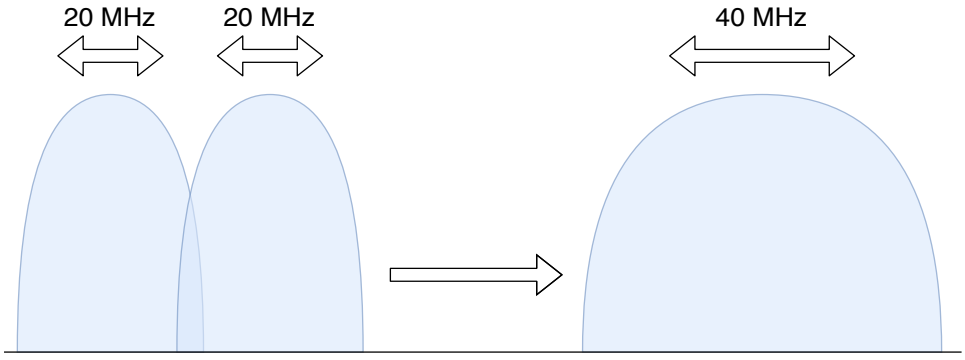

Figure 3.7: Channel Bonding in 802.11n

3.2.5.7 Multi Path Transmission Control Protocol

The Multi Path Transmission Control Protocol (MPTCP) [63] aggregates network interfaces at the Transport Layer by mapping multiple regular TCP flows to a single endpoint presented to the Application layer. The multi-path protocol is written on top of traditional TCP but maintains its own sequence numbering space instead of reusing the individual TCP sequence numbers, which can create losses, delays, and other issues at middle-boxes. To create a MPTCP connection the MPTCP client sends a SYN segment containing the MP_CAPABLE option to the open MPTCP server port. The server then replies using a SYN_ACK segment containing the MP_CAPABLE option, and finally the client responds with an ACK to complete the initial setup. To create additional flows a full handshake is carried out between the server and the new interface using the MP_JOIN option. Sub-flows are removed via the exchange of TCP FIN or RST segments.

3.2.5.8 Stream Control Transmission Protocol

The Stream Control Transmission Protocol (SCTP) [187] is a Transport layer protocol that provides sequencing and congestion control functionality similar to TCP but with additional support for multi-homing and multi-streaming. Multi-homing allows communicating end points to maintain more than a single IP address to provide an instant fail-over capability. The multi-streaming feature of SCTP allows the splitting of data into multiple, individually sequenced the streams, which are designed to prevent delay issues seen in TCP where the delivery of out-of-sequence data is delayed until correct sequencing is restored. However streams are subject to the same congestion and flow control which is in contrast to MPTCP. Although the protocol does provide some support for concurrent streaming over multiple interfaces, the feature is used purely for redundancy purposes and may not be used to aggregate link capacity.

3.2.5.9 Application Layer Approaches

Application Layer bonding solutions aggregate network links at Layer 5 of the OSI model. Habak et al. [79] categorises application layer approaches into network-aware applications and middleware-based solutions. Network-aware solutions are able to tailor the bonding implementation according to the needs of a particular application, but this places an additional burden on application developers and breaks traditional layered networking stack conventions, namely the principle of encapsulation or the hiding of implementation details between different protocol layers. Habak et al. sub-categorise middleware based solutions into hidden and palpable types. Hidden middleware use the same interfaces and semantics as Transport layer protocols and are therefore transparent, for example DBAS [82] and Operetta [80]. By contrast, palpable middleware requires a change to the service agreement between the Application and Transport layers for example MuniSocket [121].

3.2.6 Technical Challenges in the State of the Art

The main challenge for interface bonding is the inherent slow-down effect due to the in-order delivery requirement. This is caused by the heterogeneous throughput performance of individual wireless links and is exacerbated by the stochastic access delay of the 802.11 MAC mechanism. While traditional static interface selection and round robin frame scheduling work perfectly well for wired devices such as Ethernet, when used over wireless links a significant overall bond throughput slow-down effect is observed due to the in-order delivery requirement in the upper protocol layers, for example in the connection-oriented Transmission Control Protocol (TCP).

When an in-order delivery requirement is assumed the overall bond throughput becomes limited by the throughput of the slowest link of the bond as all application data frames must be successfully received over both bonded links before the structured data can be passed to the upper layer. Even for two wireless links with the same offline data rate, when one bonded link is more congested than the other it is not able to process incoming frames as quickly as its counterpart. In addition this also causes further slow-down at the Transport Layer due to out-of-order and duplicate frames at the receiver and the reaction of the TCP congestion control mechanism at the sender. Delayed frames are misinterpreted by TCP as congestion which it responds to by reducing the contention window size, thereby further reducing the overall achieved aggregate throughput.

This is a major issue impacting the bond re-sequencing unit for Layer 2 solutions when aggregating over multiple and diverse wireless interfaces and has been the subject of many related research works, for example So et al. in [180]. The use of wireless slave devices also causes difficulties for the interface selector, the bond scheduler, and the link monitoring unit as discussed in the following subsections.

3.2.6.1 Interface Selection

In most current Layer 2 and Layer 4 bonding solutions such as Linux bond and MPTCP, the bond interface selection and link attachment is performed via traditional static and manual configuration which requires the attendance of a skilled network engineer. This approach is acceptable in wired networks, for example multiple LAGs configured between leaf and spine nodes in a data center environment where all communication links are of the same data transfer speed.

However for wireless devices such as 802.11 and as demonstrated in the MatLab simulations later in the chapter, the static method can lead to severe throughput degradation if a poorly chosen interface combination is used, as the bond is unable to adapt dynamically to changes in the achievable link data rate and latency. LACP can assist in the negotiation of bonded communications between adjacent LACP-enabled network devices, but the protocol itself does not prescribe any specific method for dynamically choosing which links should be added or removed from the bond and when, based on the current link state.

3.2.6.2 Bond Scheduling

As mentioned above, the traditional static bond scheduling algorithms such as round robin work well over links with similar throughput and delay, but as demonstrated in multiple research works and also later in the chapter in the results of the static round robin bonding simulations, this leads to a poor aggregated throughput performance The problem is of particular concern for wireless networks in which such delays are considered the norm. A large number of research works have targeted the bond scheduling problem in recent years as described in the related work in Section 3.2.8 below.

3.2.6.3 Wireless Link Monitoring

In most bonding solutions such as Linux bonding and MPTCP the direct monitoring of wired Ethernet links is performed using ethtool [180] for which there is no equivalent for 802.11 devices. Bonding can still be used over wireless interfaces but with severe performance degradation. When used over wireless in the absence of ethtool, the default scheduling in the Linux kernel bonding module is round robin. Efficient interface bonding with wireless devices such as 802.11 therefore requires some other passive method of obtaining the current link statistics, such as the average link throughput or frame inter-arrival times, in order to adequately inform the other bond functions such as the scheduler of interface selection unit.

3.2.7 Recent Efforts Towards Standardisation

IEEE published the Real Time Applications TG report [95] in November 2018 which made recommendations for two different packet-based transmission modes supporting multiple uplink applications:

1. *Duplicate Mode:* similar to broadcast mode in Linux bonding.
2. *Joint Mode:* this corresponds to round-robin mode in Linux.

The report also identified two design challenges that the new transmission modes would be required to overcome:

1. *Traffic Load Balancing:* a dynamic method of load balancing was needed to assign traffic to each wireless link according to available capacity.
2. *Packet Alignment:* Joint Mode causes out-of-order packets, and such losses must be differentiated from any loss at the MAC layer.

Work on 802.11be was formally initiated in May 2019. The new specification would implement the multi-link recommendations of the RTA TG as well as providing the typical generational increases in raw PHY speed. Furthermore, coordinated TXPOs were proposed as a method of aligning multi-link transmissions.

Relevant articles and research published in late 2019 such as [25] and [219] began to explore issues such as AP coordination, but contained little technical detail on the proposed multi-link mechanism.

Further clarity came later with surveys and updates in May-August 2020 which contained the first references to the Multi-Link Device (MLD) concept and synchronous/asynchronous transmission types. In particular the survey published by author Evgeny Khorov (an original members of the RTA TG) in [101] shed light on the efficiency problems with the traditional approach to exploiting wide channels in the 802.11 OFDM PHY. Existing off-the-shelf APs support dual/tri band aggregation but the operation of the individual PHY/MAC layers is entirely independent, and this leads to inefficiencies because individual sub-bands can have different requirements in terms of channel access, for example non-identical DCF parameters and varying link throughput performance. To make things even worse, the channel access is determined by the primary 20 MHz channel so the entire bonded channel can be blocked if the primary channel is busy.

The MLD is introduced 11be to address the aforementioned issues by allowing the transmission of packets concurrently on multiple individual channels. A MLD may be viewed as a mapping of two or more 802.11 MAC/PHY interfaces that presents a single unified interface to the LLC Layer (the upper L2 protocol used with 802.11 and Ethernet), so as to contain the

bonding functions entirely within Layer 2. This contrasts with Linux where bonds are created by mapping multiple 802.11 NICs to a single virtual NIC inside the kernel network stack. Here bonding occurs above the LLC and is considered a Layer 2.5 approach. Additionally, in Linux there is no method of TXOP synchronisation.

While the thesis work anticipated the benefits of a packet-based approach to bonding at Layer 2, unfortunately the initial decisions on research direction and the design of the MatLab bonding simulator were taken before the recent insight from the 11be Working Group and related publications. Note that since the completion of the thesis in January 2021, significant strides have been made in terms of analysing the performance of MLDs in 11be. Notably, the authors Song and Kim in [182] evaluate and compare the throughput and fairness of different transmission modes and proposed coordinated access techniques using a bi-dimensional Markov chain based on the single-link DCF modelling by Bianchi.

3.2.8 Related Research Work

This subsection provides an overview of related interface bonding research categorised according to the relevant protocol layer with both wireless and wired applications considered.

Several link aggregation surveys are found in the research literature, including Habak et al. [79] and Ramaboli et al. [150]. However, both of these surveys were published before the introduction of MPTCP and since its release there has been a wealth of new link aggregation research and insight. Therefore some of the content in these surveys is now considered obsolete, but they still provide good general background information.

3.2.8.1 Link Layer

The Link Layer bonding solutions such as Linux bonding are the primary focus of the thesis research and are therefore discussed first. A key advantage of the kernel-based interface bonding types such as Linux over the deeply integrated types such as 802.11n channel bonding is that the implementation can be entirely agnostic with respect to the underlying Link Layer technology used. A good example of such an approach is the round robin transmission mode found in Linux bonding, which requires no link monitoring capability or specialist device driver interfaces whatsoever.

However, a significant disadvantage is that the technique is nowhere near as efficient as the more technically advanced channel bonding mechanisms such as 802.11n channel bonding and LTE-U carrier aggregation at the Physical Layer. In the work in [171] Shidik et al. compare the throughput achieved via 802.11n-based channel bonding to the throughput achieved by Linux-based interface bonding over dual-11n links. The authors found that 802.11n channel bonding significantly outperformed interface bonding of terms of the aggregate throughput. In fact the achieved interface bonding throughput was actually lower than the individual single 802.11n link throughput, which the authors concluded was due to the use of static round round scheduling leading to a substantial slow-down effect and high number of out-of-order and duplicated frames.

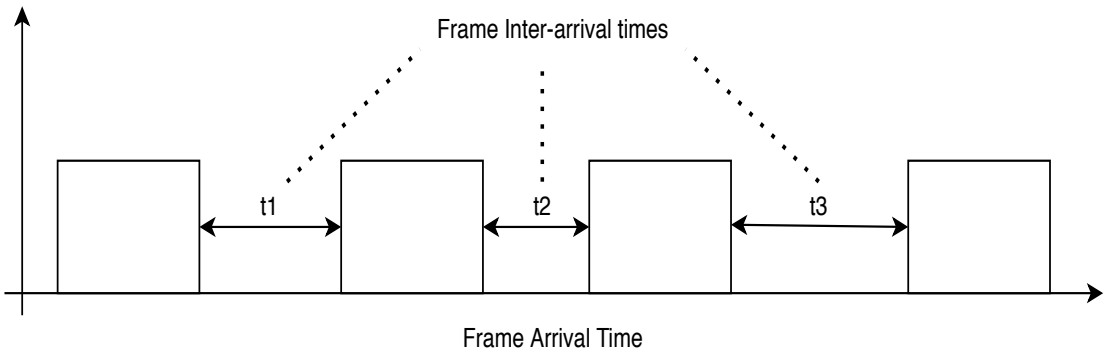

Figure 3.8: Frame Inter-Arrival Times used by So et al. [180]

A number of other research works also address Layer 2 bonding over wireless in Linux. The most notable example is that by So et al. [180] who propose a New Load Balancing (NLB) kernel module designed specifically for use with wireless devices such as 802.11. The in-order delivery of frames is achieved via a receiver side buffer, a second tier numbering system, and a fast loss detection scheme. Dynamic scheduling is performed in technology agnostic fashion using receiver-based in-line measurements of incoming frame inter-arrival times. The hardware-based test results showed a substantial improvement in bond throughput compared to traditional, static, round robin-based approaches. The solution provides automatic bond negotiation with other NLB-enabled nodes via periodic HELLO messages, but the interface selection itself is manual and static (i.e. dynamic bond scheduling only).

The work by Jarasuriya et al. in [92] experimentally evaluates the throughput performance of static round robin bonding in Linux using dual 802.11b interfaces for the same point-to-point topology as assumed in the implemented MatLab simulation presented later. The results show that short term variations in the individual packet latencies leads to unstable behaviour for TCP and UDP traffic. Furthermore the instability is also seen when both 802.11 links are statistically identical, i.e. have the same throughput, delay, and frame collision probabilities. The test results take a top-down approach by focusing on the interplay between the different protocol layers, in particular the impact of the heterogeneous 802.11 bond link delays on the Transport Layer. This contrasts with the bottom-up approach of the thesis research which measures the bond throughput at the MAC layer and therefore ignores any possible impact at the Network or Transport layers to focus on the effects of the DCF. The thesis work also differs in that only the long-term throughput performance of the bond is used in the analysis and delay variations and jitter occurring in the short-term are not considered. Because of these differences a direct comparison of results of the two works is not possible.

Other examples of Linux bonding research works include Miura et al. [120], which uses Linux bonding for high bandwidth and fault tolerance on server clusters, Rico et al. [155] which compares Linux bonding performance with MPTCP, and Saud et al. [162] which uses Linux bonding to create a hybrid wireless-optical link aggregation system. Kim et al. proposed a feedback-based traffic splitting scheme for multi-radio mobile devices in [102] that mitigates the level of packet reordering required at the receiver. In the proposed system the delay of each wireless link is measured periodically at the receiver using an active probing technique, which is then passed back to the sender which uses the information in a dynamic scheduling algorithm. The idea behind the proposed algorithm is to split traffic so that the difference in the delay of between links is minimised, which has the effect of reducing the number of out-of-order packets and increasing the overall bond throughput. The scheme was tested on a hardware testbed using Wi-Fi and WiMAX wireless interfaces and was shown to significantly increase throughput and reduce receiver buffer size.

In recent years one of the biggest research problems in WWAN-WLAN link aggregation such as LTE-WLAN Aggregation (LWA) has been the coexistence of different wireless technologies in the unlicensed portions of the radio frequency spectrum. When using free-to-use ISM bands such as 2.4. GHz in certain regulatory domains such as the UK, vendors and operators must adhere to the so-called listen-before-talk (LBT) requirement, which prescribes that computing wireless stations must always sense the medium first before transmitting in order to prevent collisions occurring on the shared channel and ensure fair access between different device types. The 802.11 WLAN standard already satisfies this requirement by virtue of its DCF MAC mechanism, but a number of works have demonstrated the detrimental effect of the more aggressive channel access strategy used by LTE-U on adjacent legacy 802.11 networks, for example [83]. This issue has been the cause of a well-documented debate between the Wi-Fi Alliance and 3GPP over fair use of ISM resources. Several solutions to the coexistence problem have been proposed, such as that by Wang et al. [212], which is based on cognitive channel switching and adaptive muting, and the Q-learning approach adopted by the authors Su et al. in [188].

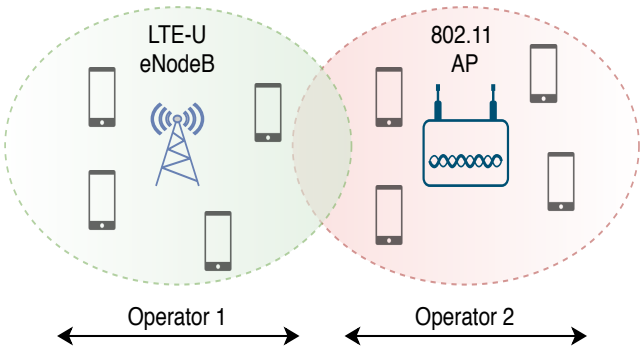

Figure 3.9: 802.11 WLAN and LTE-U WWAN Coexistence in the 2.4. GHz ISM Band

3.2.8.2 Network Layer

Most network layer solutions target the use of TCP at the Transport Layer and were published before the release of MPTCP in 2013. A TCP connection is identified by a set of source and destination IP addresses and port numbers, and therefore distributing packets from a single stream over multiple interfaces breaks the traditional TCP connection semantics. Oscar et al. [81] solve this by rewriting the source and destination addresses before passing the packet to TCP. Other works for example, Chebrolu et al. [39], which uses aggregating proxies to provide transparency to the Transport Layer. However, most of these approaches do not allow independent congestion control of sub-flows, which was a key design advantage of MPTCP. Many of these pre-MPTCP Network Layer bonding approaches such as the those described above and the various proposals surveyed by Habak may therefore be better understood as Layer 3.5 approaches and characterised by a lack of independent congestion control.

3.2.8.3 Transport Layer

Much recent link aggregation research has targeted MPTCP. For example, Ageneau et al. [16] examines wireless link aggregation in MPTCP using network coding applied at Layer 3. Emulation results show that this significantly improved bond throughput without any alteration of the MPTCP stack. The authors Rao [151] demonstrate the use of MPTCP to improve communication reliability and throughput in robotic systems. Park et al. [141] use a receiver-centric approach for minimising application-level delay for MPTCP in wireless networks. Wang et al. [209] highlight the importance of proper path selection selection for MPTCP using the NORNET testbed. The performance impact of receiver buffer size is examined by Zhou et al. in [233]. Hwang et al. [90] propose a method for MPTCP scheduling that freezes the slow link when the difference in delay across the bond is large. Test results show that the proposed scheduling algorithm significantly reduces flow completion time for short-lived sub-flows. A energy efficient MPTCP solution was proposed by Shamani et al. [167] where the entropy-based approach is shown to reduce energy consumption by up to 42% for large data transfers.

3.2.8.4 Application Layer

In [105] the authors Krishna et al. propose a dynamic link aggregation scheme called DLAS for use over wireless links such as 802.11. There are two types of DLAS scheme proposed in the work, Sequential DLAS and Parallel DLAS. In Sequential DLAS, the mobile device requests a number of data chunks through each available interface. The receiver then calculates the achieved throughput of each interface and requests further data chunks in proportion. Parallel DLAS uses parallel TCP sub-flows within the same data flow to increase the overall data transfer rate. The proposed system was shown to improve aggregate throughput by 9-10% compared to similar solutions. An Application Layer aggregation scheme implemented in the Android operating

system is proposed in [193] in which the overlay network concept is adopted. Both individual network interfaces run separate TCP/IP stacks with congestion control applied individually on each link. Because applications cannot directly invoke device driver functions, the Android runtime is used to issue the required kernel functions. The packet scheduling for each client is determined by the data distribution ratio sent to the distribution servers which changes dynamically according to the measured throughput. Distribution ratio updates are sent via all available interfaces for redundancy.

3.3 Bonding Simulator Implementation

This section presents the novel simulation evaluation framework for 802.11 interface bonding implemented using the MatLab programming language.

The bonding framework is built on top of the single-link simulator for the 802.11 DCF described and validated in Chapter 3 and is further extended in Chapter 4 to facilitate the evaluation of state-of-the-art link selection algorithms. The simulator output provides the aggregate throughput and delay of a dual-link 802.11 interface bond at Layer 2 as a function of the applied bond load given as the number of competing stations present in each WLAN under the assumption of saturated arrival traffic. Other metrics provided include the link throughput difference, link delay difference, and the link bandwidth utilisation of the bond.

3.3.1 Practical 802.11 Bonding Use Case

The practical 802.11 interface bonding use case considered in the implemented framework is illustrated in Figure 3.10 below in which a multi-homing WLAN branch user is connected via multiple single-hop, point-to-point wireless links to an multi-RAT, SDN-enabled customer edge device used to communicate with other branch locations and with private or public clouds hosted on the Internet. Note that the above point-to-point topology was chosen to match that used in most Layer 2 802.11 interface bonding researches such as those in [92, 171, 180].

Under this scenario there are several ways in which wireless users can exploit the multi-connectivity of the branch node over the wireless last hop: (1) aggregating the capacity of the individual wireless links to increase the overall data transfer speed, (2) broadcasting data frames over all available links to increase reliability and reduce latency, and (3) using both available network interfaces in a redundancy or active-standby configuration in case of hardware or link failure.

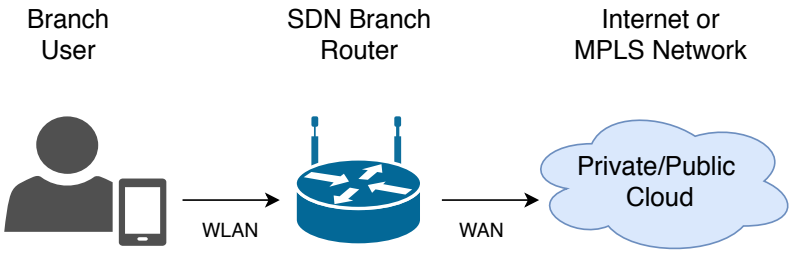

Figure 3.10: SD-WAN/SD-WLAN Interface Bonding Use Case

Many vendors already provide single-link WLAN connectivity for branch users in small to medium sized enterprise and campus networks. For example, the Cisco Meraki MX Series SD-WAN branch routers provide both 11ac and legacy 11n on-board connectivity options (using a single hardware interface) in addition to integrated LTE-based WAN-side connectivity. Other state-of-the-art examples of SDN-enabled branch devices with 802.11 WLAN connectivity include the VMWare VeloCloud SD-WAN Edge router and the Nokia-Nuage Network Services Gateway.

3.3.2 Categorisation of Interface Bond Type

As mentioned above the interface bonds in the implemented framework are formed at the Link Layer, or more precisely Layer 2.5, of the protocol stack such that both individual enslaved interfaces are presented as a single unified resource to the Network Layer and above and that the bonding is performed in a manner that is agnostic to the underlying link layer technology and does not require any direct device driver intervention. The layered categorisation used is illustrated in the diagram below in Figure 3.11.

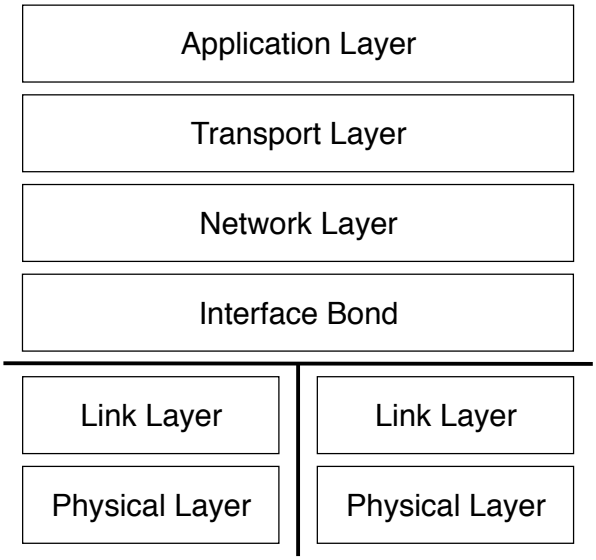

Figure 3.11: Bonding at Layer 2.5 of the Protocol Stack (e.g. LACP)

3.3.3 Bonding Functionality Implemented

Using the classification given by Habak et al. in [79] illustrated in Figure 3.1 above, in the implemented framework the required bonding functionality is implemented on top of two parallel single-link DCF simulations as follows:

- Scheduler: The scheduling algorithm used in the developed framework is traditional round robin wherein all enslaved bond links are served with a single frame per transmission round. This is a commonly used static form of bond scheduling found in most current implementations, for example MPTCP and Linux bonding, where the frame allocation is fixed for the entire duration of the simulation. In the simulations this is modelled by filling the transmit buffers of each individual slave link with half of the number total frames assigned to the bond node.
- Interface Selector: In this chapter in order to evaluate the performance of the traditional manual link selection approach, a static bond interface selection method is used. As per the round robin scheduling algorithm the 802.11 bond interface configuration is fixed and does not change for the entire duration of the simulation regardless of the current individual link state and overall achieved aggregated throughput performance of the bond.
- Re-assembler: The reassembly is responsible for the buffering and re-ordering of the transmitted data received over all individual slave links attached to the bond. In the simulation a simplistic re-assembler is modeled with an infinite size receiver buffer and the re-structuring of transmitted payload data is not considered, only the rate at which the data is received over both links.
- Link Monitoring: Due to the use of static round robin scheduling and a fixed bond interface configuration there is no need to monitor the instantaneous state of the individual links, and therefore no link monitoring functionality is implemented in the bonding evaluation framework presented in the current chapter.

3.3.4 Bonding Simulation Assumptions

In addition to the assumptions for the single-link DCF simulations described in Section 2.4.1.1 of Chapter 2, a further series of assumptions are made with respect to the extended interface bonding functionality implemented in the current chapter. These additional assumptions are described in the following subsections.

3.3.4.1 Interface Bond Type

In the implemented framework a dual-802.11 statically configured round robin interface bond is assumed similar to the types used in the experimental works using Linux bonding by Jarasuriya [92], So [180] and Shidik [171]. Importantly an in-order delivery requirement is assumed in the upper protocol layers, for example as with a connection-oriented Transport Layer protocol such as TCP, in order to evaluate the bond throughput performance and associated slow-down under this commonly found constraint.

3.3.4.2 Logical Bonding Topology

The practical use case described previously is translated into the logical bonding topology shown in the diagram below in Figure 3.12. A single-hop, point-to-point bond is assumed between a dual-interface 802.11 access point and a multi-homed mobile user node also similar to that used by the authors by Jarasuriya [92], So [180] and Shidik [171]. Also, as per the assumptions for the single-link DCF simulations in Chapter 2, a flat single-hop wireless topology is assumed for each separate 802.11 network.

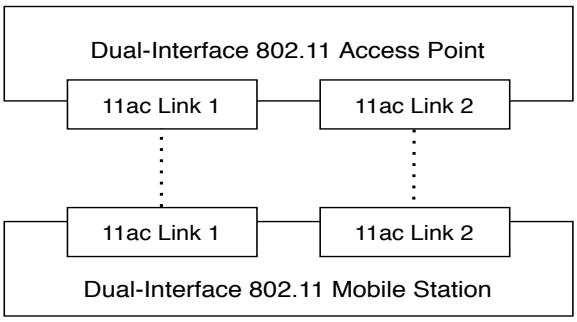

Figure 3.12: Logical Topology Assumed in the Bonding Simulations

3.3.4.3 Independent 802.11 Interfaces

Each individual 802.11 slave link attached to the interface bond is assumed to be statistically independent. Because the PHY layer is not modelled in the implemented framework this essentially means that the probability of two interfaces simultaneously experiencing a collision on the channel is independent, i.e. $p = p_1 p_2$. That is to say that collisions occurring on link does not impact or influence the collision likelihood seen on the other bond link. This is satisfied in the following homogeneous 11ac-11ac bond testing by assuming disjoint physical channels for each similar bonded link.

3.3.5 Calculation of Bond Performance Metrics

In order to measure the bond throughput in the simulations and determine the impact of the inherent slow-down for heterogeneous link performance, the following bond related performance metrics are measured.

3.3.5.1 Maximum Bond Throughput

This is the maximum theoretical throughput of the bond with no slow-down effect present and is given in Mbps and calculated as the sum of the throughputs of the individual bonded 802.11 links:

$$S_{Max} = S_n^1 + S_n^2 \tag{3.1}$$

3.3.5.2 Achieved Bond Throughput

This is the achieved bond throughput in units of Mbps and calculated by counting the number of payload bits received over *both* links and dividing by the maximum link transmission time:

$$S_B = \frac{BitsRcvd_1 + BitsRcvd_2}{Max(TxTime_1, TxTime_2)} \tag{3.2}$$

3.3.5.3 Link Throughput Difference

This metric provides the absolute difference in achieved throughput between the individual 802.11 links attached to the bond as function of the applied load and is given in units of Mbps:

$$S_{Diff} = abs(S_n^1 - S_n^2) \tag{3.3}$$

3.3.5.4 Link Bandwidth Utilisation

This gives an indicator of the size of the slow-down effect in relation to the theoretical maximum bond throughput, and is calculated by dividing the achieved bond throughput by the maximum throughput:

$$S_{Util} = \frac{S_B}{S_{Max}} * 100 \tag{3.4}$$

When the resulting utilisation is less than 100% this indicates that a throughput slow-down effect is present and its magnitude is in proportion to the value of the metric. When equal to 100% this indicates the ideal conditions for the bond throughput with no observable throughput slow-down effect.

3.3.5.5 Bond Delay

The overall bond delay is given in units of μs and calculated as the maximum of the achieved access delays of the individual bonded 802.11 links:

$$D_{Bond} = max(D_1, D_2) \tag{3.5}$$

3.3.5.6 Link Delay Difference

The link delay difference is given in units of μs and provides the absolute difference in access delay between the individual links a function of the applied load:

$$D_{Diff} = abs(D_1 - D_2) \tag{3.6}$$

3.3.6 Simulation Operation

A diagram illustrating the operation of the MatLab-based 802.11 bonding simulation is shown in Figure 3.13 below for a single transmission round of round robin scheduling. In the implemented framework two independent DCF simulations are run in parallel for each 802.11 link attached. It is assumed that the timing of both links is synchronised at the start of the simulation. Each link receives one of the total $2M$ number of frame for immediate transmission on its respective DCF access channel. Then as per the DCF CSMA/CA algorithm both links calculate their individual random waiting times before first attempting to transmit their frames.

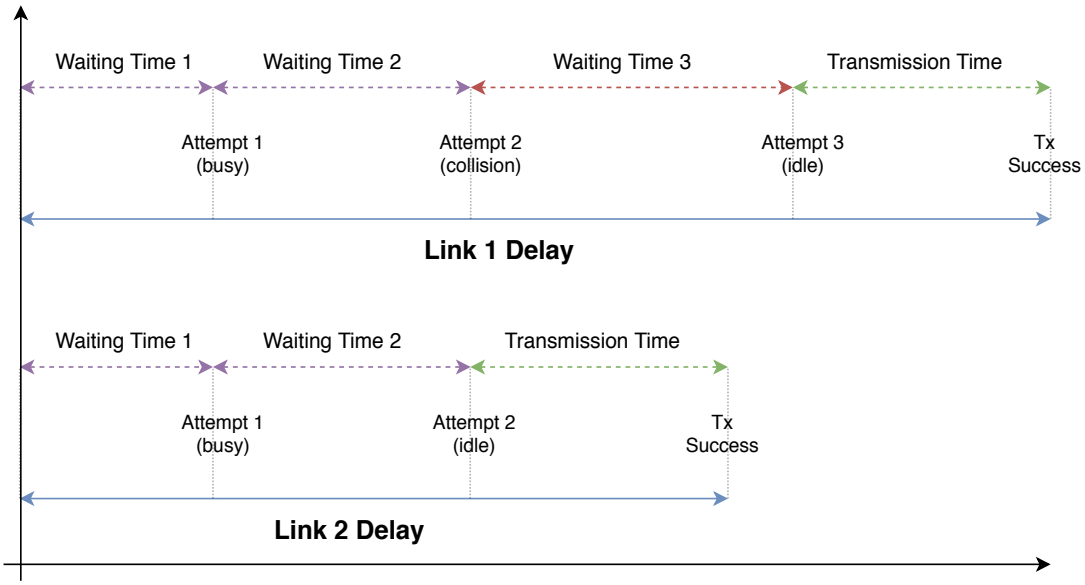

Figure 3.13: Illustration of MatLab DCF Bonding Simulation

In the example above, Link 2 is the first to successfully transmit. After the initial waiting time it attempts transmission but senses the channel as busy due to an ongoing transmission. In response the link chooses a new waiting time from the same range and waits for its next opportunity. When the second waiting period has elapsed, Link 2 senses that the channel is idle and re-attempts transmission which is successful.

By contrast Link 1 requires three attempts to access the channel before its frame is transmitted resulting in a longer overall channel access delay and lower link throughput. At the first transmission attempt the channel is found to be busy and the link chooses a new random waiting time from the same contention window range and re-contends for channel access. On the second effort the channel is sensed as idle and the link tries to transmit but experiences a collision due to another simultaneous transmission attempt. In response it increments its binary exponent

and selects a new random waiting time. At the third and final attempt Link 1 wins contention and successfully transmits its frame. For the simple example above the bond throughput is calculated by counting the number of payload bits received in both frames and then dividing by the maximum of the two transmission times.

3.4 Experiment Setup

This section gives the various simulation input parameters used and describes the different applied load tests performed in the following evaluation of the statically configured 802.11 interface bonds.

3.4.1 Input Parameters

The various input parameters used in the static round robin evaluation are summarised below in Table 3.1. These are the same as per the 11ac vs 11ah comparison in Section 2.3.3.4 of the previous chapter, with the exception of the payload size which is increased to 16386 bits. The MCS settings used at the PHY layer are also the same as those used previously, i.e. MCS Index 6 for the 11ac link and MCS Index 7 for the 11ah which both provide a raw PHY data rate of 65 Mbps. The high total number of nodes in each individual 802.11 of n = 100 was chosen to simulate the densely populated and highly contentious environment expected in future wireless networks and the Internet of Things (IoT).

DCF/PHY Parameter	**11ac Value**	**11ah Value**
Number of Nodes (N)	100	100
Slot Time (us)	9	52
CWmin (slots)	15	15
CWmax (slots)	1023	1023
Payload Size (bits)	16368	16368
MAC Header Size (bits)	288	288
PHY Header Size (bits)	192	192
ACK Size (bits)	112	112
SIFS Duration	16	160
DIFS Duration	34	264
Propagation Time (us)	1	6
PHY Data Rate (Mbps)	65	65

Table 3.1: Table of DCF and PHY Parameters used in Static Round Robin Tests

3.4.2 Dual 802.11 Bond Test Scenarios

In order to replicate the conditions in a practical 802.11 bonding situation, multiple individual tests are performed in which the long-term throughput, link bandwidth utilisation, and delay of the bond are evaluated under different combinations of applied link load.

As summarised in Table 3.2 below the homogeneous 11ac-11ac bond is evaluated under four individual applied load scenarios labelled 1A through 1D and likewise the heterogeneous 11ac-11ah bond is evaluated in under similar applied loads in tests labelled 2A-2D.

For each sub-test a total of $2M$ DCF frames and 20.48 GB are transmitted by all competing stations in each 802.11 WLAN. Once the $2M$ number of frames have been successfully transmitted the applied load is then adjusted and the process repeated until all load combinations prescribed for the test are evaluated.

The four basic applied load tests A-D for each bond type are sub-categorised into balanced and unbalanced load types. To provide a performance bench mark a balanced load test was performed which creates a non-stress scenario designed to yield the maximum possible bond throughput. Then in order to stress the bond throughput and evaluate the impact of the throughput slow-down effect three unbalanced load tests are performed wherein the applied load on the first bond link is varied while maintaining a constant low, moderate, and high applied load respectively on the second bond link. All eight tests are discussed individually on the following pages.

	Applied Bond Load
Test 1A	$\{n_1 = n_2 = n\}$
Test 1B	$\{n_1 = n,\ n_2 = 1\}$
Test 1C	$\{n_1 = n,\ n_2 = 50\}$
Test 1D	$\{n_1 = n,\ n_2 = 100\}$
Test 2A	$\{n_{11ac} = n_{11ah} = n\}$
Test 2B	$\{n_{11ac} = n,\ n_{11ah} = 1\}$
Test 2C	$\{n_{11ac} = n,\ n_{11ah} = 50\}$
Test 2D	$\{n_{11ac} = n,\ n_{11ah} = 100\}$

Table 3.2: Summary of Link Load Tests Performed

The maximum considered number of $N = 100$ competing nodes was chosen to demonstrate the bond performance under heavily loaded and highly contentious network conditions as expected in large next generation 802.11 WLANs such as wireless sensor networks and IoT. The light load of n = 1, moderate load of n = 50, and the heavy load of n = 100 used in the unbalanced tests correspond roughly with the conditions used in the testing by Nam in [130] to evaluate dynamic MPTCP link selection in a small-scale 802.11 testbed. However in the Nam work work the individual achievable link performance was manipulated artificially using Linux Traffic Control with Network Emulation instead of increasing channel contention by incrementing the

number of competing wireless stations as in the developed MatLab framework. The work by Nam and the comparisons to the current research are discussed in further detail in Chapter 4.

3.4.2.1 Homogeneous 11ac-11ac Bond

First we present the channel load tests performed during the static round robin configuration evaluation for the homogeneous 11ac-11ac interface bond.

3.4.2.1.1 RR Test 1A: Balanced Load Scenario The first test was the benchmark balanced load scenario illustrated in Figure 3.14 on the opposite page. At each iteration the number of competing wireless nodes present in both 11ac networks was incremented simultaneously from 1 to the maximum considered value of $N = 100$. Related performance metrics such as the mean bond throughput and delay are calculated at each stage and appended to an output data vector in MatLab. The balanced load scenario was intended to create the ideal conditions for maximising throughput and act as a performance benchmark against which to compare the results from the unbalanced load scenarios and heterogeneous 11ac-11ah bond tests.

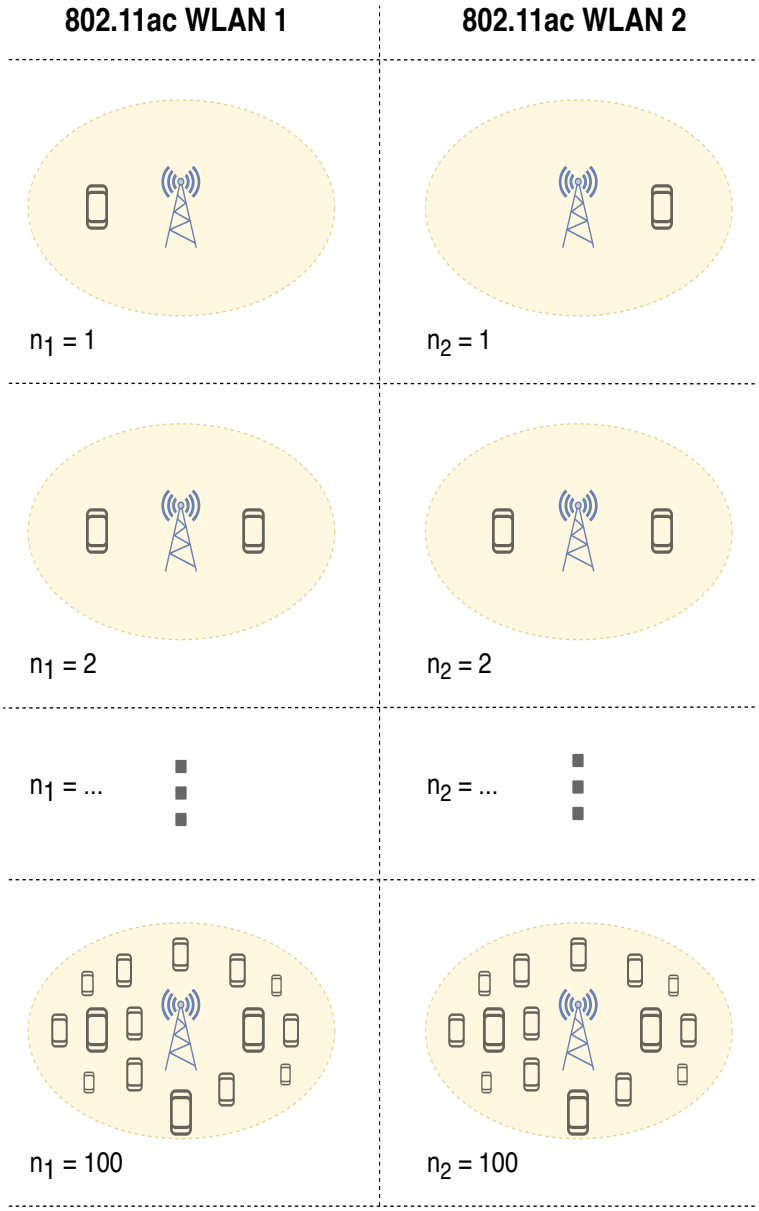

Figure 3.14: RR Test 1A 11ac-11ac Bond Balanced Load Scenario

3.4.2.1.2 RR Test 1B: Unbalanced Scenario (Low Load) The next test was the first unbalanced load scenario for the dual-11ac bond designed to stress the aggregated throughput performance using a light static load on the second 11ac link while varying the load on the first 11ac link. In the implemented framework this was done by keeping the first link load constant at one wireless node while the load on the second link was increased by incrementing the number of competing stations from 1 to the maximum of $N = 100$ as illustrated in Figure 3.15 below.

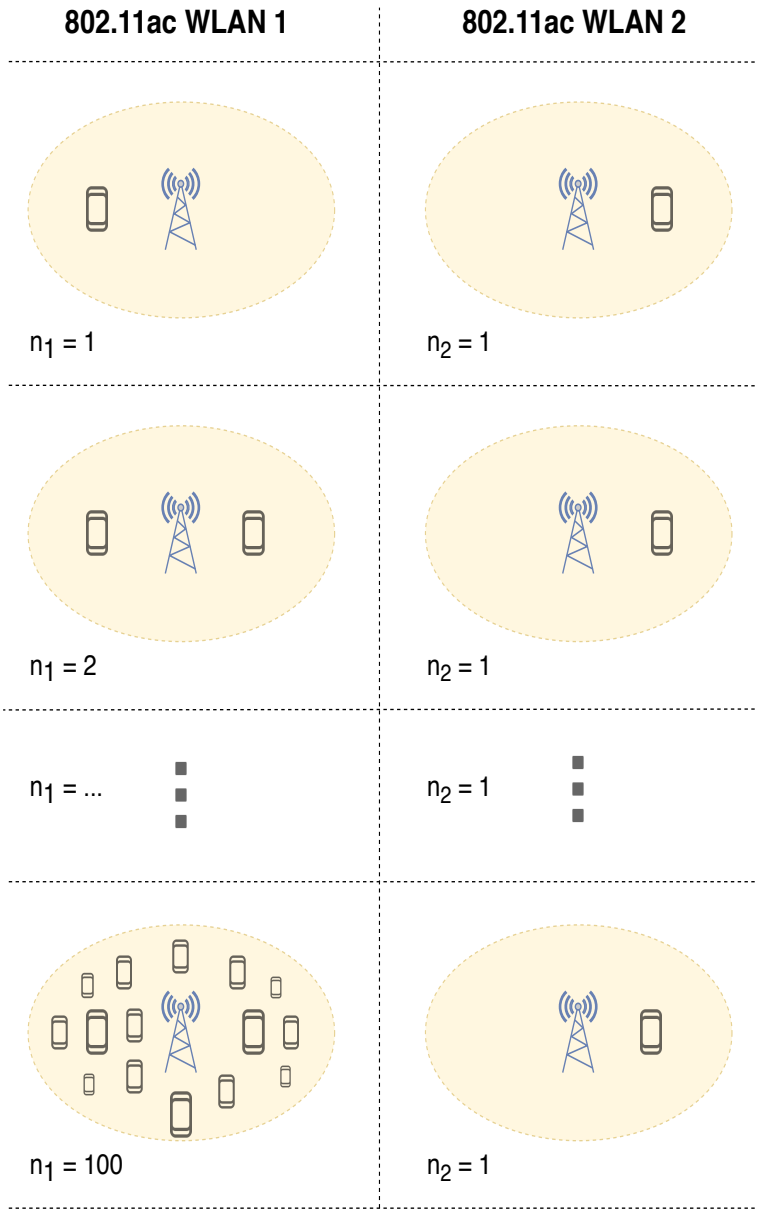

Figure 3.15: RR Test 1B 11ac-11ac Bond Unbalanced Load Scenario (Low Load)

3.4.2.1.3 RR Test 1C: Unbalanced Scenario (Moderate Load) Next was the second unbalanced load test for the homogeneous 11ac-11ac bond also designed to stress the compound throughput this time using a moderate static load on the second 11ac link. In similar fashion to the previous test the applied load on the second link was held constant at n = 50 nodes while the load on the first link was incremented from $n = 1$ to $n = 100$ as illustrated below in Figure 3.16.

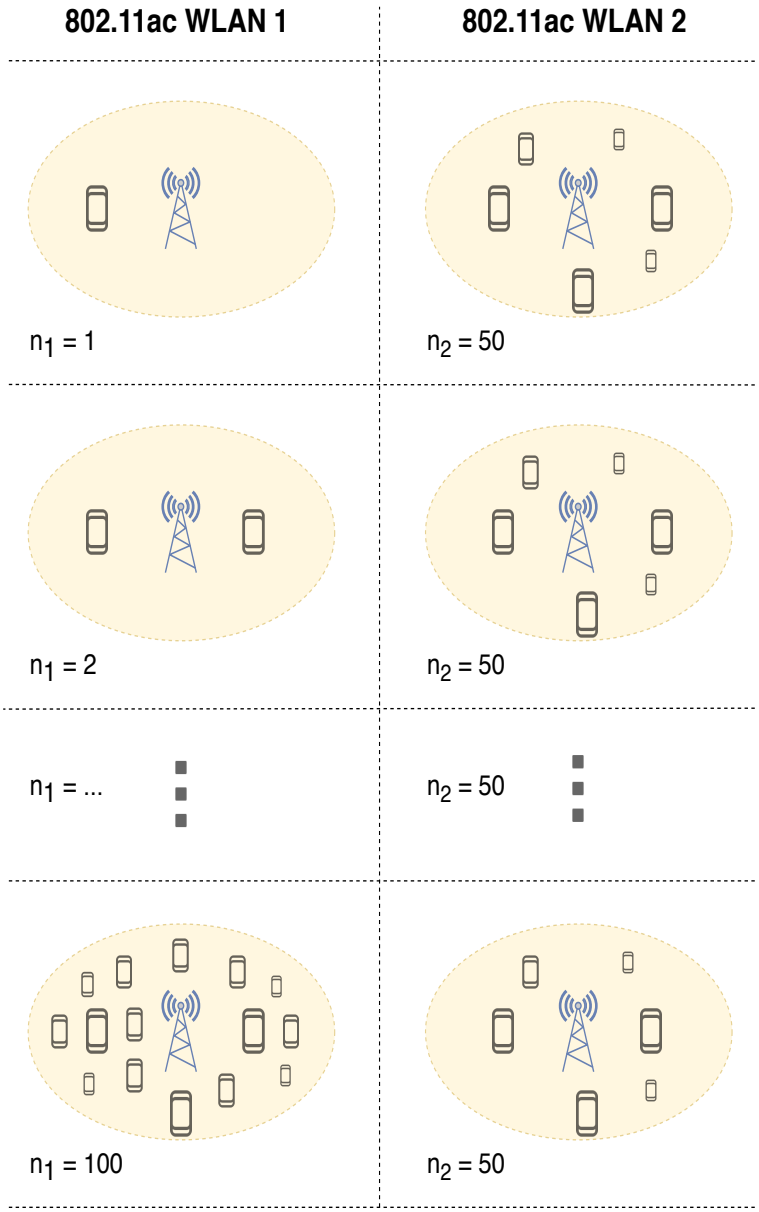

Figure 3.16: RR Test 1C 11ac-11ac Bond Unbalanced Load Scenario (Moderate Load)

3.4.2.1.4 RR Test 1D: Unbalanced Scenario (Heavy Load) The final 11ac-11ac bond test was the third unbalanced scenario illustrated in Figure 3.17 below. During the test the static load on the second 11ac link was set to the maximum value of n = 100 competing nodes while incrementing the load on the first 11ac link from n = 1 to n = 100 nodes. The goal again was to stress and evaluate the overall bond throughput performance this time using the heaviest possible static load of 100 competing wireless stations on the second 11ac link.

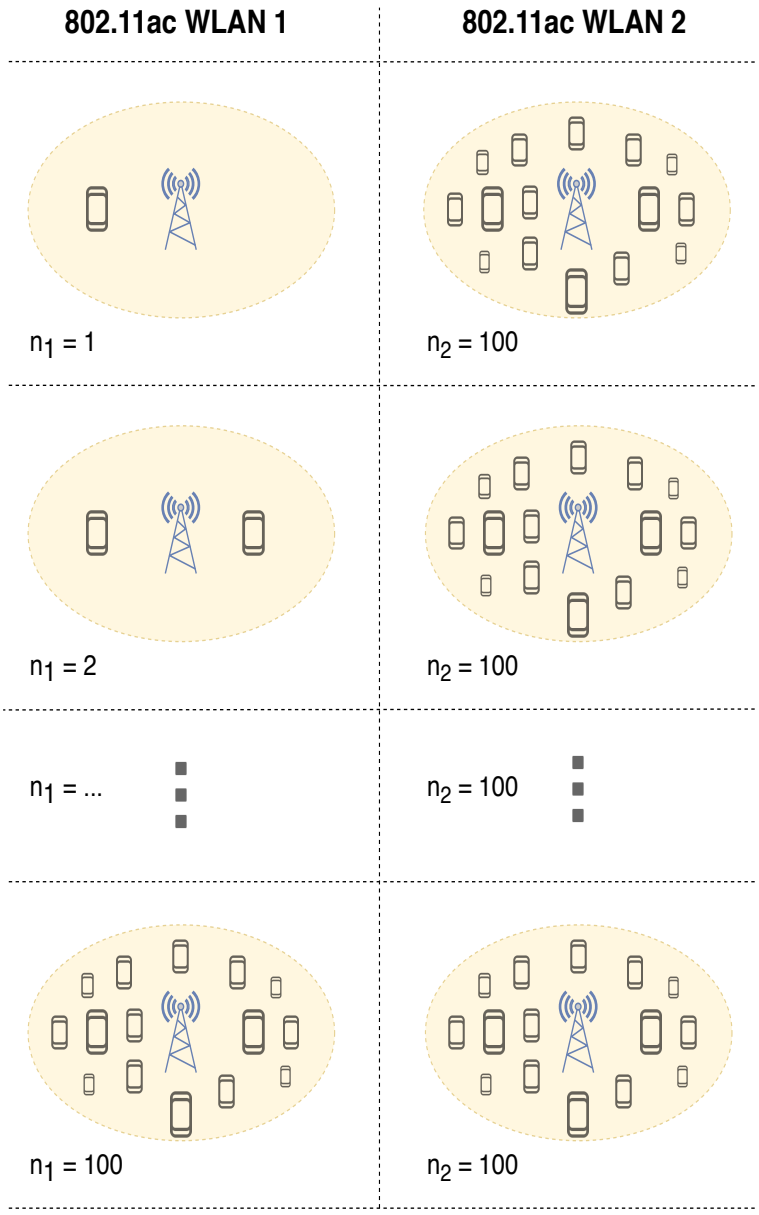

Figure 3.17: RR Test 1C 11ac-11ac Bond Unbalanced Load Scenario (Heavy Load)

3.4.2.2 Heterogeneous 11ac-11ah Bond

Next we present the channel load tests performed during the static round robin evaluation for the heterogeneous 11ac-11ah interface bond.

3.4.2.2.1 RR Test 2A: Balanced Load Scenario The first test was the balanced scenario illustrated in Figure 3.18 on the opposite page. At each iteration the applied load on the 11ac and 11ah links was increased simultaneously by incrementing the number of competing nodes in each individual 802.11 network from $n = 1$ to the maximum considered value of $n = 100$. Again the mean bond throughput and other metrics such as the overall bond delay are calculated for each link load combination. The applied loads used in the test were chosen to provide the best case scenario for the 11ac-11ah throughput and provide the basis for comparison against the homogeneous 11ac-11ac bond performance in Test 1A.

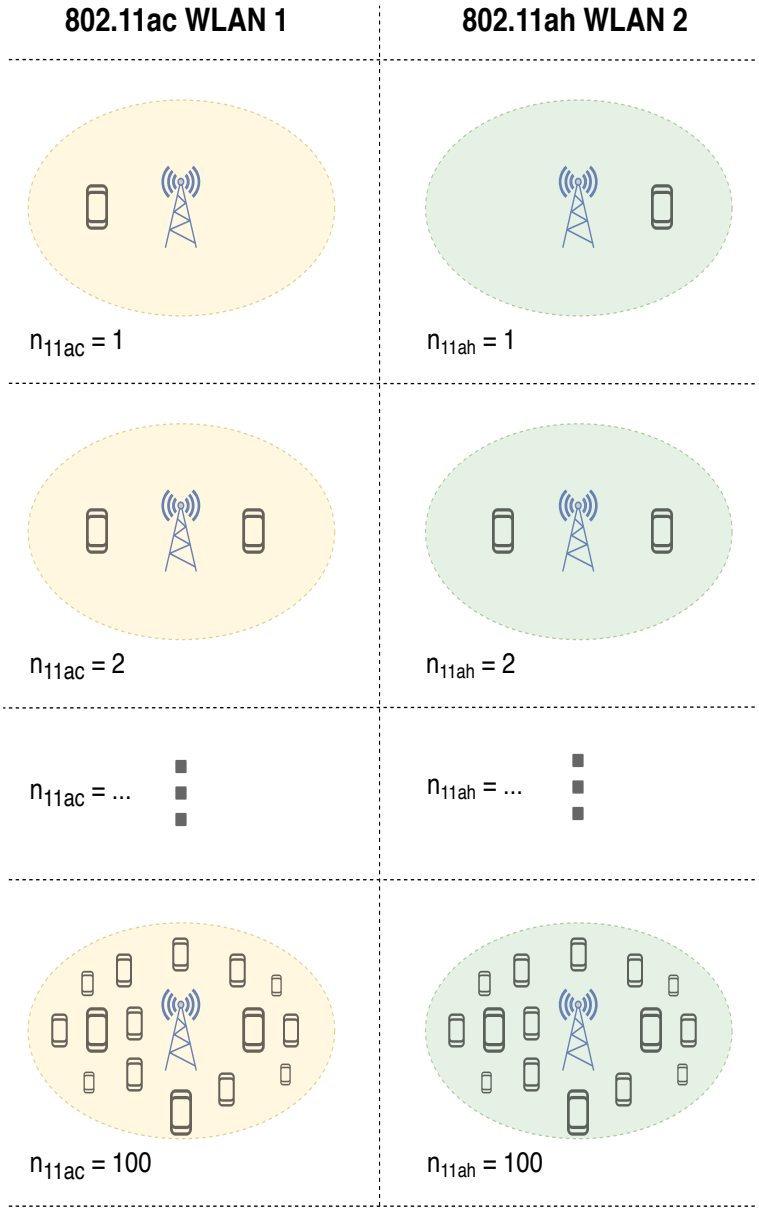

Figure 3.18: RR Test 2A 11ac-11ah Bond Balanced Load Scenario

3.4.2.2.2 RR Test 2B: Unbalanced Scenario (Low Load) In the first unbalanced link load scenario for the heterogeneous 11ac-11ah bond the applied load on the 11ah link was held constant at the low level of $n = 1$ nodes while the load on the 11ac link was varied by incrementing the number of competing stations from $n = 1$ to $n = 100$. Again the average bond throughput was calculated for each combination of applied link load across. The link loads used in the test were chosen to stress the aggregated throughput performance of the 11ac-11ah bond and provide the basis for comparison against the homogeneous 11ac-11ac bond in Test 1B.

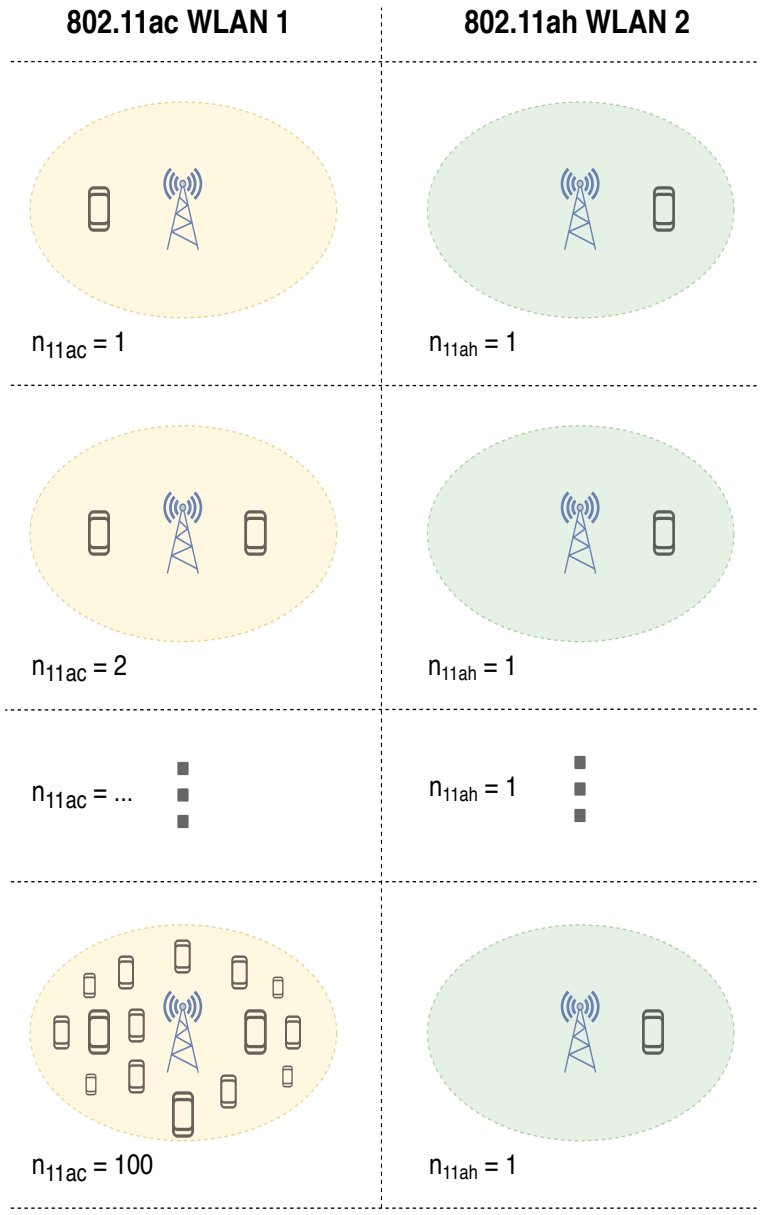

Figure 3.19: RR Test 2B 11ac-11ac Bond Unbalanced Scenario (Low Load)

3.4.2.2.3 RR Test 2C: Unbalanced Scenario (Moderate Load) The next test performed was the second unbalanced load scenario for the heterogeneous 11ac-11ah bond. As illustrated in Figure 3.20 at each iteration the applied load on the 11ah link was held constant at the moderate level of $n = 50$ nodes while the load on the 11ac link was incremented from $n = 1 to\ n = 100$. The link loads used in the test were chosen to stress the aggregated bond throughput performance using similar applied loads to the homogeneous 11ac-11ac bond in Test 1C.

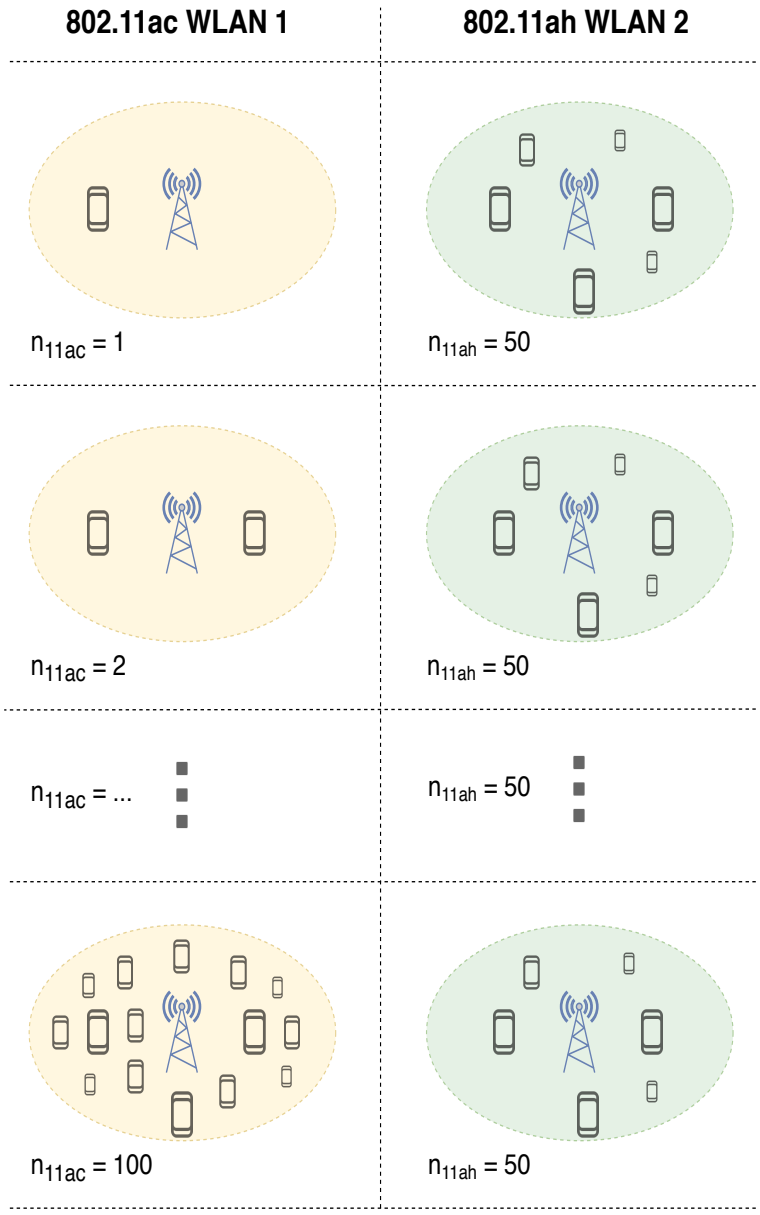

Figure 3.20: RR Test 2C 11ac-11ac Bond Unbalanced Scenario (Moderate Load)

3.4.2.2.4 RR Test 2D: Unbalanced Scenario (Heavy Load) The final test was the third unbalanced load scenario for the heterogeneous 11ac-11ah bond in which the applied load on the 11ah link was held constant at $n = 100$ competing nodes while the load on the 11ac link was incremented from $n = 1$ to $n = 100$ as illustrated in Figure 3.21 below. The link combinations used were chosen to stress the aggregated bond throughput performance and provide the basis for comparison against the homogeneous 11ac-11ac bond in Test 1D.

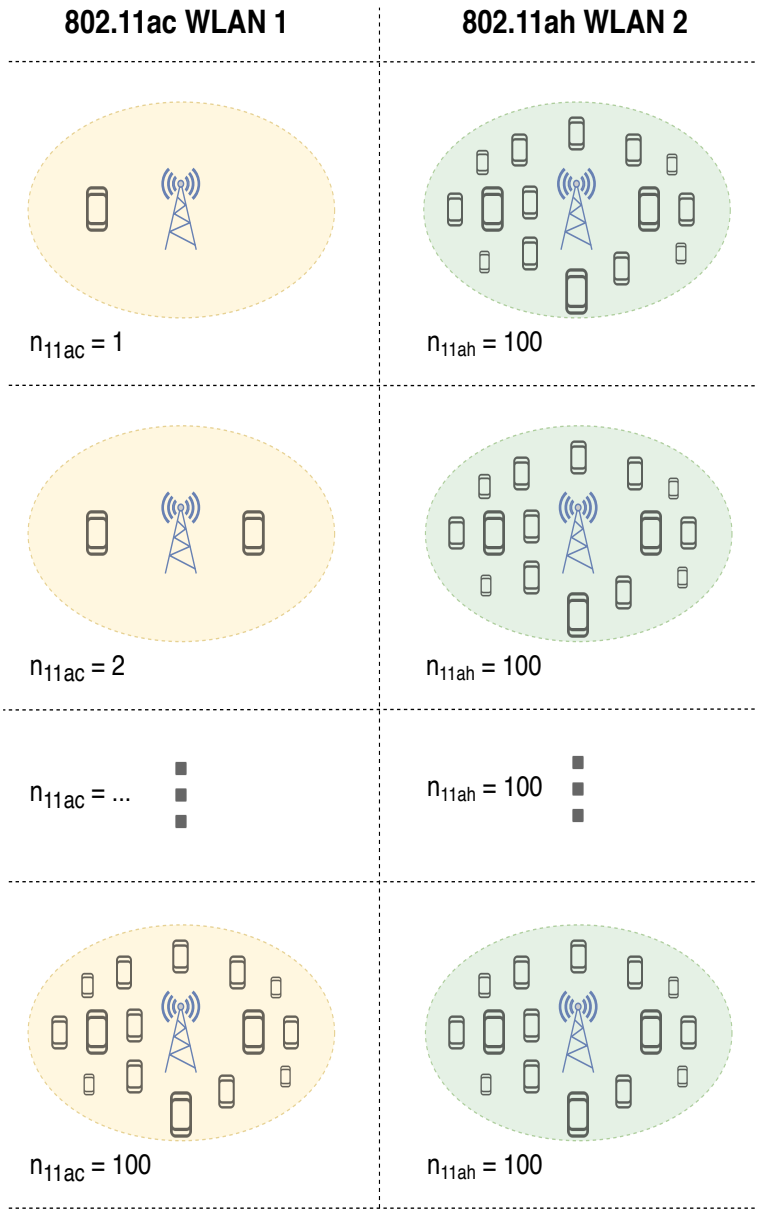

Figure 3.21: RR Test 2D 11ac-11ac Bond Unbalanced Scenario (Heavy Load)

3.5 Presentation of Results

This section presents the results from the evaluation of the static round robin 802.11 bond.

3.5.1 Homogeneous 11ac-11ac Bond

First we present the results for the homogeneous 11ac-11ac bond in Tests 1A-1D.

3.5.1.1 RR Test 1A: Results

The throughput results for static round robin under the benchmark case in Test 1A are given in Figure 3.22 on the opposite page. The round robin throughput is plotted as the dashed red line and the maximum theoretical bond throughput as the dotted and dashed black line. The overall mean achieved static round robin throughput was 2.4799 Mbps while the theoretical maximum mean bond throughput was 2.4803 Mbps. As seen from the graph the achieved throughput was equal to or very close to the maximum theoretical bond throughput for all applied load combinations considered in the evaluation. Both output metrics decreased exponentially as the applied load was raised on both 11ac links simultaneously. At the initial load of $\{n_1 = n_2 = 1\}$ both the round robin throughput and the maximum bond throughput were measured at approximately 57 Mbps, and when the load was incremented to $\{n_1 = n_2 = 2\}$ both then reduced significantly by around 50% to 28.22 Mbps. The exponential decrease then continued as the load was further raised with both metrics eventually reducing to a minimum of approximately 0.277 Mbps at the heaviest considered test load of $\{n_1 = n_2 = 100\}$.

As expected in light of the previous bond throughput results the overall mean link throughput difference of 1.8162 kbps seen in the test was negligible compared to the absolute throughputs of the individual 11ac links. For most applied loads the measured throughput difference was < 1 kbps with the exception of a brief spike to 7 kbps seen near the start of the experiment.

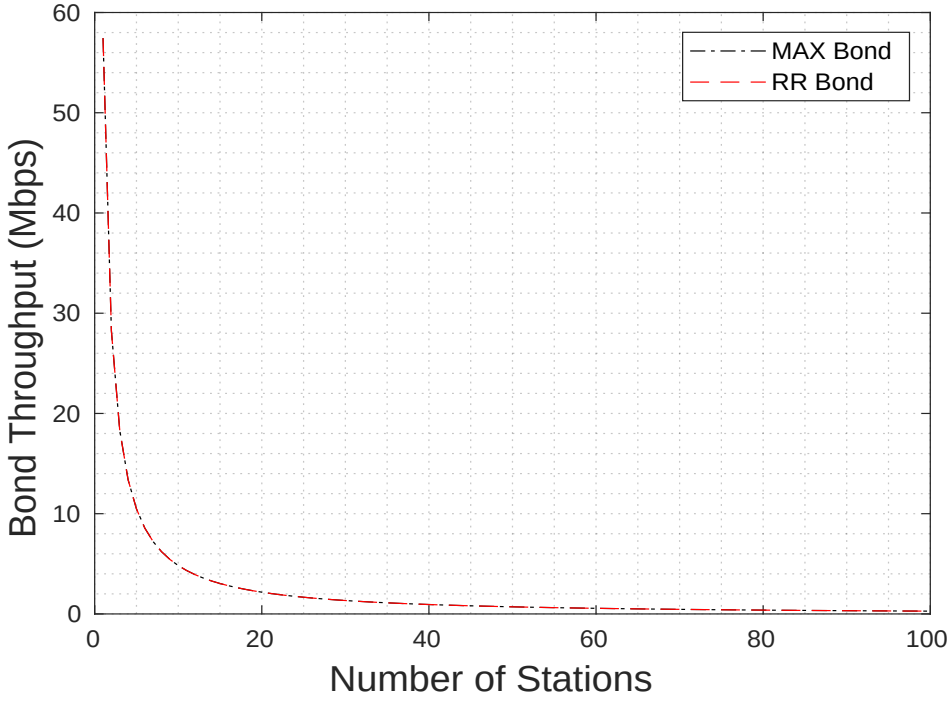

Figure 3.22: Test 1A Static Round Robin Bond Throughput

The link bandwidth utilisation results for the static round robin configuration in Test 1A are presented in Figure 3.23 below. As seen from the graph the achieved bandwidth utilisation was near to 100% for all applied loads from $\{n_1 = n_2 = 1\}$ to $\{n_1 = n_2 = 100\}$ with an overall mean value of 99.99%.

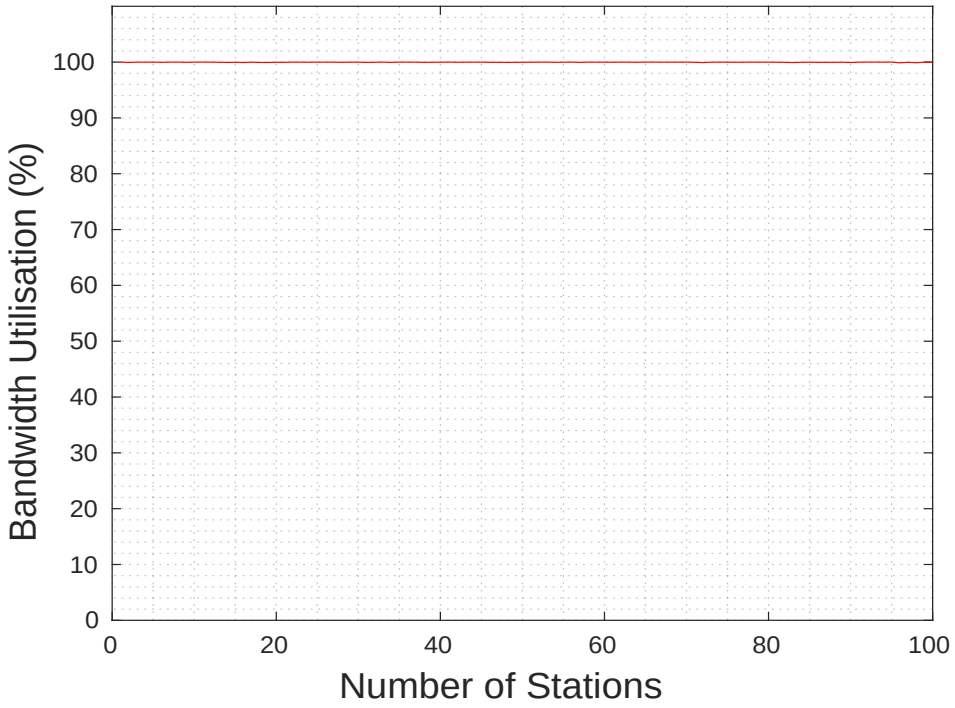

Figure 3.23: Test 1A Static Round Robin Link Bandwidth Utilisation

The bond delay results for the static round robin configuration in Test 1A are shown in Figure 3.24below. The overall bond delay is given as the dotted black line and the individual Link 1 and Link 2 11ac delays as the solid red and dashed blue lines respectively. The mean bond delay for all applied load combinations used was 50263 μs. Notably the Link 1 and 2 delays were equal throughout and a non-linear increasing curve was observed for both outputs as the applied load was raised. As seen from an initial value of 0.00086 μs at $\{n_1 = n_2 = 1\}$ the bond delay increased to a maximum of 117500 μs at the heaviest considered load of $\{n_1 = n_2 = 100\}$. The overall mean link delay difference for all applied load combinations used was 95.37 μs.

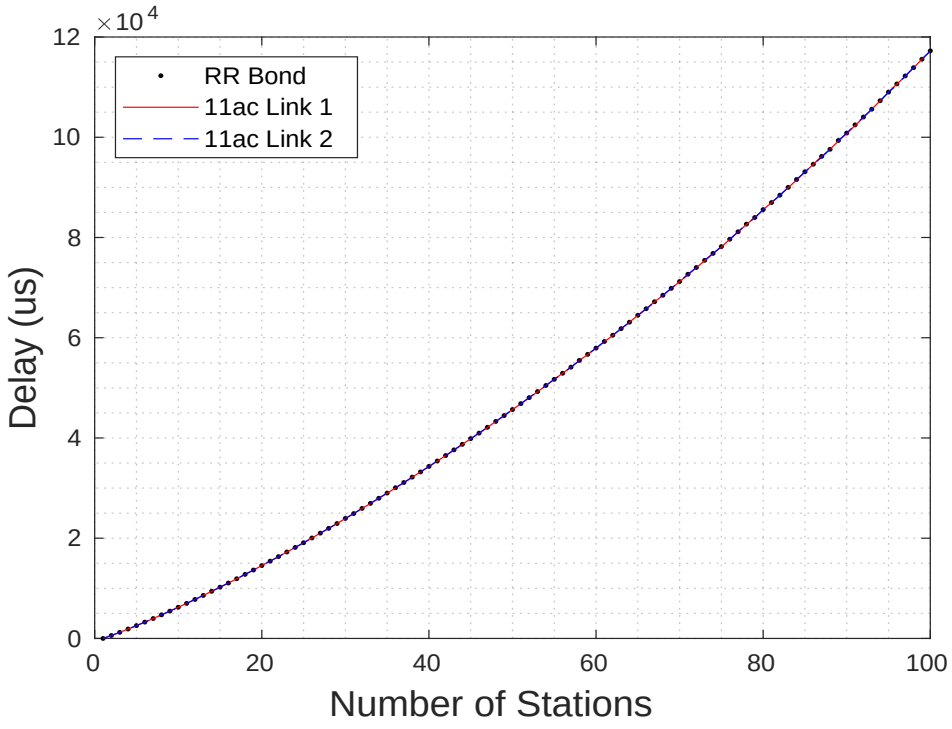

Figure 3.24: Test 1A Static Round Robin Bond Delay

3.5.1.2 RR Test 1A: Discussion

The main observation from the benchmark case in Test 1A was that the static round robin bond throughput was approximately equal to the maximum theoretical bond throughput for all applied loads providing a mean link bandwidth utilisation of 99.99% which was the highest of all four tests performed for the homogeneous 11ac-11ac bond.

As discussed previously to provide a performance benchmark in Test 1A a balanced applied load was used to create the ideal conditions for the homogeneous 11ac-11ac bond throughput. The optimal throughput performance was confirmed by the results in Figure 3.22 above. Furthermore the mean achieved round robin throughput and the mean theoretical maximum bond throughput were both approximately equal at 2.48 Mbps.

There are three conditions that contributed to the optimal bond throughput performance seen: (1) both 11ac links have the same raw PHY data rate of 65 Mbps, (2) both 11ac slave links also have the same DCF parameter values, for example the same minimum contention window size and slot duration, and (3) the number of competing stations in each 802.11ac WLAN was incremented simultaneously such that the applied load was balanced across the bond during every iteration of the testing. When all three conditions are satisfied they lead to an identical throughput and delay response for each individual 802.11 device such that on average both links will achieve a similar throughput and delay at similar levels of applied load. Therefore both slave links finish their transmissions on average at approximately the same time and neither link is forced to wait for the other to complete before the overall bonded transmission is considered complete. The bond throughput slow-down effect due to the in-order delivery requirement is minimised and the bond throughput performance is near optimal and approximately equal to the sum of the throughputs of the individual links as seen in Figure 3.22 previously. The optimal performance was also reflected in Figure 3.23 where the bond throughput efficiency was near or equal to 100% for all applied loads considered. The identical throughput response was confirmed by the link throughput difference results for Test 1A in Figure **??** with a overall mean of 0.0018 kbps. Similarly the identical delay response was also confirmed in Figure **??** above which showed that the resulting delay difference between the two 11ac slave links was small compared to the absolute individual link delays with a mean value of 95.37 μs. The bond delay results in Figure 3.24 above show a similar picture of the bond performance. The delay increased non-linearly as the applied load was raised following both the 11ac Link 1 and Link 2 delays which were approximately equal for all load combinations used. Note that although the mean bond throughput of 2.48 Mbps may seem low but is in fact caused by the decreasing throughputs of the slave links due to increased resource sharing as opposed to a bond throughput slow-down effect as seen in later tests.

When any of the above three conditions are not satisfied then a bond throughput slow-down effect is observed due to the commonly found in-order delivery requirement. To evaluate the impact of the slow-down when requirement (1) above is not satisfied Test 1A was repeated and the raw PHY data rate of the first 11ac link was increased by approximately doubling to 128 Mbps, 256 Mbps, 1024 Mbps, and 2048 Mbps. Most notably the results in Figure 3.25 reveal that the aggregate bond throughput is *always* limited by the throughput of the slower 65 Mbps device regardless of the capacity of the faster link. For example even the contribution of the high speed 2048 Mbps link was reduced to that of the slower 65 Mbps device as a consequence of the throughput slow-down effect due to the in-order delivery requirement.

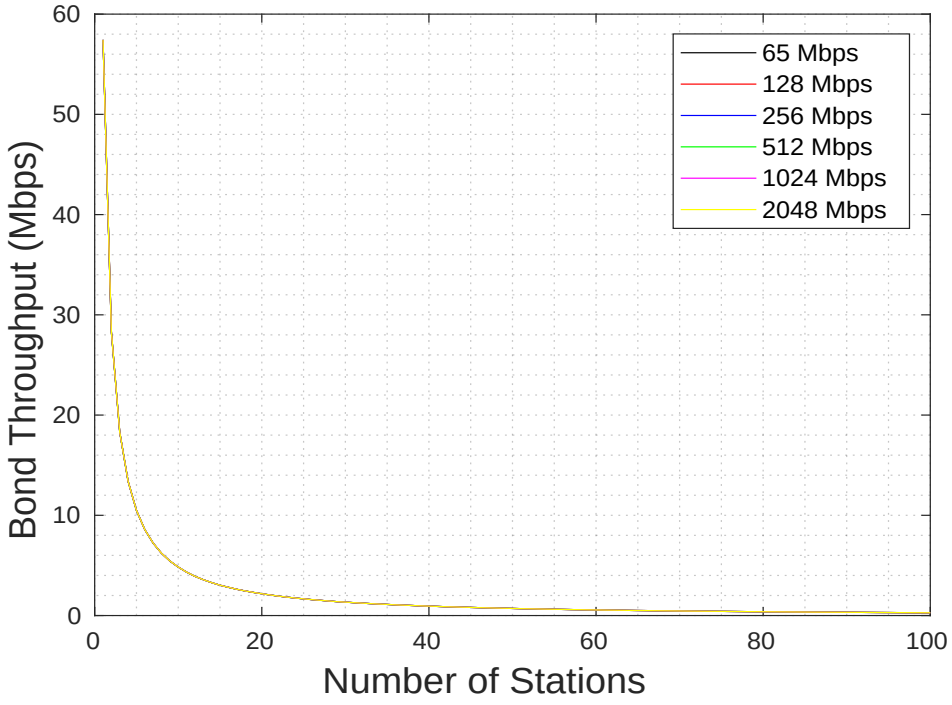

Figure 3.25: Test 1A Static Round Robin Throughput (Heterogeneous Data Rates)

Similarly Figure 3.26 below shows how a change in the Link 1 data rate link impacted the link throughput difference of the bond. All device pairings followed a similar exponential decreasing curve as the applied load was increased and only their absolute values differed according to the size of the throughput difference. For example as expected the device pairing with the highest overall link throughput difference was the heavily unbalanced 2048Mbps-65Mbps interface bond.

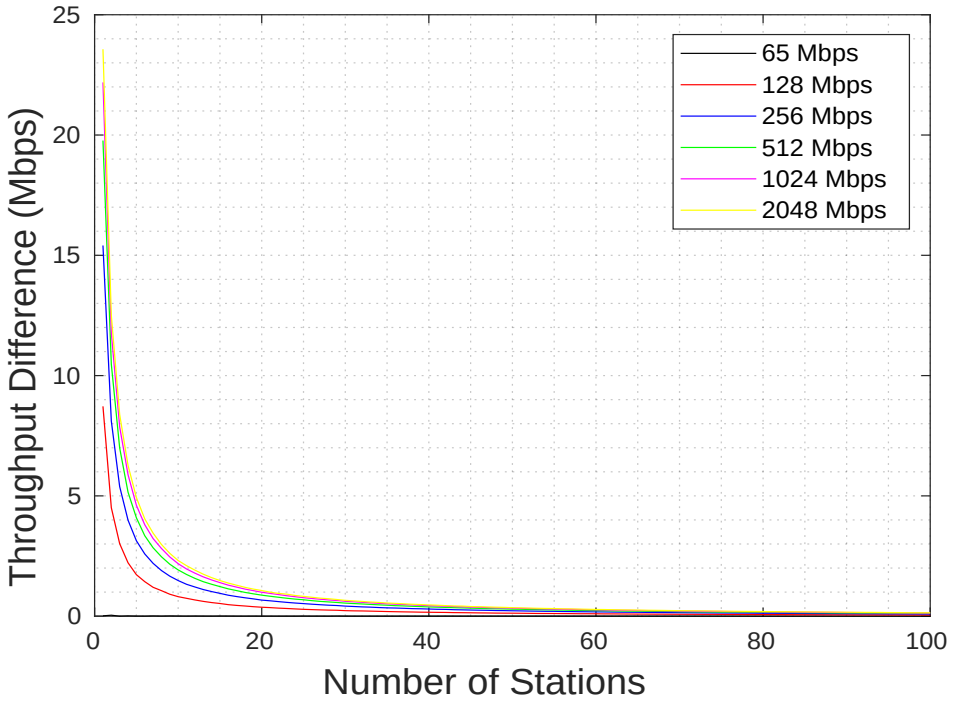

Figure 3.26: Test 1A Static Round Robin Throughput Difference (Heterogeneous Data Rates)

Of greater interest is the link bandwidth utilisation in Figure 3.27 below which illustrates the impact for the same PHY data rate change. An almost flat efficiency output was seen for all data rate pairings as the applied bond load was increased. As expected the highest utilisation was seen for the homogeneous 65Mbps-65Mbps bond where the resulting efficiency was equal to 100% or near for all applied loads considered. As expected the next highest was the 128Mbps-65Mbps pairing with a mean of 85% followed by the 256Mbps-65Mbps, 512Mbps-65Mbps, and 1024Mbps-65Mbps bonds with means of 76%, 71%, and 68% respectively. The lowest rating was 66% for the 2048Mbps-65Mbps pairing such that roughly one third of the available link bandwidth was wasted due to the inherent slow-down effect.

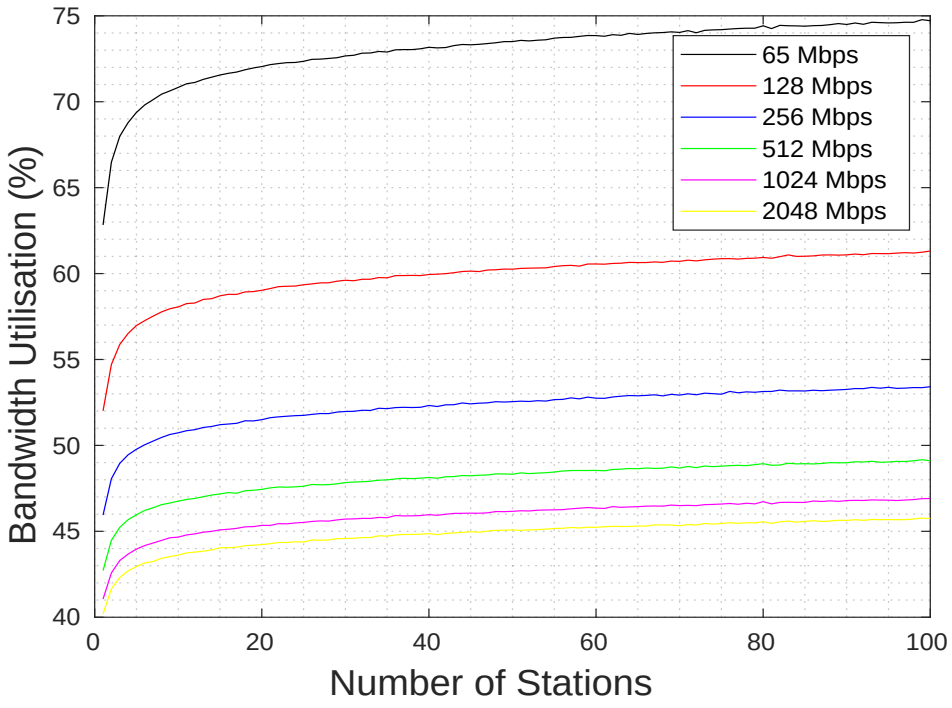

Figure 3.27: Test 1A Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.5.1.3 RR Test 1B: Results

The bond throughput results for the static round robin configuration and unbalanced load scenario in Test 1B are shown in Figure 3.28 below. The round robin bond throughput is plotted as the solid red line and the maximum theoretical bond throughput as the solid black line. The overall mean bond throughput for all applied loads considered was 19.76 Mbps while the theoretical maximum mean bond throughput was 29.971 Mbps. Both metrics displayed a non-linear decreasing curve with a clearly visible difference in the response as the applied bond load was raised. At $\{n_1 = 1, n_2 = 1\}$ the achieved round robin throughput was equal to the theoretical maximum bond throughput of 57.45 Mbps but by $\{n_1 = 2, n_2 = 1\}$ the two outputs had already begun to diverge. As the applied load was increased towards $\{n_1 = 100, n_2 = 1\}$ the achieved static round robin throughput reduced non-linearly to 13.39 Mbps while the maximum theoretical bond throughput was decreased to 28.84 Mbps.

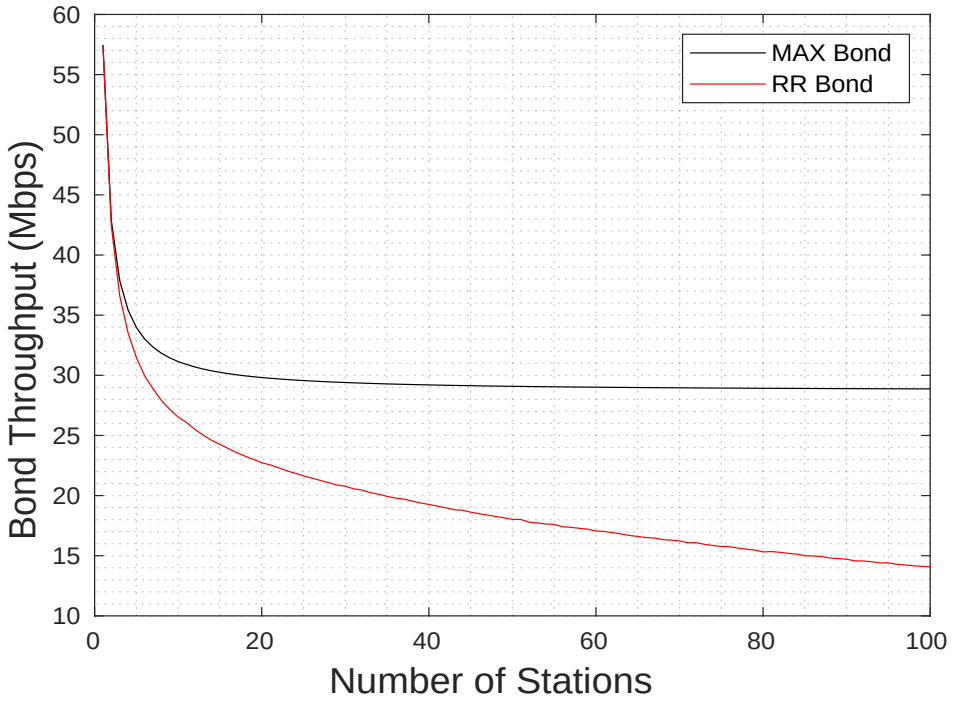

Figure 3.28: Test 1B Static Round Robin Bond Throughput

The link throughput difference results for static round robin in Test 1B are shown in Figure 3.29 below. The mean link throughput difference of the dual-11ac interface bond for all applied load combinations used was 24.489 Mbps. At the lightest test load of $\{n_1 = 1, n_2 = 1\}$ the link throughput difference was negligible at 1.596 kbps but as the load incremented to $\{n_1 = 2, n_2 = 1\}$ the simulation output increased to 14.61 Mbps which was approximately 50% of the maximum value . As the applied load was further increased towards $\{n_1 = 100, n_2 = 1\}$ the link throughput difference increased non-linearly to a maximum value of 28.59 Mbps.

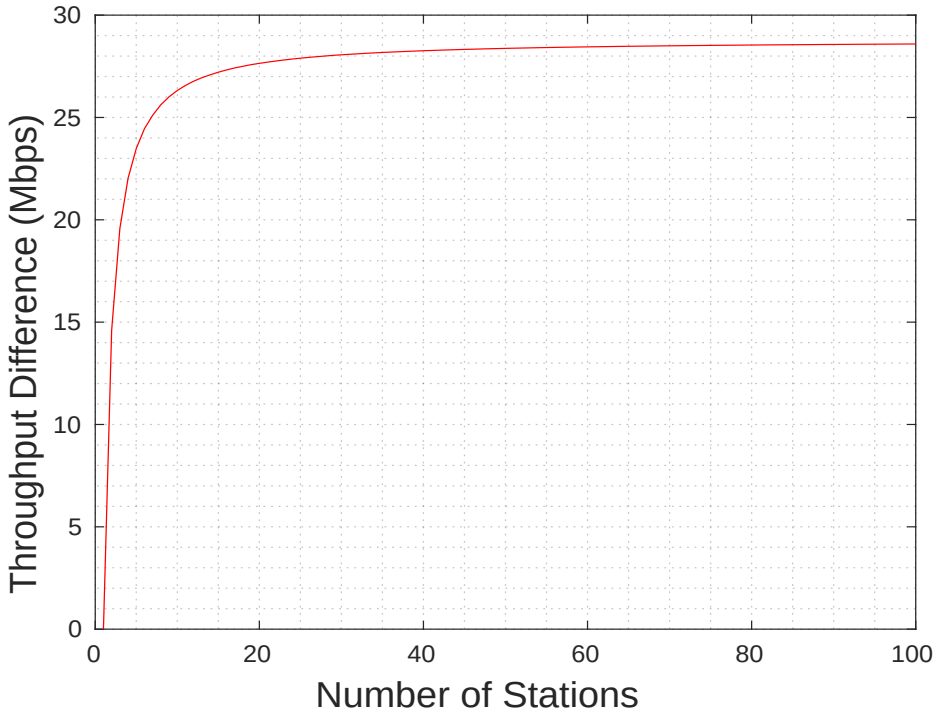

Figure 3.29: Test 1B Static Round Robin Link Throughput Difference

The link bandwidth utilisation results for static round robin in Test 1B are given in Figure 3.30. The overall mean utilisation for all applied load combinations used was 65.03%. As seen below a non-linear decreasing curve was observed in the simulation output as the applied load was increased. At the lightest test load of $\{n_1 = 1, n_2 = 1\}$ the bandwidth utilisation was at a maximum value of 99.91%. Then the applied load was increased towards $\{n_1 = 100, n_2 = 1\}$ the achieved utilisation reduced non-linearly to a minimum value of 48.7%.

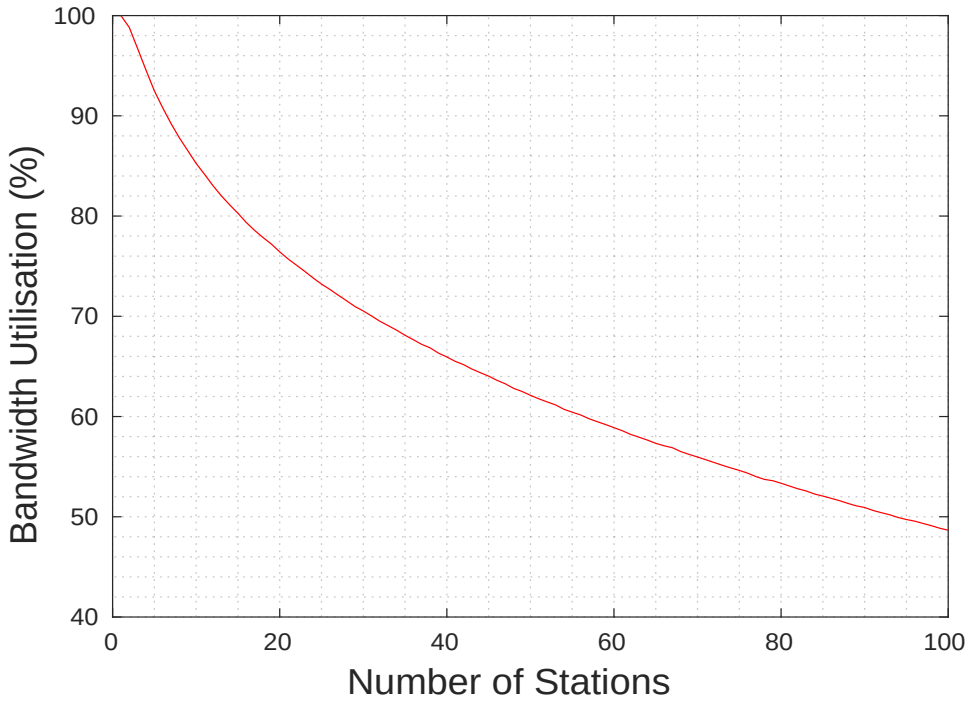

Figure 3.30: Test 1B Static Round Robin Link Bandwidth Utilisation

The bond delay results for the static bond configuration in Test 1B are shown in Figure 3.31 below. The overall bond delay is plotted as the dotted black line and the individual Link 1 and Link 2 11ac delays as the dash and dotted red line and dotted blue line respectively. The mean bond delay for all applied load combinations considered was 50226.553 μs. For the entire range of applied loads considered the bond delay followed the Link 1 which increased non-linearly from a minimum of 0.00086 μs at $\{n_1 = n_2 = 1\}$ to a maximum value of 117500 μs at the heaviest prescribed load of $\{n_1 = n_2 = 100\}$.

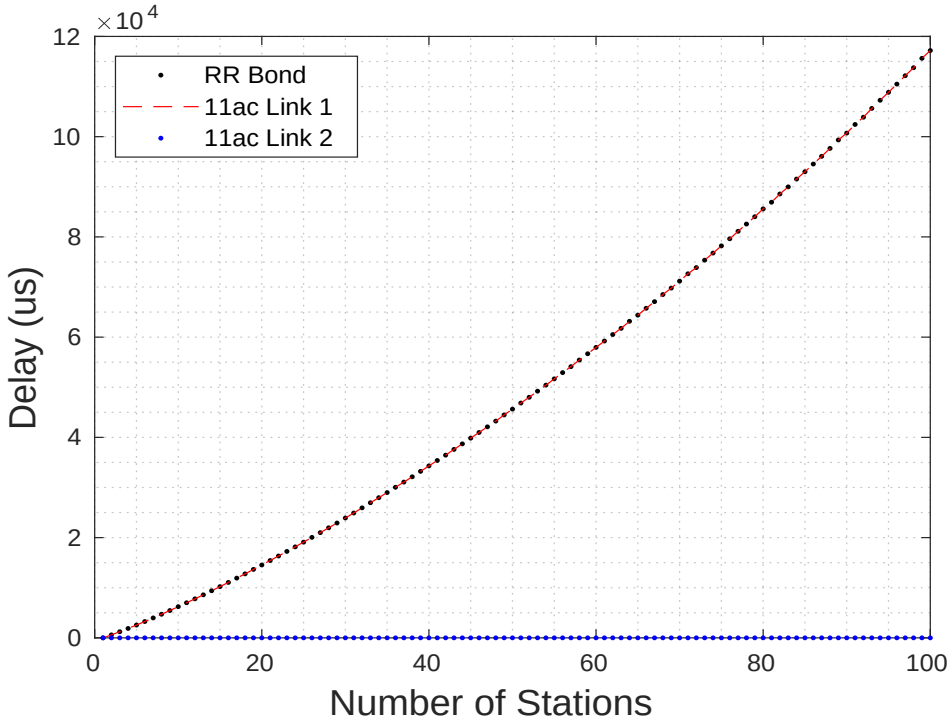

Figure 3.31: Test 1B Static Round Robin Bond Delay

The link delay difference results for the static round robin bond in Test 1B are presented in Figure 3.32. The overall mean delay difference between for all applied load combinations used was 50226.5514 μs. As seen from the graph below a non-linear increasing curve was observed in the simulation output. At the initial load of $\{n_1 = 1, n_2 = 1\}$ delay difference between links was negligible at 0.00072 μs. Then as the applied load was gradually raised towards $\{n_1 = 100, n_2 = 1\}$ the delay difference increased non-linearly to a maximum value of 117200 μs.

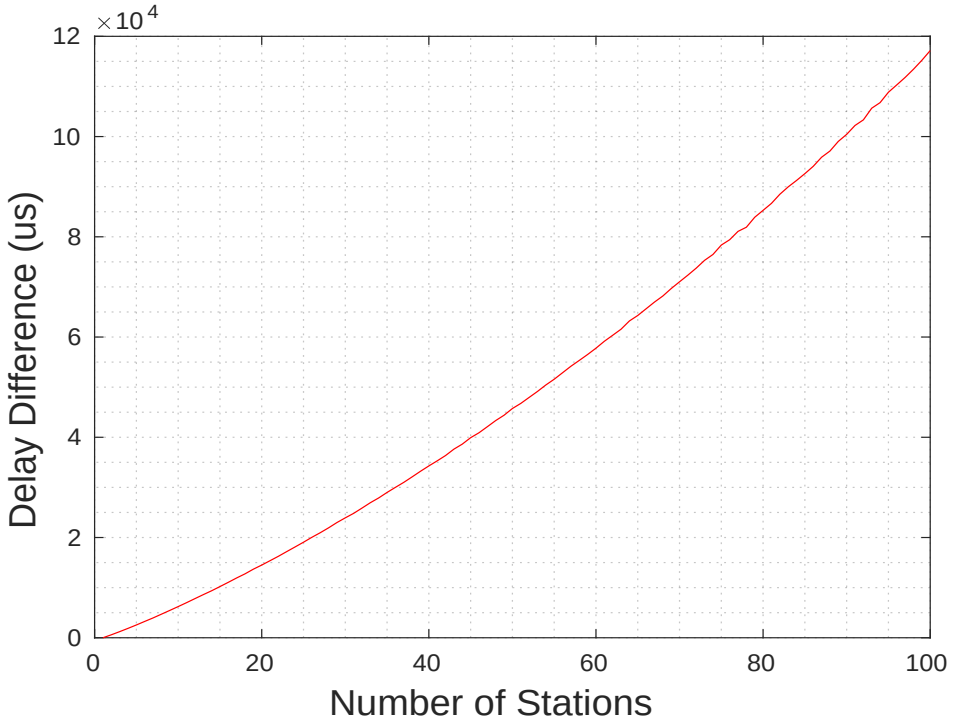

Figure 3.32: Test 1B Static Round Robin Link Delay Difference

3.5.1.4 RR Test 1B: Discussion

The main observation from Test 1B was that the mean static round robin throughput of 27.49 Mbps was 8.28% lower than the theoretical maximum bond throughput due to the impact of the throughput slow-down effect and in-order delivery assumption. The achieved bond throughput was lower than the maximum theoretical bond throughput for all applied loads considered with the exception of $\{n_1 = 1, n_2 = 1\}$ where both were approximately equal. Furthermore the mean link bandwidth utilisation of 65.03% was the lowest of tests 1A-1D such that on average over one third of the available link bandwidth was wasted unnecessarily.

In Test 1B conditions (1) and (2) described in Section 3.5.1.2 above were met but point (3) was not satisfied as the applied load on the second 11ac link was kept constant at n = 1 while the number of nodes on the first 11ac link was varied from 1 to 100. Note that at $\{n_1 = 1, n_2 = 1\}$ both bond links had the same applied load and therefore the resulting throughput was equal to the maximum theoretical bond throughput, which is expected following the results of Test 1A. This was due to the equivalent performance of each 11ac link as seen in Figure 3.29 and Figure 3.32 where the link throughput difference and link delay difference were both measured at approximately zero presenting a similar situation to that seen throughout in the benchmark case in Test 1A. The optimal performance at $\{n_1 = 1, n_2 = 1\}$ was also reflected in the link bandwidth utilisation results in Figure 3.30 where it was measured at a maximum test value of 99.98%. However as the applied load was incremented to $\{n_1 = 2, n_2 = 1\}$ as seen in Figure 3.29 the slow-down effect became noticeable for the first time as the throughput difference between links increased and the bandwidth utilisation in Figure 3.30 was consequently reduced. The performance disparity between the bond links became larger as the applied load was further increased thereby increasing the impact of the slow-down effect. At the largest test load of $\{n_1 = 100, n_2 = 1\}$ the link throughput difference reached a maximum test value of 28.59 Mbps and as a consequence the achieved bond throughput efficiency fell to a minimum of 40%.

Another key finding from the unbalanced applied load scenario used in Test 1B is illustrated in Figure 3.33 below. Notably for applied bond loads where $\{n_1 >= 7, n_2 = 1\}$ the achieved static round robin bond throughput was reduced below that of the single fastest available link, which was in this case 11ac Link 2 with the static applied load of n = 1.

This particular finding highlights exactly how bad things can get for the achieved bond throughput when a dynamic and heterogeneous load is applied to a statically configured round robin 802.11 interface bond. For the applied loads above the impact of the throughput slow-down effect is so severe that the bond throughput is reduced below what would have been possible using just the single faster 11ac Link 2, therefore rendering the use of the bonding technique counter productive to the general goal of increasing the overall data transfer rate.

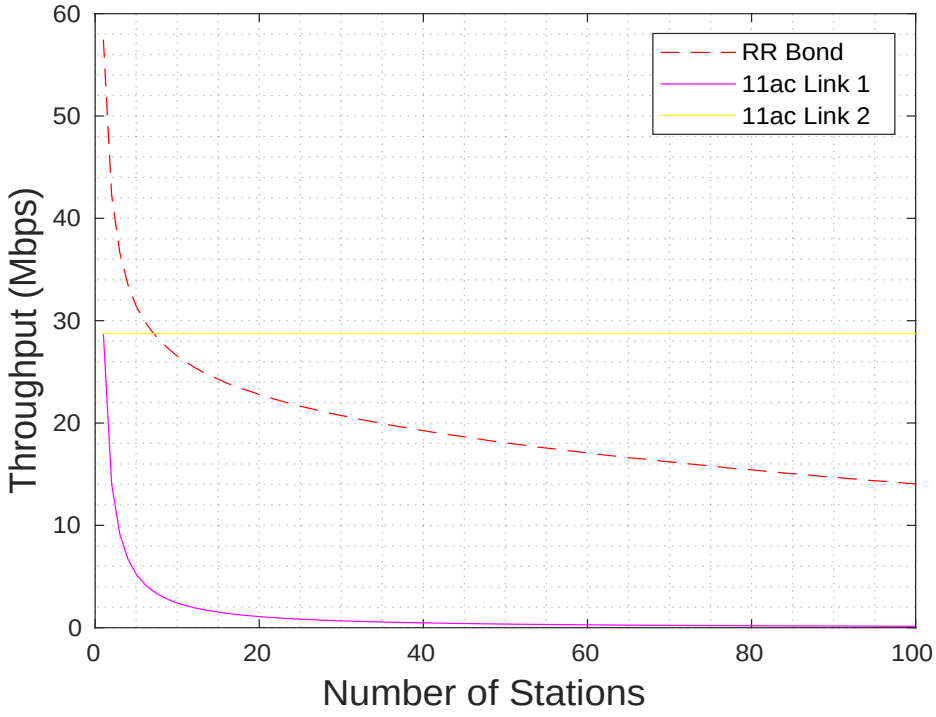

Figure 3.33: Test 1B Static Round Robin Link Throughput Comparison

Again to demonstrate what happens when condition (1) of Section 3.5.1.2 above is not met Test 1B was repeated and the raw PHY data rate of the first 11ac link was increased by doubling to 128 Mbps, 256 Mbps, 1024 Mbps, and 2048 Mbps. Similar to Test 1A the results for the current test in Figure 3.34 show that the overall aggregate bond throughput is *always* limited by the throughput of the slower 65 Mbps device regardless of the capacity of the faster link. Even the throughput contribution of the high speed 2048 Mbps interface was reduced to that of the slower 65 Mbps link as a consequence of the throughput slow-down effect and in-order delivery requirement.

The 65Mbps-65Mbps bond results are as discussed above but for the other device pairings the simulation output displayed a different behaviour. Take for example the 65Mbps-128Mbps pairing where the bond throughput output is different at applied loads of $\{n_1 <= 22,\ n_2 = 1\}$ compared to $\{n_1 > 22,\ n_2 = 1\}$. At $\{n_1 = 1,\ n_2 = 1\}$ the bond throughput was 57.46 Mbps and the speed of the bond was limited by the throughput of the slower second 11ac link. Then as the load on the 128 Mbps link increased its achievable throughput was reduced until eventually becoming lower than the 65 Mbps link throughput at $\{n_1 > 22,\ n_2 = 1\}$ such that the overall bond speed was instead then limited by the throughput of the first 11ac link. A similar behaviour is seen for the 65Mbps-256Mbps, 65Mbps-512Mbps, 65Mbps-1024Mbps, and 65Mbps-2048 Mbps device pairings for which the same event occurs at applied loads of $\{n_1 = 52,\ n_2 = 1\}$, $\{n_1 = 74,\ n_2 = 1\}$, $\{n_1 = 88,\ n_2 = 1\}$ and $\{n_1 = 93,\ n_2 = 1\}$ respectively.

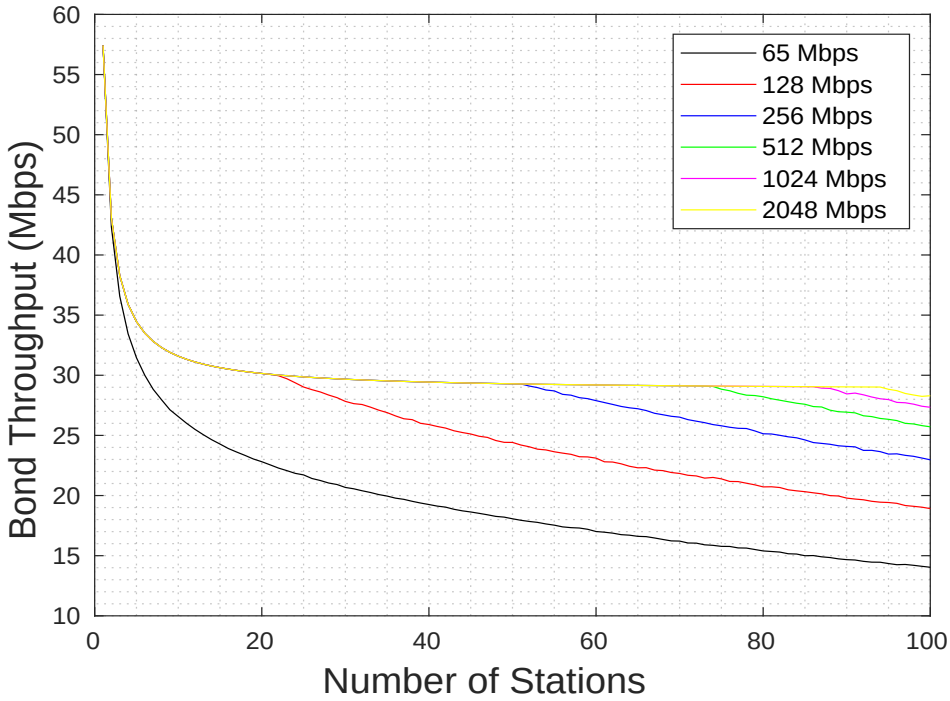

Figure 3.34: Test 1B Static Round Robin Throughput (Heterogeneous Link Data Rates)

The same behaviour was reflected in the link bandwidth utilisation results shown in Figure 3.30 below. Note how the utilisation of the 65Mbps-128Mbps bond increased non-linearly over $\{n_1 <= 22, n_2 = 1\}$ until it reached a maximum of 100% at n = 22 while for $\{n_1 > 22, n_2 = 1\}$ it was gradually reduced. The same event is seen for the 65Mbps-256Mbps, 65Mbps-512Mbps, 65Mbps-1024Mbps, and 65Mbps-2048 Mbps device pairings at applied bond loads of $\{n_1 = 52, n_2 = 1\}$, $\{n_1 = 74, n_2 = 1\}$, $\{n_1 = 88, n_2 = 1\}$ and $\{n_1 = 93, n_2 = 1\}$ respectively.

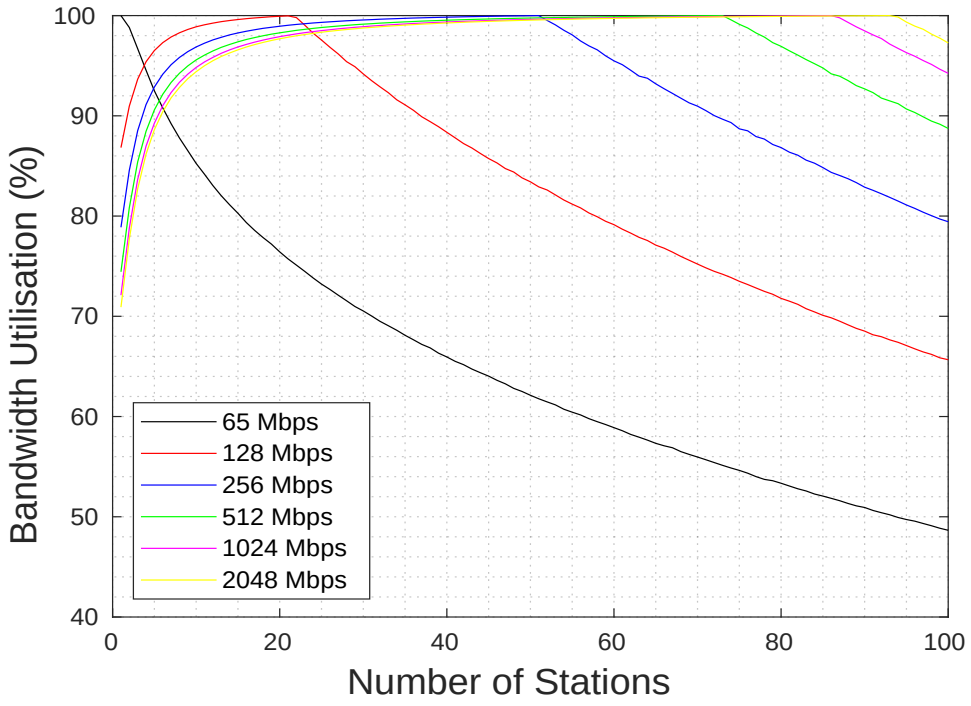

Figure 3.35: Test 1B Static Round Robin Bandwidth Utilisation (Heterogeneous Link Data Rates)

3.5.1.5 RR Test 1C: Results

The bond throughput results for the static round robin configuration and unbalanced load scenario in Test 1C are given in Figure 3.36 below. The overall mean achieved bond throughput for all applied load combinations considered was 1.2422 Mbps. An exponentially decreasing curve was seen for both metrics as the applied load was raised with an apparently smaller difference compared to the previous test. At the first load of $\{n_1 = 1, n_2 = 50\}$ the achieved bond throughput of 18.07 Mbps was already 46.7% lower than the theoretical maximum bond throughput of 29.08 Mbps. As the load increased to $\{n_1 = 2, n_2 = 50\}$ the achieved throughput reduced to 9.213 Mbps and the maximum bond throughput to 14.47 Mbps. Close inspection of the graph below shows that as the load was further increased the difference between the two metrics gradually reduced until eventually at a load of $\{n_1 = 50, n_2 = 50\}$ the round robin throughput and the maximum bond throughput were both approximately equal at 0.7037 Mbps. Then as the applied load was raised in excess of $\{n_1 = 50, n_2 = 50\}$ the two outputs began to diverge until finally at the heaviest considered load of $\{n_1 = 100, n_2 = 50\}$ the bond throughput was reduced to 0.4161 Mbps and the maximum bond throughput to 0.4918 Mbps such that the achieved bond throughput was 16.67% lower than the theoretical maximum.

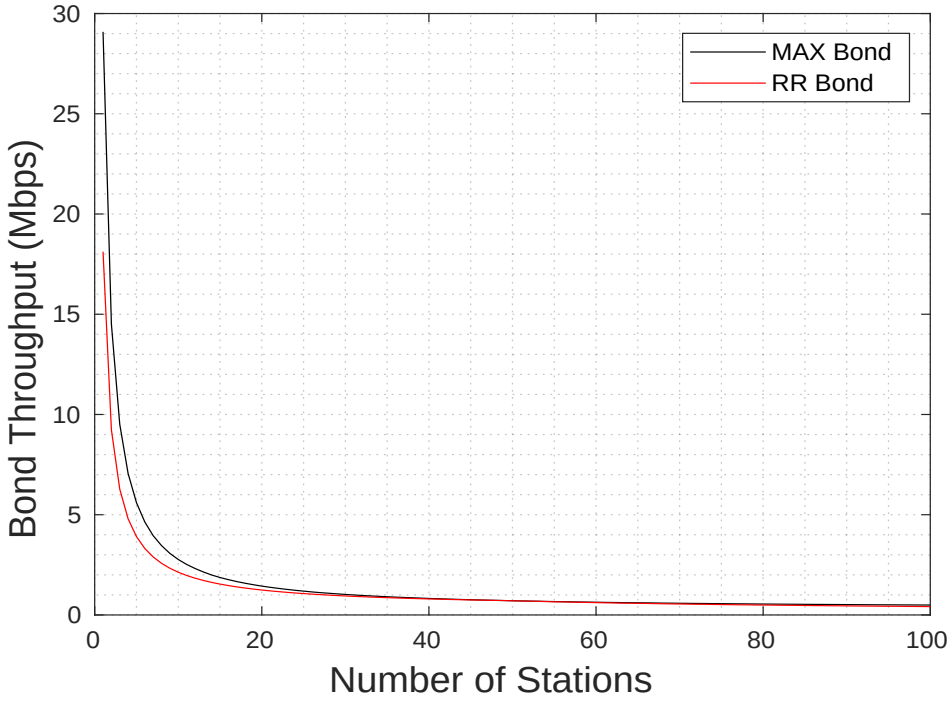

Figure 3.36: Test 1C: Static Round Robin Bond Throughput

The link throughput difference results for static round robin in Test 1C are shown in Figure 3.37 below. The overall mean link throughput difference for all applied load combinations considered was 1.0223 Mbps. During the test a non-linear decreasing output was observed as the applied bond load was raised. At $\{n_1 = 1, n_2 = 50\}$ the throughput difference was measured at a maximum value of 28.39 Mbps and then as the applied load was increased the link throughput difference gradually reduced exponentially to a minimum value of 0.21 Mbps at the heaviest considered load of $\{n_1 = 100, n_2 = 50\}$.

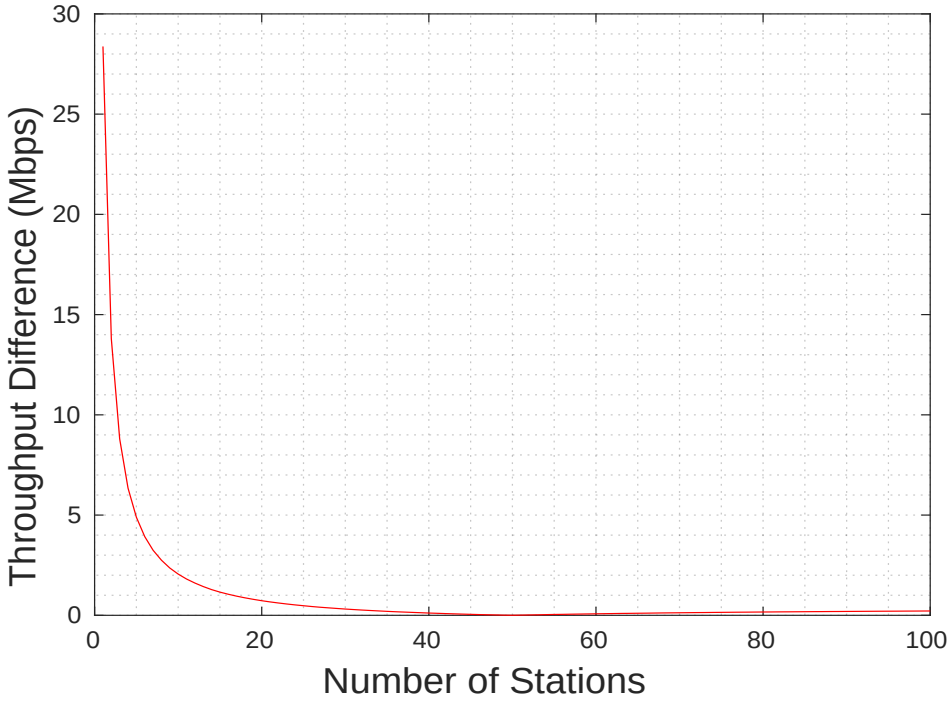

Figure 3.37: Test 1C: Static Round Robin Link Throughput Difference

The link bandwidth utilisation results for the static round robin bond configuration in Test 1C are shown in Figure 3.38 below. The overall mean utilisation for all applied loads considered was 91.942%. Furthermore there was a clear difference in behaviour compared to the previous test. At $\{n_1 = 1, n_2 = 50\}$ the utilisation was measured at a minimum of 62% and as the applied load increased the output rose non-linearly to a maximum of 99.74% at exactly $\{n_1 = 50, n_2 = 50\}$. Then as the applied load was further increased the achieved utilisation reduced gradually to a final value of 84.36% at the heaviest considered load of $\{n_1 = 100, n_2 = 50\}$.

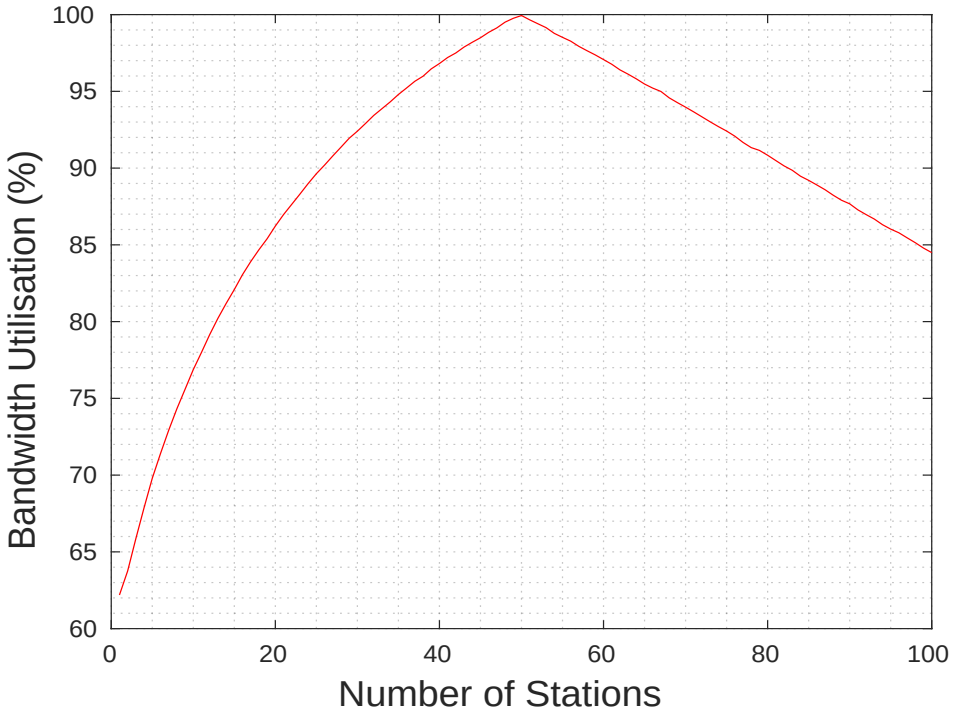

Figure 3.38: Test 1C: Static Round Robin Link Bandwidth Utilisation

The static round robin delay results for the homogeneous 11ac-11ac bond in Test 1C are presented in Figure 3.39 below. The overall bond delay is plotted as the dotted black line and the individual Link 1 and Link 2 delays as the dashed red and blue lines respectively. The overall mean bond delay for all applied load combinations considered was 62596.3 μs. For applied loads where $\{n_1 < 50,\ n_2 = 50\}$ the bond delay followed the Link 2 delay and remained constant at approximately 45420 μs. However for $\{n_1 > 50,\ n_2 = 50\}$ the bond delay followed Link 1 and increased non-linearly from 45740 μs to a maximum value of 117000 μs at the heaviest considered load of $\{n_1 = 100,\ n_2 = 50\}$.

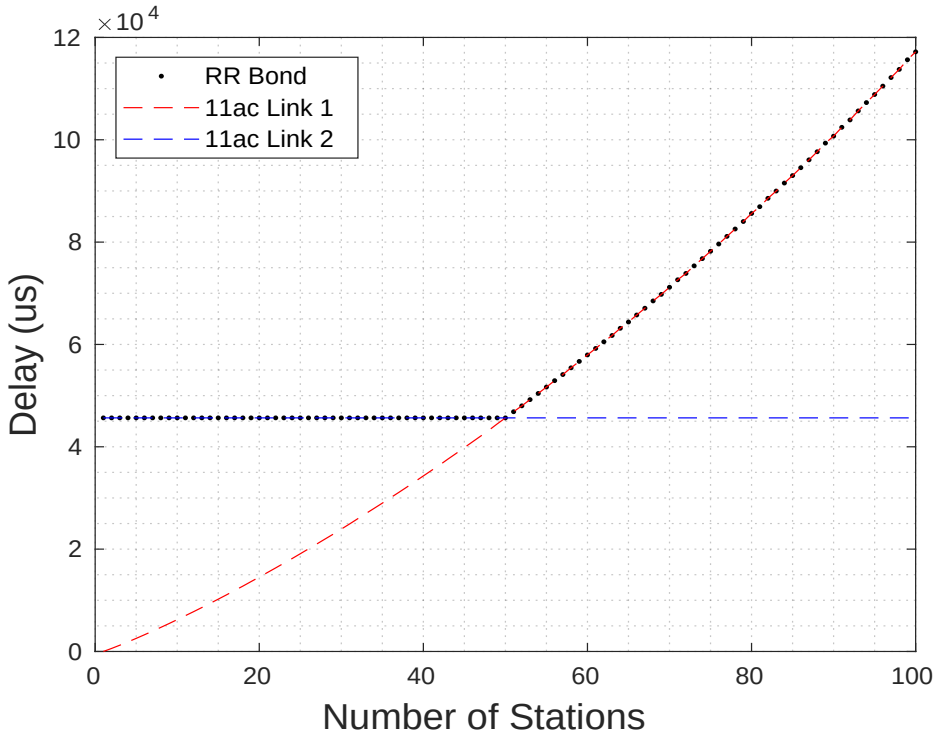

Figure 3.39: Test 1C: Static Round Robin Bond Delay

The link delay difference results for the static round robin bond configuration in Test 1C are presented in Figure 3.40 below. The overall mean link delay difference for all applied loads used in the test was 22646.8 μs. At the initial load of $\{n_1 = 1, n_2 = 50\}$ the delay difference was approximately 45540 μs. Then as the load was increased the output reduced non-linearly to a minimum value of 326 μs at exactly $\{n_1 = 50, n_2 = 50\}$. Then as the load was further raised towards $\{n_1 = 100, n_2 = 50\}$ the link delay difference increased non-linearly to a maximum of 71560 μs.

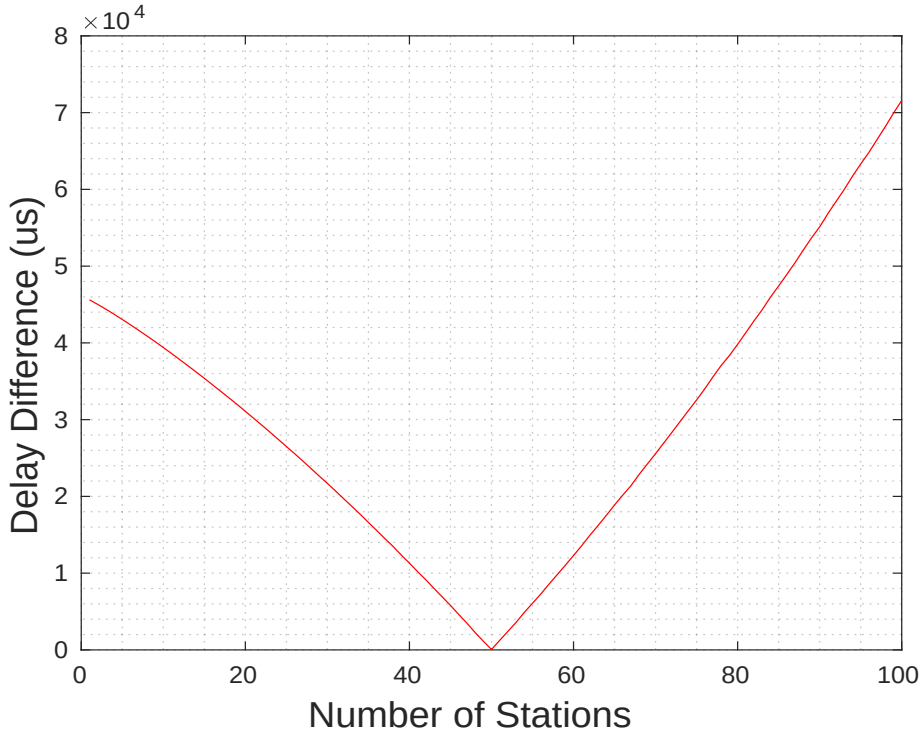

Figure 3.40: Test 1C: Static Round Robin Link Delay Difference

3.5.1.6 RR Test 1C: Discussion

The main finding from Test 1C was that the achieved mean round robin bond throughput of 1.2422 Mbps was less than 1% of the theoretical maximum bond throughput.

The throughput was the second highest of the unbalanced load scenarios for the homogeneous 11ac-11ac bond, which was expected due to the moderate static load of n = 50 on the second 11ac link. The overall mean bond throughput efficiency was 99.19% and the highest of all three unbalanced load scenarios and just 0.8% lower than the static round robin performance in the benchmark case in Test 1A.

The observed behaviour differs to Test 1B where the difference in applied load was always increasing. In this test we see the load on the first 11ac link of the bond increasing until eventually becoming equal to and then surpassing the static load of n = 50 on the second 11ac link. Therefore we see three different types of behaviour in the test:

1. Increasing Bandwidth Utilisation: For $\{n_1 < 50, n_2 = 50\}$ the applied load on the first 11ac link was increased towards that of the second thereby reducing the performance disparity and increasing the efficiency but not increasing the overall achieved bond throughput as the speed of the bond was still limited by the second 11ac link with the static load of $n = 50$.

2. Maximum Bandwidth Utilisation: At an applied load of $\{n_1 = 50, n_2 = 50\}$ the achieved throughputs and delays of each bond link were approximately equal. For the homogeneous 11ac-11ac bond this creates the optimal conditions for maximising the aggregated bond throughput and results in minimal or no throughput wastage due to the slow-down effect similar to the results seen throughout in the benchmark case in Test 1A.

3. Decreasing Bandwidth Utilisation: For $\{n_1 < 50, n_2 = 50\}$ the applied load on the first 11ac link was increased *beyond* that of the second link and therefore the overall bond speed became limited by the first link with the variable load. The bond throughput therefore reduced with the individual Link 1 throughput as the applied load increased and the link utilisation also decreased as the performance disparity became larger.

Note that on first examination of Figure 3.36 both the achieved bond throughput and the maximum theoretical bond throughput appear to display a similar straightforward exponentially decreasing curve to Test 1B. However a closer inspection reveals that for $\{n_1 < 50, n_2 = 50\}$ the difference between the two metrics was gradually reducing as the applied load was increased. The same general behaviour is reflected in all measured performance metrics for the test. The link bandwidth utilisation in Figure 3.38 reaches a maximum of 100% at $\{n_1 = 50, n_2 = 50\}$ and then decreases non-linearly for $\{n_1 > 50, n_2 = 50\}$ as the impact of the slow-down effect lessens. The behaviour was also reflected in the bond delay plot in Figure 3.39 where for $\{n_1 >= 50, n_2 = 50\}$ the bond delay was constant and following the Link 2 delay with the static load while for $\{n_1 > 50, n_2 = 50\}$ it followed the variable delay of Link 1. Note that because both 11ac links

have identical throughput and delay responses the load at which the individual link throughput performance becomes equal was easily identified as n = 50.

Another key finding from Test 1C is illustrated in Figure 3.49 below. Notably for applied loads where $\{n_1 <= 14, n_2 = 50\}$ the achieved static round robin bond throughput was lower than that of the single fastest available link, which was in this case 11ac Link 1 with the variable applied load. This scenario therefore renders the use of bonding counter productive to the goal of increasing the overall throughput.

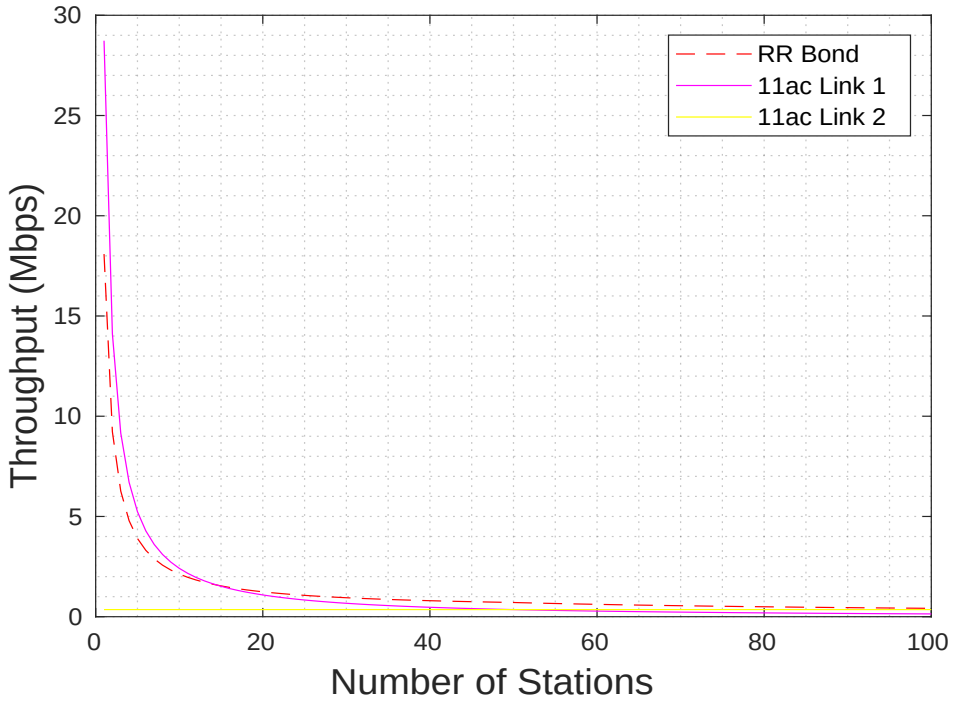

Figure 3.41: Test 1C Static Round Robin Link Throughput Comparison

Figure 3.42 below shows how the achieved bond throughput was impacted by a change in the data rate of the first 11ac link with the variable load. Again as per the previous test the results show that for all data rate pairings considered that the overall compound throughput of the bond is *always* limited by the speed of the slower 65 Mbps bond link.

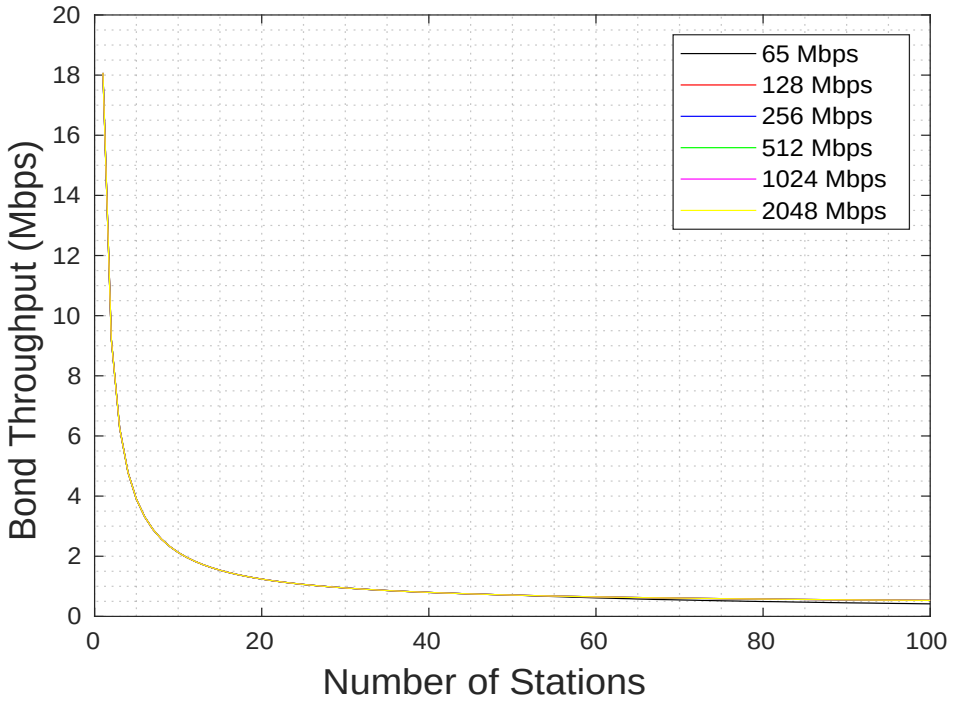

Figure 3.42: Test 1C Static Round Robin Throughput (Heterogeneous Data Rates)

The same behaviour was also reflected in the bandwidth utilisation results in Figure 3.43. The utilisation of the 65Mbps-65Mbps bond reached a maximum of 100% at $\{n_1 = 50, n_2 = 50\}$, while for the remaining 65Mbps-128Mbps, 65Mbps-512Mbps, 65Mbps-1024Mbps, and 65Mbps-2048Mbps device pairings the utilisation increased with the applied load but never reached the maximum value.

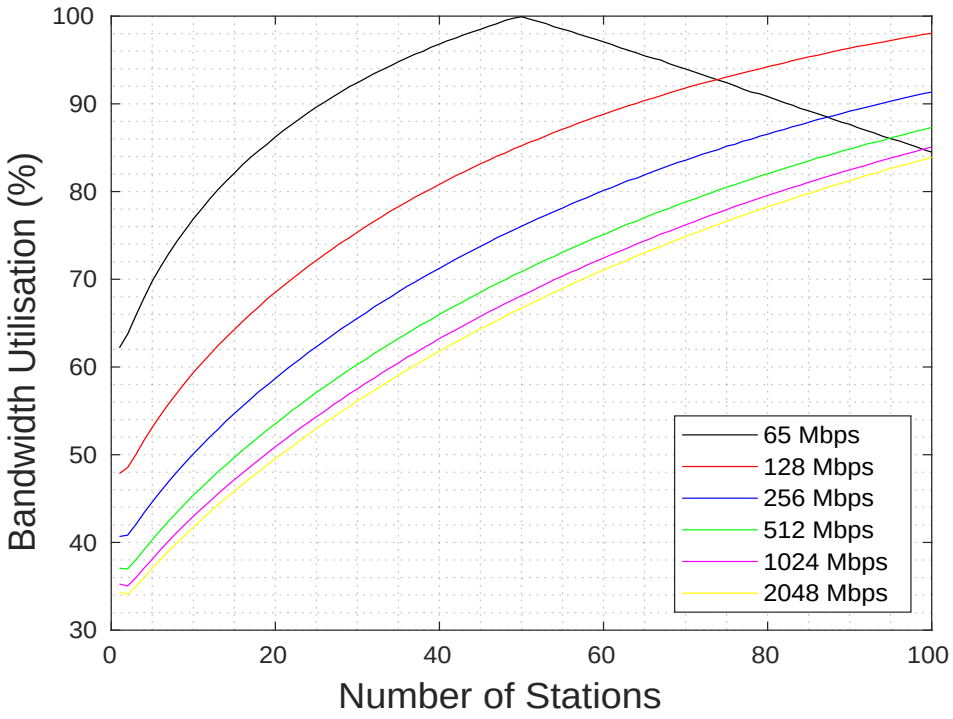

Figure 3.43: Test 1C Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.5.1.7 RR Test 1D: Results

The bond throughput results for the static round robin configuration in Test 1D are given in Figure 3.44 below. The overall mean achieved bond throughput for all applied load combinations considered was 0.85805 Mbps. As with the previous tests a similar exponentially decreasing curve was seen for both the achieved bond throughput and the maximum theoretical bond throughput. At the lightest load of $\{n_1 = 1, n_2 = 100\}$ the theoretical maximum throughput was 28.87 Mbps while the actual bond throughput was 51.33% lower at 14.05 Mbps. Then as the applied load was increased both metrics decreased exponentially while the difference between them became smaller until finally at the heaviest considered load of $\{n_1 = 100, n_2 = 100\}$ the achieved throughput of 0.278 Mbps was exactly equal to the maximum theoretical bond throughput.

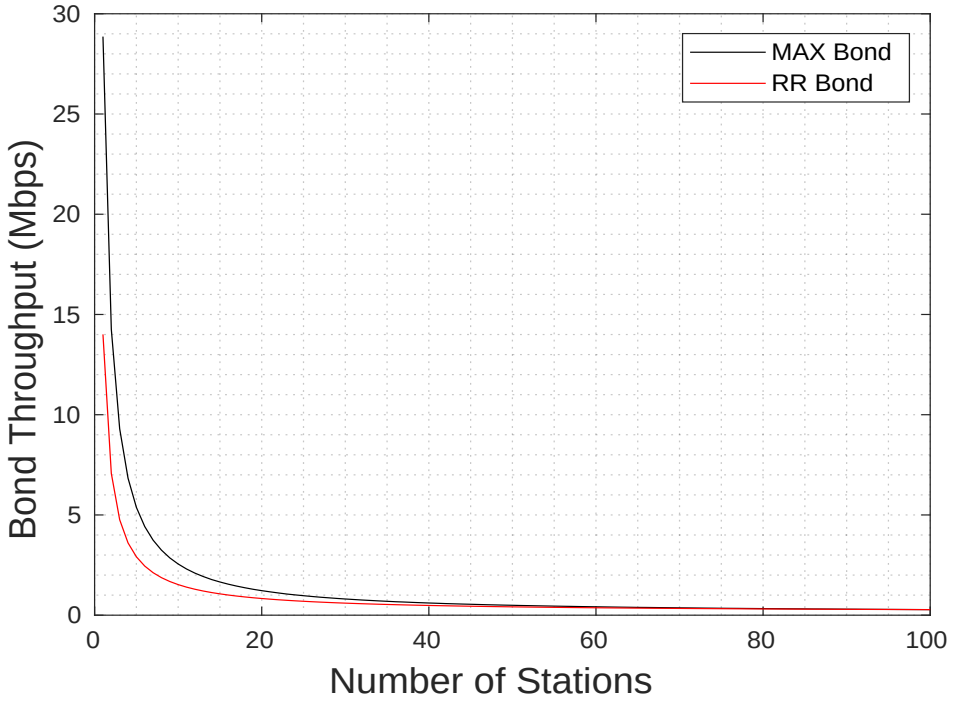

Figure 3.44: Test 1D Static Round Robin Bond Throughput

The link throughput difference results for static round robin in Test 1D are shown in Figure 3.45 below. The overall mean link throughput difference for all applied loads considered was 1.1 Mbps. An exponentially decreasing curve was observed as the applied bond load was increased. At the lightest load of $\{n_1 = 1, n_2 = 100\}$ the difference between the achievable throughputs of each bonded link was measured at a maximum value of 28.59 Mbps. As the applied load was raised towards the heaviest prescribed level of $\{n_1 = 100, n_2 = 100\}$ the link throughput difference reduced gradually to a minimum value of 0.02392 kbps.

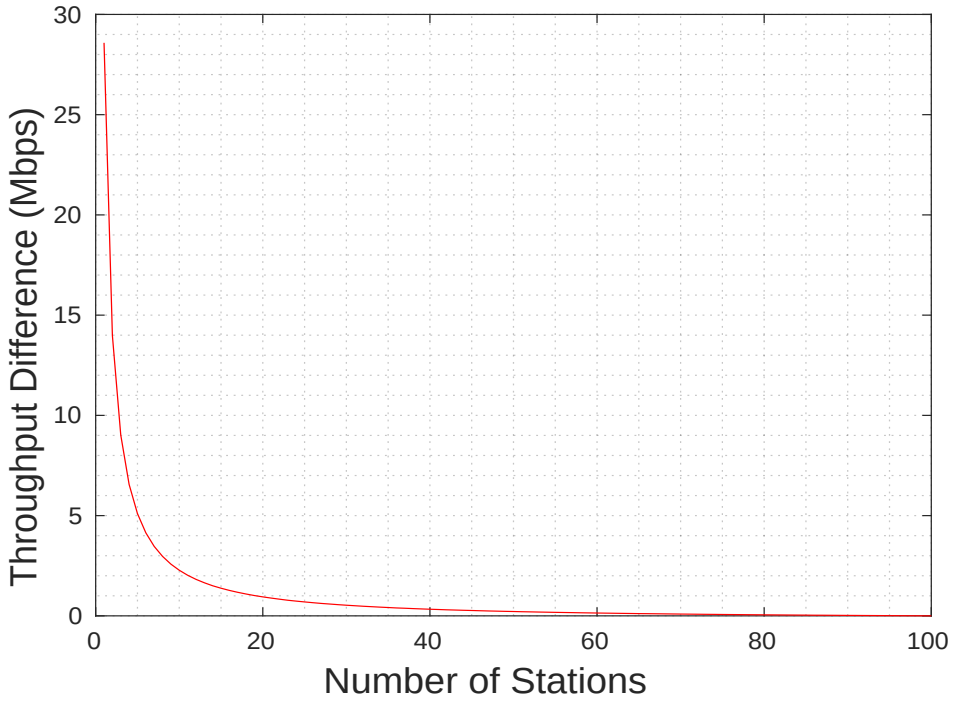

Figure 3.45: Test 2D Static Round Robin Link Throughput Difference

The link bandwidth utilisation results for static round robin in Test 1D are shown in Figure 3.46 below. The overall mean utilisation for all applied load used was 81.615%. A non-linear increasing curve was seen as the applied bond load increased. At $\{n_1 = 1, n_2 = 100\}$ the utilisation was measured at a minimum value of 48.51%. Then the load was gradually increased towards $\{n_1 = 100, n_2 = 100\}$ the achieved link utilisation increased non-linearly to a maximum value of 99.97%.

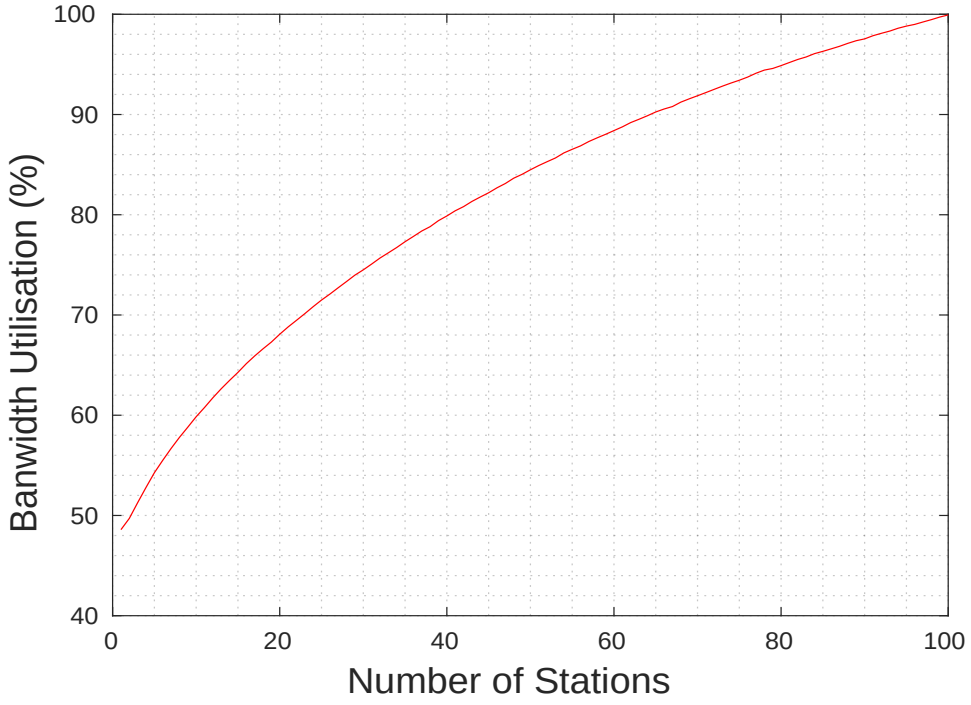

Figure 3.46: Test 1D Static Round Robin Link Bandwidth Utilisation

The bond delay results for the static round robin configuration in Test 1D are shown in Figure 3.47. The bond delay is plotted as the dotted black line and the individual Link 1 and Link 2 delays as the solid blue and red lines respectively. Notably for the entire range of applied loads considered from $\{n_1 = 1, n_2 = 100\}$ to $\{n_1 = 100, n_2 = 100\}$ the bond delay followed the statically loaded Link 2 delay which remained constant at 116700 μs.

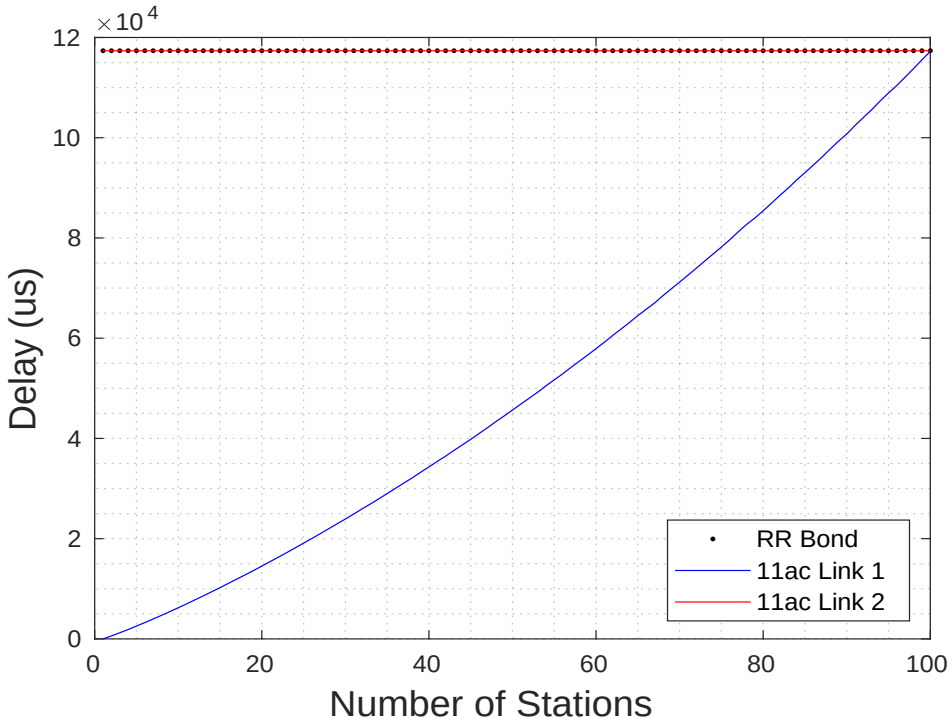

Figure 3.47: Test 1D Static Round Robin Bond Delay

The delay difference results for static round robin in Test 1D are shown in Figure 3.48 below. The overall mean link delay difference for all applied loads considered was 66917.11 μs. At $\{n_1 = 1, n_2 = 100\}$ the delay difference was measured at a maximum value of 116700 μs. Then as the applied load was increased towards $\{n_1 = 100, n_2 = 100\}$ the resulting link delay difference decreased non-linearly to a minimum of 222.9 μs.

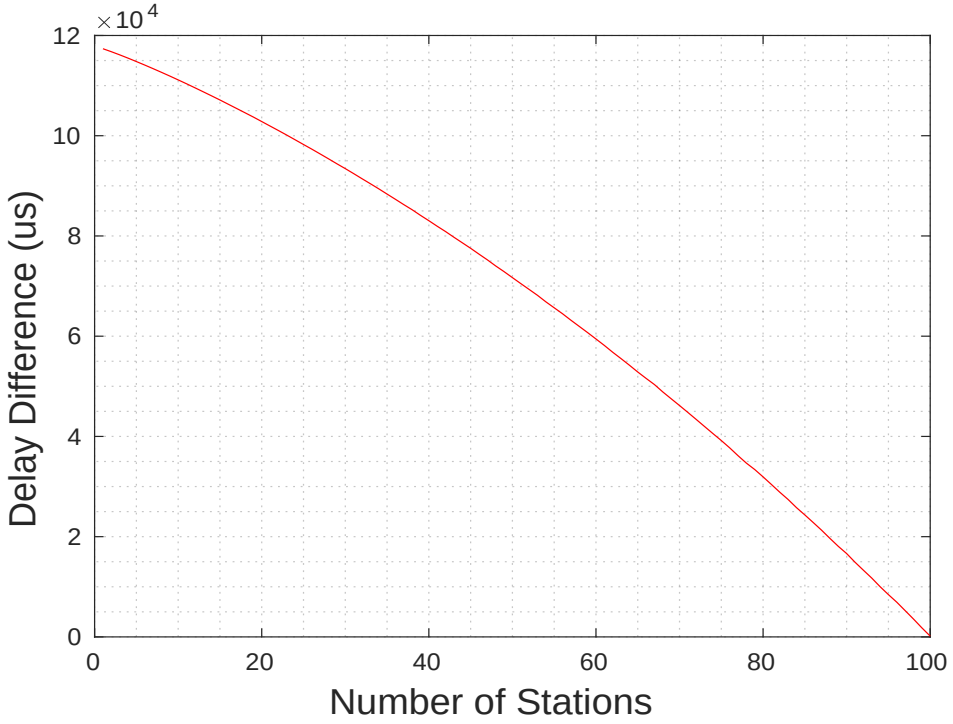

Figure 3.48: Test 1D Static Round Robin Link Delay Difference

3.5.1.8 RR Test 1D: Discussion

The main result from Test 1D was that the static round robin mean bond throughput of 0.85805 Mbps was 18% lower than the theoretical maximum bond throughput due to the severe throughput slow-down effect observed. Furthermore the achieved bond throughput was the lowest of all four tests due to the high static load of $n = 100$ applied to the second 11ac link. The achieved link utilisation was the lowest of the three unbalanced tests at 81.165%.

In Test 1D again conditions (1) and (2) of Section 3.5.1.2 above were met but requirement (3) was unsatisfied. In the tests as the load on the first link was increased its achievable throughput was gradually reduced towards that of the second link which decreased the performance disparity between them and lessened the impact of the bond throughput slow-down effect. This was reflected in the link bandwidth utilisation in Figure 3.46 which displayed a non-linear increasing curve reaching a maximum value of 100% at $\{n_1 = 100, n_2 = 100\}$. Here the individual link throughput performance of both links was identical due to the balanced load as seen in the link throughput difference in Figure 3.45. This created the ideal conditions for maximising the bond throughput as seen throughout the balanced benchmark scenario in Test 1A. The same general behaviour was seen in the delay plot for Test 1D in Figure 3.47 which showed that the overall bond delay was constant at 116700 μs and followed the Link 1 delay throughout.

The other key finding from Test 1D is illustrated in Figure 3.49. For applied loads where $\{n_1 <= 37, n_2 = 50\}$ the static round robin throughput was actually substantially lower than the single fastest available link, which was in this case 11ac Link 1 with the variable applied load. Again this therefore renders the use of bonding counter productive to increasing the overall throughput.

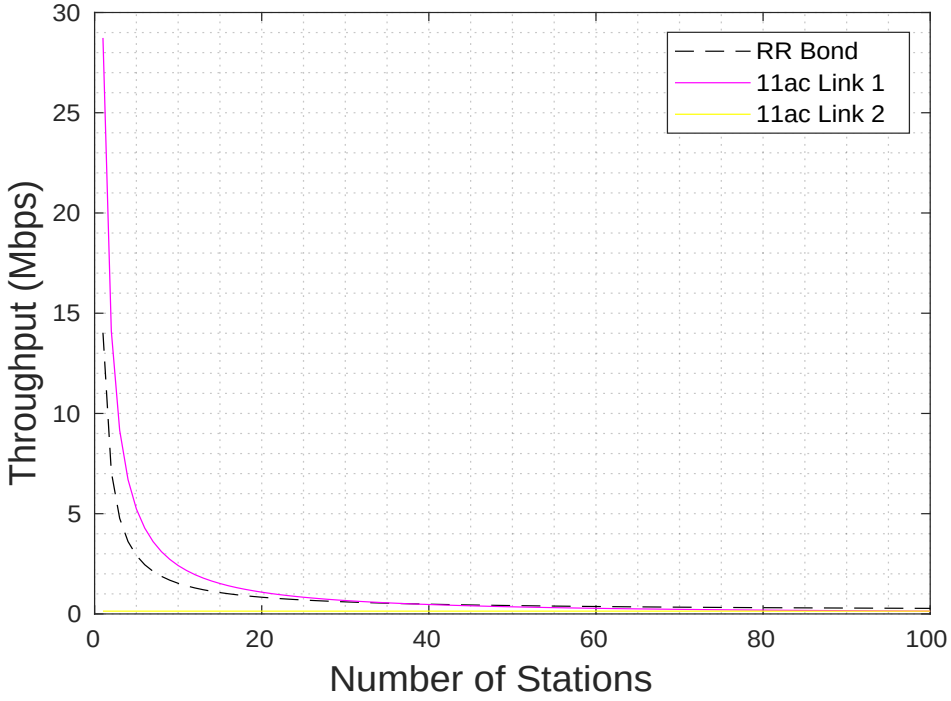

Figure 3.49: Test 1D Static Round Robin Link Throughput Comparison

Figure 3.50 below shows how the bond throughput was impacted by a change in the raw PHY data rate of the variably loaded first 11ac link. As per the previous tests for all data rate pairings and all applied loads considered from $\{n_1 = 1, n_2 = 100\}$ to $\{n_1 = 100, n_2 = 100\}$ the aggregate bond throughput was *always* constrained by the individual throughput of the static and highly loaded second 11ac link.

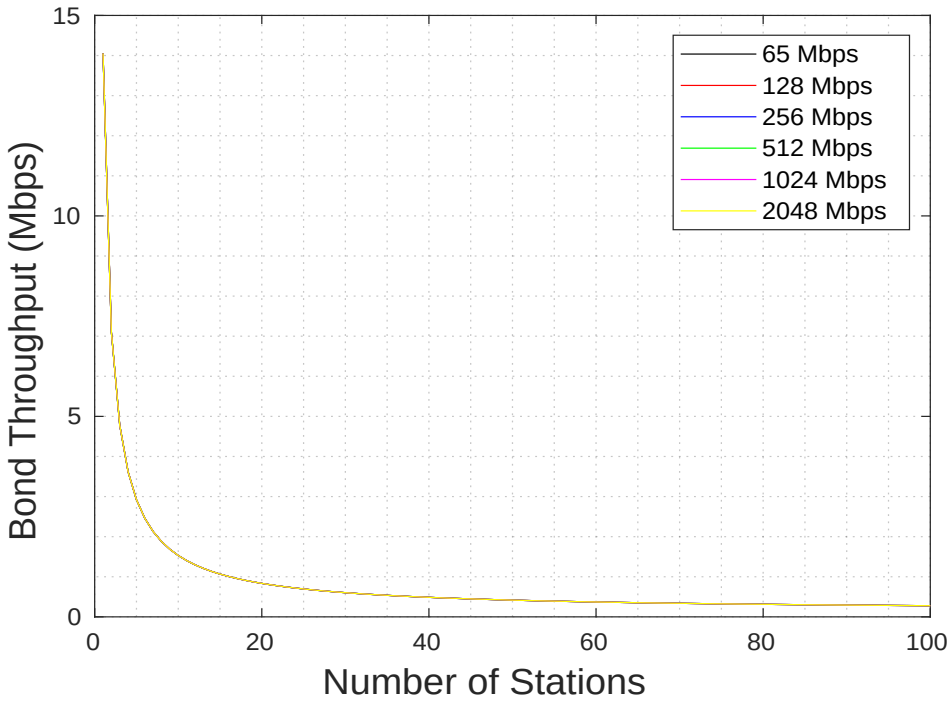

Figure 3.50: Test 1D Static Round Robin Bond Throughput (Heterogeneous Data Rates)

The behaviour was also reflected in the link throughput difference shown in Figure 3.51 below. In general a decreasing throughput difference was observed as the applied load on the first bond link was increased. However only for the 65Mbps-65Mbps device pairing was the throughput difference reduced to zero which occurred precisely at $\{n_1 = 100, n_2 = 100\}$.

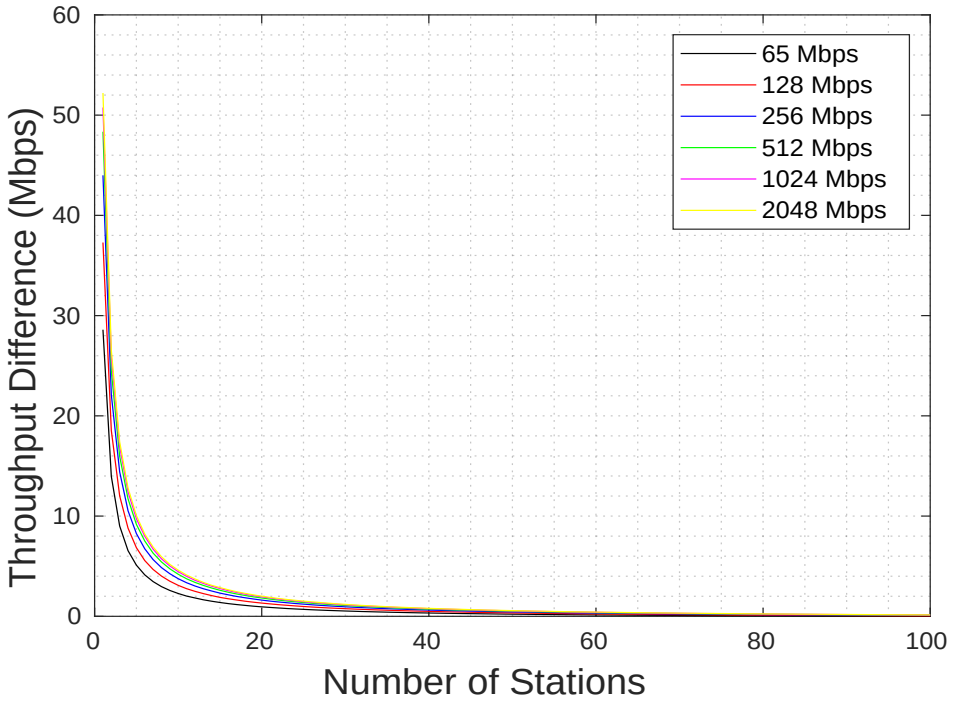

Figure 3.51: Test 1D Static Round Robin Throughput Difference (Heterogeneous Data Rates)

The link bandwidth utilisation results in Figure 3.52 below also show the impact of above change in the Link 1 PHY data rate. For all data rate pairings the utilisation increased with the applied load but never reached 100% with the exception of the 65Mbps-65Mbps device pairing for which this occurred exactly at $\{n_1 = 100, n_2 = 100\}$.

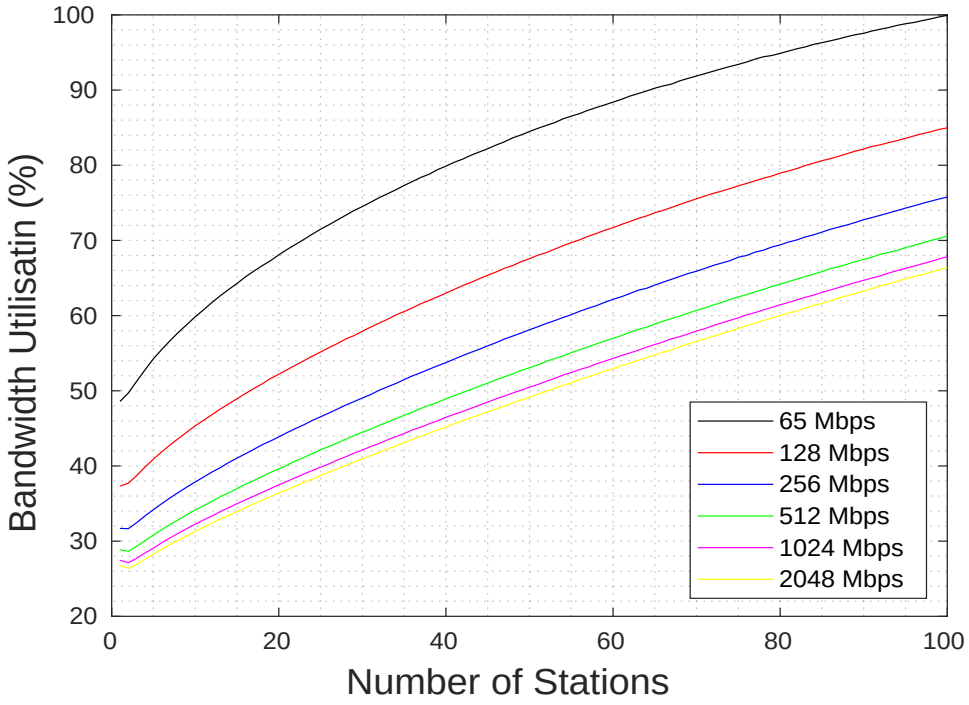

Figure 3.52: Test 1D Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.5.2 Homogeneous 11ac-11ac Bond

Next we present the results for the heterogeneous 11ac-11ah bond in Tests 2A-2D.

3.5.2.1 RR Test 2A: Results

The bond throughput results for the static round robin configuration and balanced applied load in Test 2A are shown in Figure 3.53. The round robin bond throughput is plotted as the red line and the maximum theoretical 11ac-11ah bond throughput as the solid black line. The overall mean bond throughput for all applied load combinations used in the test was 1.299 Mbps and the mean theoretical maximum bond throughput was 1.8897 Mbps. A non-linear decreasing output was seen for both metrics as the applied load was raised and the achieved round robin throughput was less than the theoretical maximum bond throughput throughout. At $\{n_{11ac} = n_{11ah} = 1\}$ the achieved throughput was 26.32 Mbps and the theoretical maximum bond throughput was 41.89 Mbps. As the load incremented to $\{n_{11ac} = n_{11ah} = 2\}$ the bond throughput reduced to 14.04 Mbps and the maximum bond throughput to 21.14 Mbps. The exponential decrease continued with the round robin bond throughput reducing to a minimum of 0.1651 Mbps as the applied load was gradually raised towards the heaviest considered level of $\{n_{11ac} = n_{11ah} = 100\}$ while the maximum theoretical bond throughput reduced to 0.2216 Mbps.

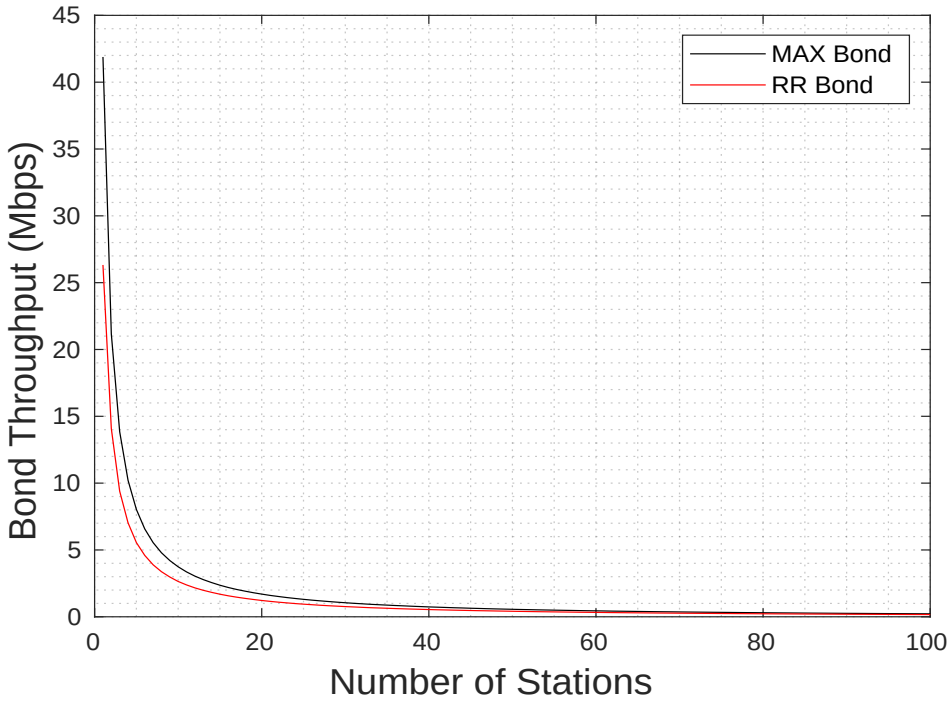

Figure 3.53: Test 2A Static Round Robin Bond Throughput

The link throughput difference results for static round robin in Test 2A are shown in Figure 3.54. The mean link delay difference for all applied loads used was 0.59016 Mbps. An exponentially decreasing curve was seen as the load was increased on both links simultaneously. At $\{n_{11ac} = n_{11ah} = 1\}$ the throughput difference was measured at a maximum of 15.55 Mbps and when the applied load increased towards $\{n_{11ac} = n_{11ah} = 100\}$ it reduced non-linearly to a minimum of 0.05651 Mbps.

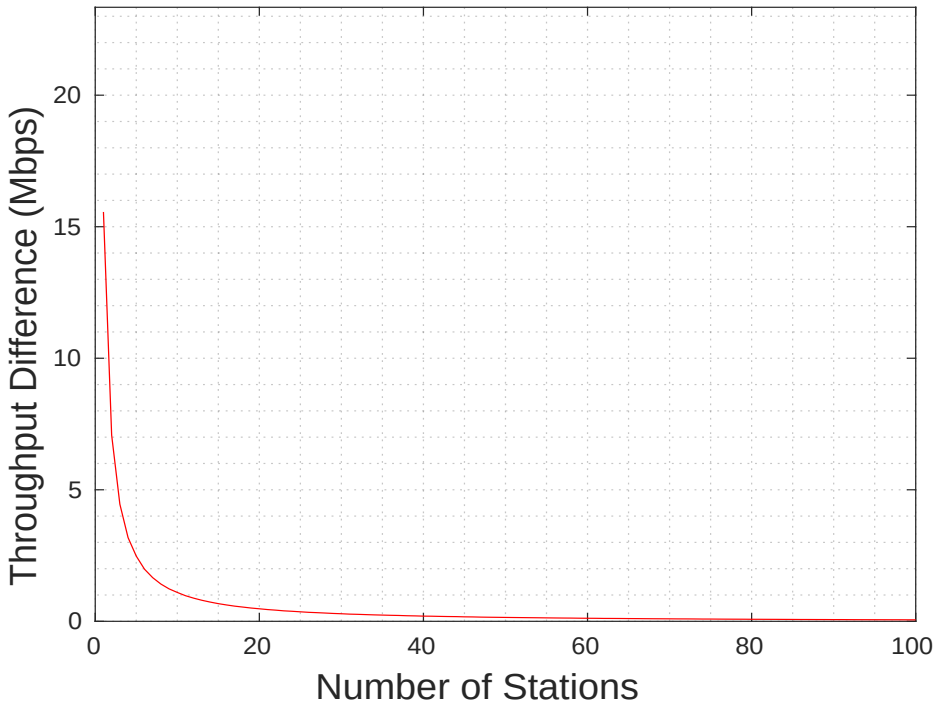

Figure 3.54: Test 2A Static Round Robin Link Throughput Difference

The link bandwidth utilisation results for static round robin in Test 2A are shown in Figure 3.55. The overall mean utilisation for all applied load considered was 72.981%. A non-linear increasing curve was seen in the simulation output as the applied bond load was raised. At the initial load of $\{n_{11ac} = n_{11ah} = 1\}$ the achieved utilisation was measured at a minimum of 62.85% but as the load was increased towards $\{n_{11ac} = n_{11ah} = 100\}$ the utilisation increased gradually to a maximum test value of 74.5%.

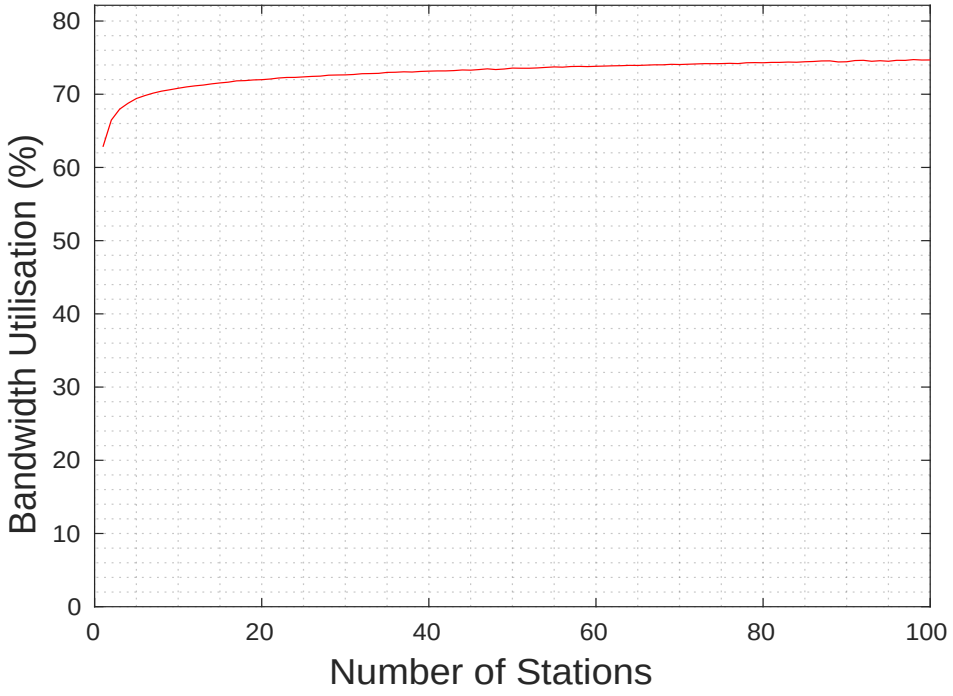

Figure 3.55: Test 2A Static Round Robin Link Bandwidth Utilisation

The bond delay results for the static round robin configuration in Test 2A is presented in Figure 3.56 below. The overall bond delay is plotted as the dotted black line and the individual 11ac and 11ah link delays as the solid blue and dashed red lines respectively. The mean bond delay for all applied load combinations considered was 85313.46 μs. A non-linear increasing curve was observed in the simulation output as the applied bond load was increased. Notably for all applied loads considered the bond delay followed the 11ah link delay which increased non-linearly from a minimum of 0.0136 μs at the initial load of $\{n_{11ac} = n_{11ah} = 1\}$ to a maximum value of 196500 μs at $\{n_{11ac} = n_{11ah} = 100\}$.

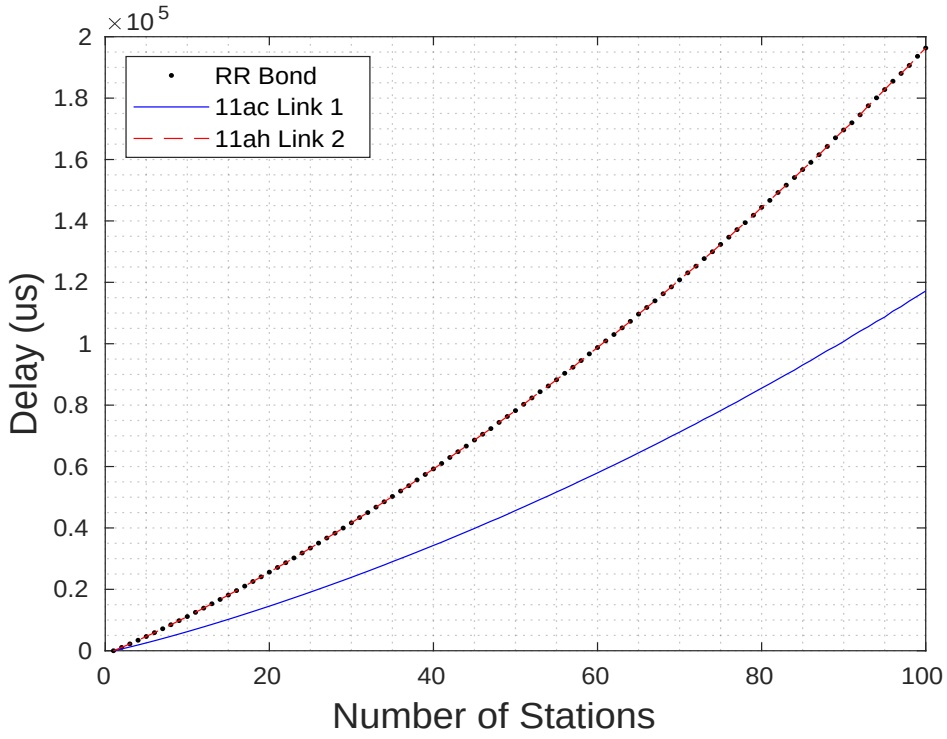

Figure 3.56: Test 2A Static Round Robin Bond Delay

The link delay difference results for the static round robin configuration in Test 2A are given in Figure 3.57 below. The overall mean delay difference between the 11ac and 11ah links for all applied load combinations used was 35112.9 μs. A non-linear increasing curve was seen in the simulation output as the applied bond load was increased. At the initial load of $\{n_{11ac} = n_{11ah} = 1\}$ the delay difference between each bonded link was measured at a minimum of 0.00122 μs. Then as the applied load was raised towards $\{n_{11ac} = n_{11ah} = 100\}$ the delay difference increased gradually to a maximum overall test value of 79630 μs.

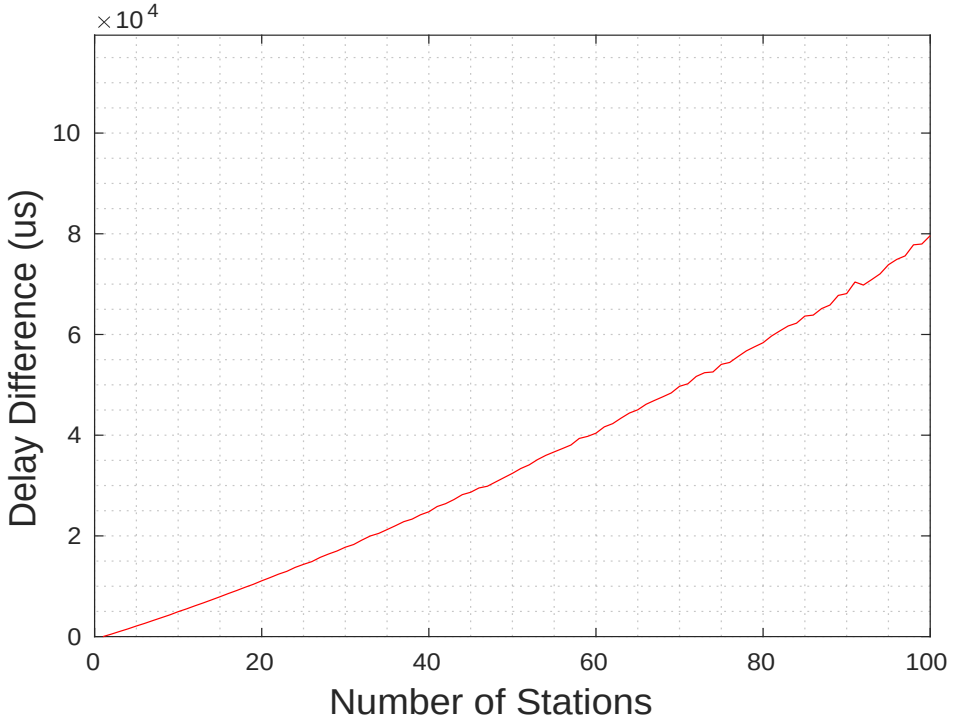

Figure 3.57: Test 2A Static Round Robin Link Delay Difference

3.5.2.2 RR Test 2A: Discussion

The main result from Test 2A was that the although a balanced load was applied this did not lead to an optimal throughput performance as was the case for the homogeneous 11ac-11ac bond in Test 1A. The overall mean link utilisation for the heterogeneous 11ac-11ah bond in Test 2A was 72.98% with a mean bond throughput of 1.299 Mbps.

The main factor contributing to the poor throughput performance seen in the results was that the heterogeneous 11ac and 11ah device types have an entirely different set of DCF parameter values, i.e. although conditions (1) and (3) of Section 3.5.1.2 above were satisfied in Test 2A requirement (3) was not. In spite of both 802.11 types using the saw raw PHY data rate of 65 Mbps the larger slot size and inter-frame spacing used in 11ah creates an uneven delay response between the two bond links as the applied load is raised on both devices simultaneously. The knock-on reduction in the 11ah link throughput leads to the bond becoming unbalanced and susceptible to the inherent slow-down effect which causes a reduction in the achieved bond throughput and link bandwidth utilisation.

The unbalance is seen in the delay difference in Figure 3.57 which became larger as the applied bond load was raised. Furthermore the bond delay in Figure 3.56 followed the delay of the slower 11ah link throughout the test. The difference in achieved throughput is seen in Figure 3.54 where initially at $\{n_{11ac} = n_{11ah} = 1\}$ the 11ah throughput was almost 16 Mbps slower compared to 11ac link due to the difference in DCF parameters and achieved access delay.

This in turn impacted the achieved bond throughput as the performance disparity resulted in a severe slow-down effect as the faster 11ac link was forced to wait for the slower 11ah link to finish sending its data before the bonded transmission could be considered complete. This can be seen in Figure 3.53 where there is a visible difference between the achieved bond throughput and the theoretical maximum bond throughput. As the applied load was increased the achievable throughput of both links reduced but the 11ac throughput reduced at a faster rate therefore making the performance of the two links more similar and minorly increasing the achieved link bandwidth utilisation.

The other key finding from Test 2D is that for applied loads where $\{n_{11ac} = n_{11ah} = 1\}$ the achieved static round robin bond throughput was substantially lower than the single fastest available link, in this case the 11ac Link 1 with the variable applied load, therefore rendering the use of bonding counter productive to the main goal of increasing the overall throughput. This can be seen from the graph below in Figure 3.58.

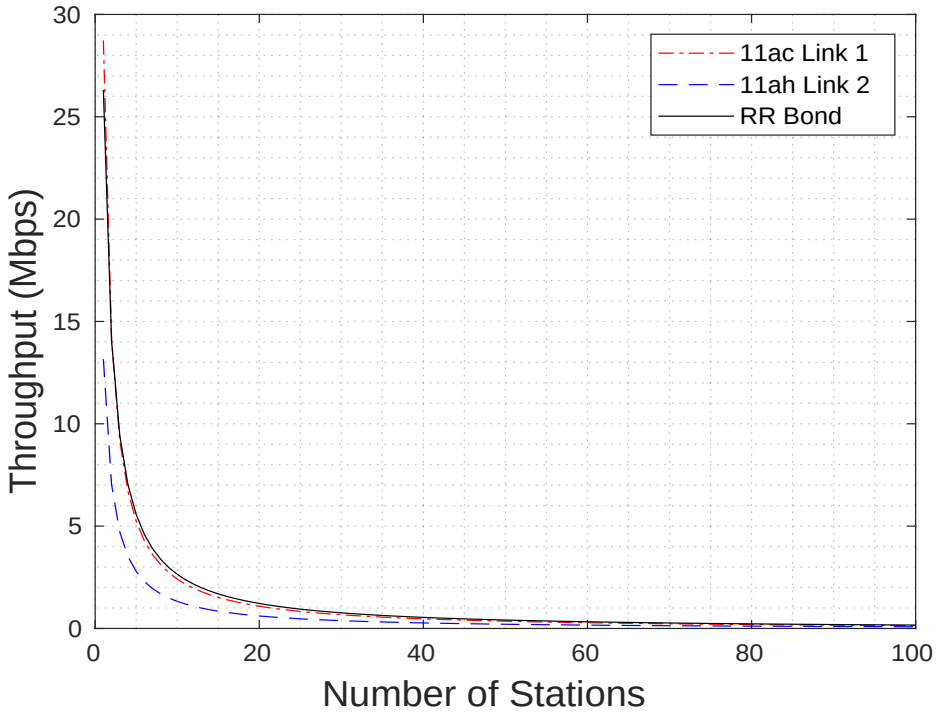

Figure 3.58: Test 2A Static Round Robin Link Throughput Comparison

To demonstrate what happens when condition (1) in Section 3.5.1.2 above is not satisfied Figure 3.59 below shows how the bond throughput is impacted by an increase in the raw PHY data rate of the 11ac slave link. Notably as per the results in Test 1A for all data rate pairings evaluated the overall bond throughput was *always* limited by the throughput of the slower 65Mbps 11ah device. Again this was true regardless of the capacity of the higher speed 11ac link.

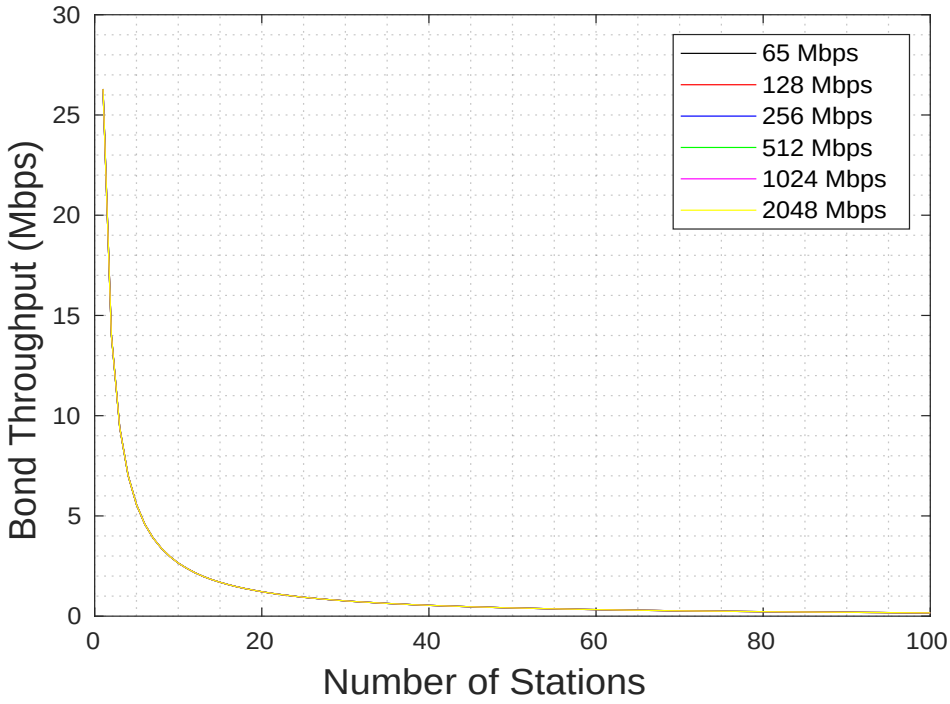

Figure 3.59: Test 2A Static Round Robin Throughput (Heterogeneous Data Rates)

Similarly Figure 3.60 below which shows how the link bandwidth utilisation was influenced by an increase in the PHY data rate of the 11ac link. As seen below each data rate pairing produced a non-linear increasing utilisation output as the applied load was raised on both links simultaneously. In general the higher the throughput of the first 11ac link the lower the overall bond throughput efficiency. As expected the dual 65 Mbps pairing achieved the highest overall efficiency while the worse performance was seen for the 2048Mbps-65Mbps bond.

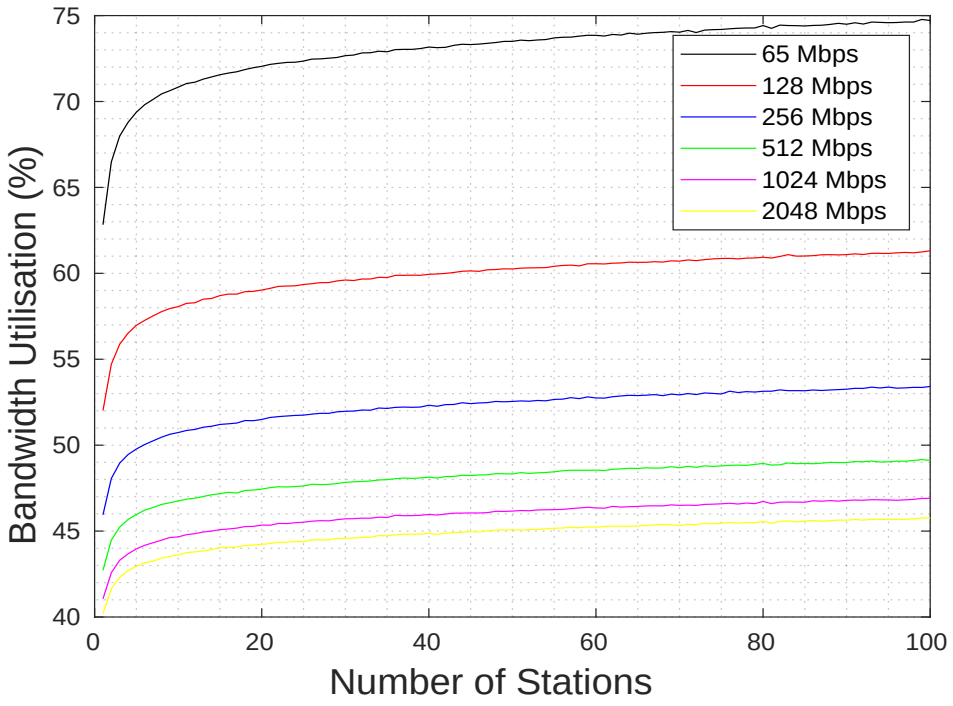

Figure 3.60: Test 2A Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.5.2.3 RR Test 2B: Results

The bond throughput results for the static round robin configuration in Test 2B are presented in Figure 3.61 below. The achieved round robin throughput is plotted as the solid red line and the maximum theoretical bond throughput as the solid black line. In the simulation output a non-linear decreasing curve was seen for both metrics and the achieved bond throughput was lower throughout compared to the theoretical maximum. The overall mean bond throughput for all applied loads used was 13.8537 Mbps and the maximum theoretical mean was 14.402 Mbps. At $\{n_{ac} = 1, n_{ah} = 1\}$ the achieved bond throughput was 26.34 Mbps but as the load was increased towards $\{n_{ac} = 100, n_{ah} = 1\}$ the output decreased non-linearly to a minimum of 0.163 Mbps. Similarly the maximum theoretical bond throughput decreased from a maximum of 41.89 Mbps at $\{n_{ac} = 1, n_{ah} = 1\}$ to a minimum of 0.2221 Mbps at the heaviest considered load of $\{n_{ac} = 100, n_{ah} = 1\}$.

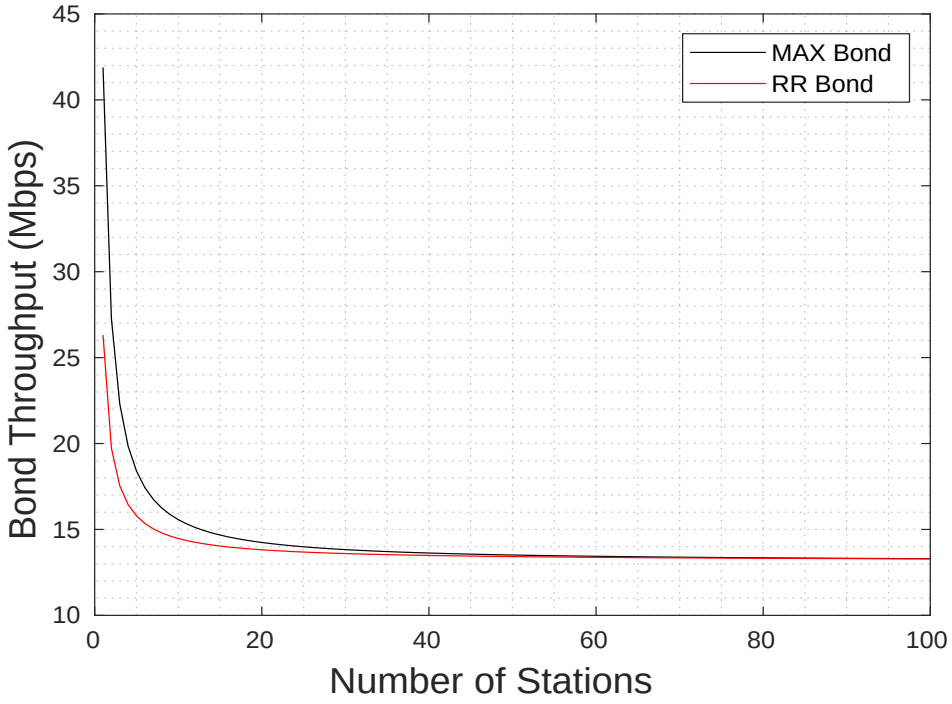

Figure 3.61: Test 2B Static Round Robin Bond Throughput

The mean link throughput difference for the heterogeneous 11ac-11ah bond in Test 2B was 12.2602 Mbps. The simulation results are plotted in Figure 3.62 below with a non-linear increasing curve seen as the applied bond load was raised. At $\{n_{ac} = 1, n_{ah} = 1\}$ the link throughput difference was measured at a maximum of value 15.6 Mbps. Then the applied load was raised towards the heaviest considered level of $\{n_{ac} = 100, n_{ah} = 1\}$ the link throughput difference gradually reduced to a test minimum of 0.05683 Mbps.

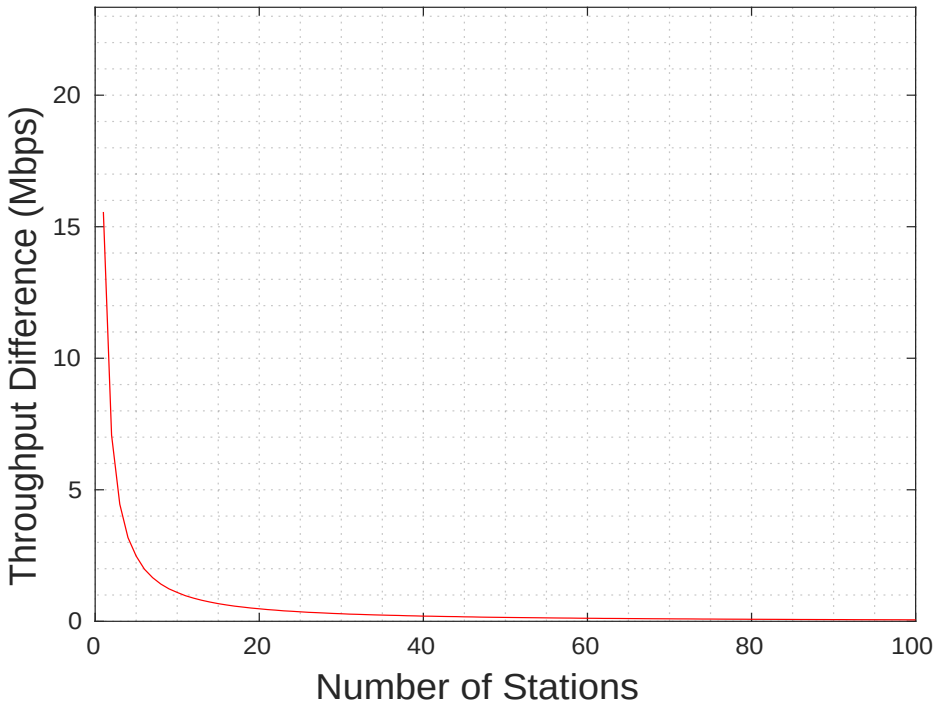

Figure 3.62: Test 2B Static Round Robin Link Throughput Difference

The link bandwidth utilisation results for the static round robin bond in Test 2B are given in Figure 3.63 below. As seen from the graph a non-linear increasing curve was observed throughout as the applied bond load was raised. At $\{n_{ac} = 1, n_{ah} = 1\}$ the achieved utilisation was measured at a minimum value of 62.88%. Then as the applied load was raised towards $\{n_{ac} = 100, n_{ah} = 1\}$ the efficiency increased gradually to a maximum of 99.94%. The overall mean link utilisation for all applied loads considered was 97.363%.

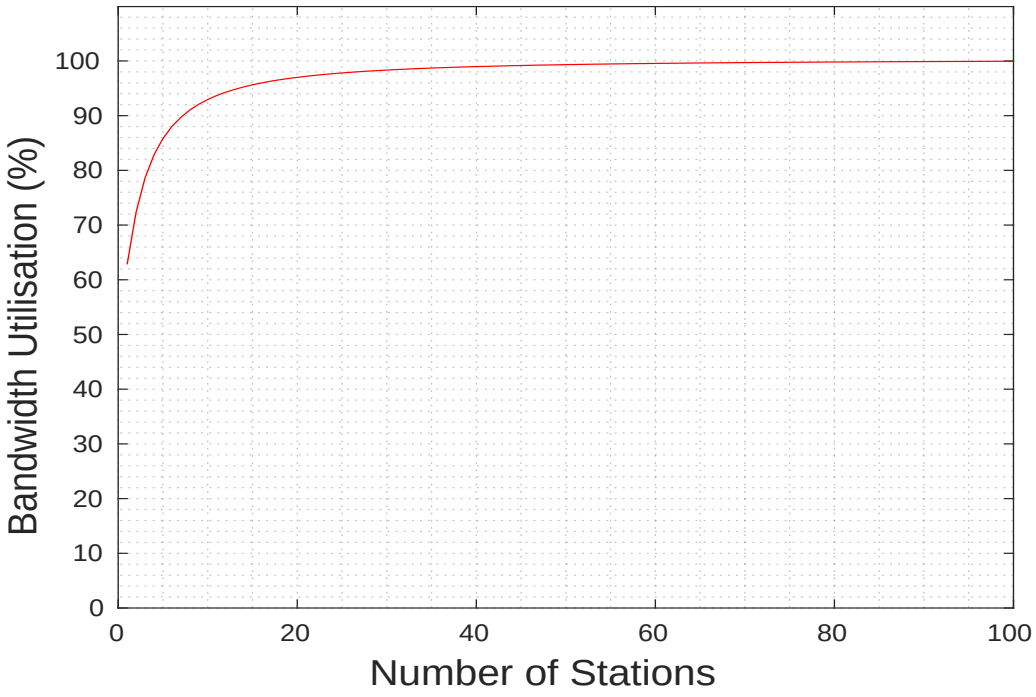

Figure 3.63: Test 2B Static Round Robin Link Bandwidth Utilisation

The delay results for the static round robin configuration for the homogeneous 11ac-11ah bond in Test 2B are provided in Figure 3.64. The bond delay is plotted as the dotted black line and the individual 11ac and 11ah link delays as the dashed red and dotted blue respectively. The overall mean bond delay for all applied load combinations considered was 50197.7 μs. As seen below for all applied loads from $\{n_{ac} = 1, n_{ah} = 1\}$ to $\{n_{ac} = 100, n_{ah} = 1\}$ the bond delay followed the variable access delay of the 11ah link and increased non-linearly from 0.00736 μs to 196300 μs.

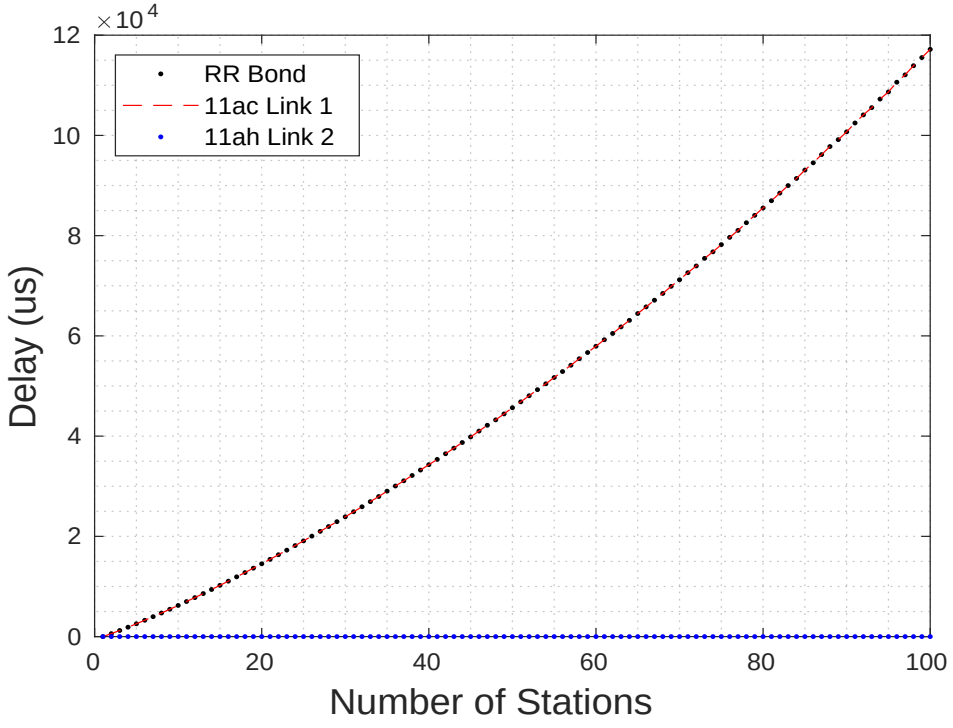

Figure 3.64: Test 2B Static Round Robin Bond Delay

The link delay difference results for Test 2B are shown in Figure 3.65 below with a non-linear increasing curve seen across the range of simulation output. The overall mean link delay difference for all applied load combinations used was 50197.7 μs. At the lightest load of $\{n_{ac} = 1, n_{ah} = 1\}$ the delay difference was measured at a test minimum of 0.00434 μs and as the applied load was raised towards $\{n_{ac} = 100, n_{ah} = 1\}$ the output increased gradually to a maximum value of 79810 μs.

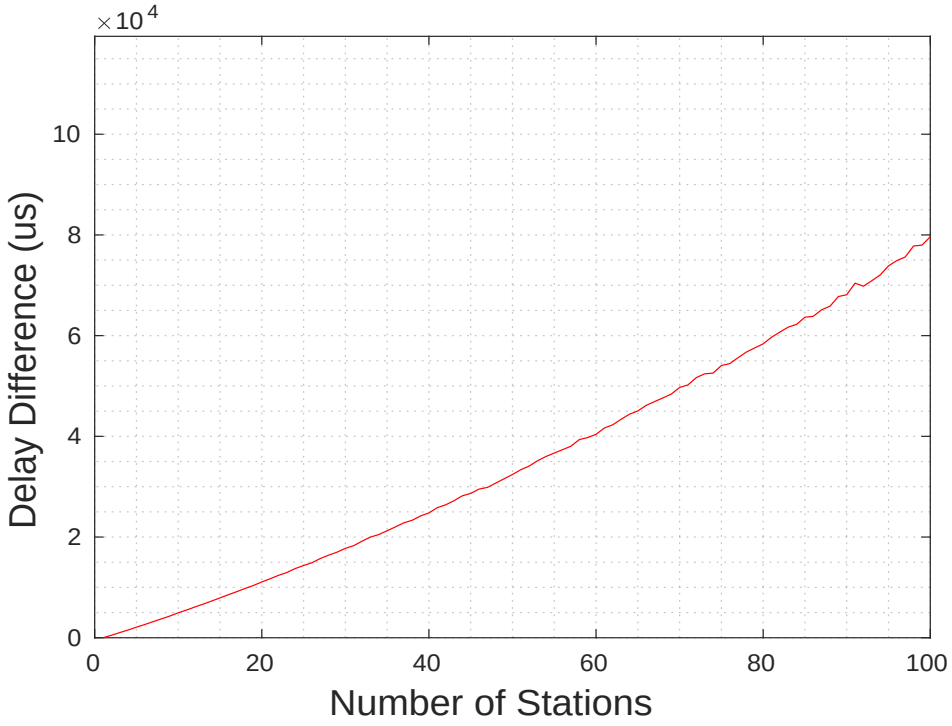

Figure 3.65: Test 2B Static Round Robin Link Delay Difference

3.5.2.4 RR Test 2B: Discussion

The main result from Test 2B was that under the static round robin configured the achieved mean throughput for the heterogeneous 11ac-11ah bond was 3.64% lower than the theoretical maximum due to the impact of the slow-down effect for the unbalanced applied load scenario considered.

Test 2B demonstrates the impact of the throughput slow-down effect for static round robin when there is a difference in applied load on each link of a heterogeneous 11ac-11ah bond. The test used the first unbalanced scenarios to stress the 11ac-11ah bond throughput performance by lightly loading the 11ah link with a single node and varying the 11ac load from n = 1 to n = 100. The 11ac-11ah bond throughput in Figure 3.61 above shows the same general non-linear decreasing behaviour as the dual 11ac bond in Test 1B. However due to condition (2) in Section 3.5.1.2 above not being met for the test at an applied load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ there was already a substantial performance disparity between the 11ac and 11ah links creating a noticeable slow-down effect. It is difficult to determine the general behaviour by inspection of the bond throughput alone due to the increased resource sharing in the 11ac DCF channel so examining the link bandwidth utilisation results in Figure 3.63 above provides better insight. As seen from the graph the bandwidth utilisation *increased* as the 11ac link load was raised due to the decrease in the achievable throughput of the link which reduced the performance disparity between the two links.

Another key finding from Test 2B is that for applied loads where $\{n_{11ac} = 1, n_{11ah} = 1\}$ the achieved static round robin bond throughput was substantially lower than the single fastest available link, in this case the 11ac Link 1 with the variable load, therefore rendering the use of bonding counter productive to increasing the overall throughput as seen in Figure 3.58 below.

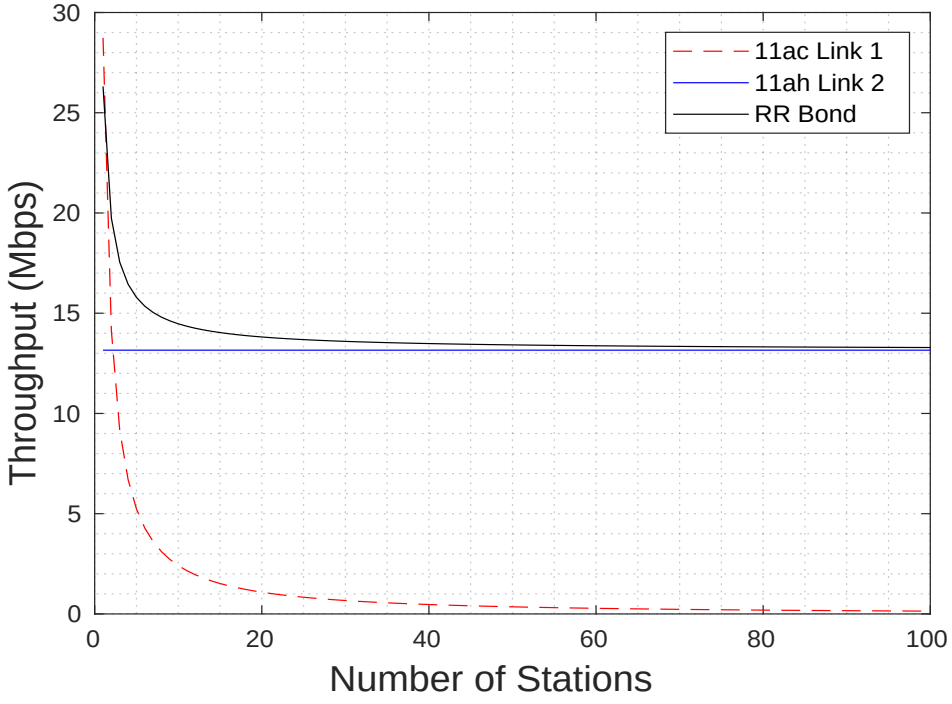

Figure 3.66: Test 2B Static Round Robin Link Throughput Comparison

To demonstrate what happens when condition (1) in Section 3.5.1.2 above is not met Figure 3.67 below illustrates how the bond throughput was influenced by a change in the 11ac PHY data rate. Again for all device pairings evaluated the bond throughput was limited by that of the slower 65 Mbps 11ah device regardless of the capacity of the faster link.

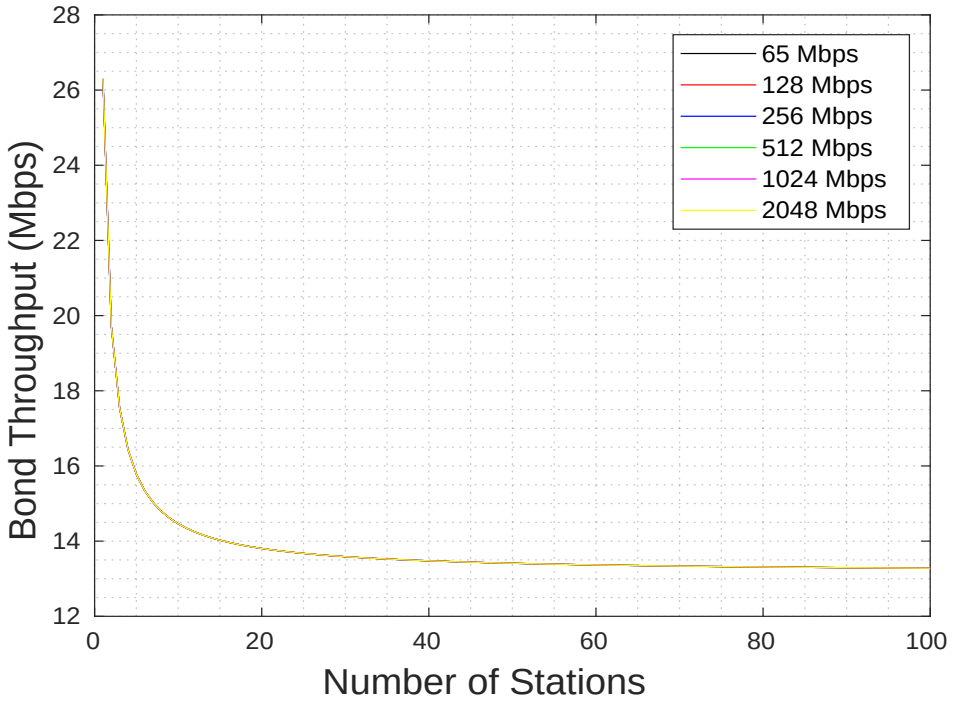

Figure 3.67: Test 2B Static Round Robin Bond Throughput (Heterogeneous Data Rates)

Figure 3.68 shows how the link bandwidth utilisation was impacted by a similar PHY data rate increase. Again all device pairings followed the same exponentially increasing output curve and as expected the higher the PHY rate of the variably loaded 11ac link the lower the resulting bandwidth utilisation.

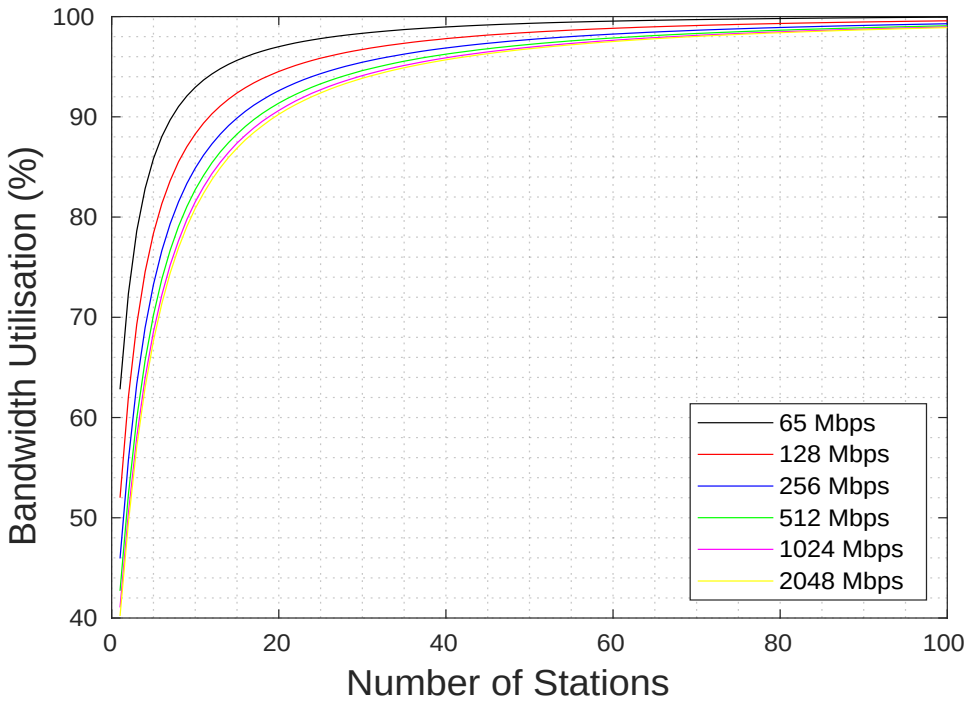

Figure 3.68: Test 2B Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.5.2.5 RR Test 2C: Results

The bond throughput results for the static round robin bond and unbalanced applied load scenario in Test 2C are presented in Figure 3.69 below. The overall mean bond throughput for all applied load combinations used was 0.7406 Mbps. An exponential decreasing curve was seen for both the achieved bond throughput and the maximum theoretical bond throughput. Furthermore there was an apparently larger difference between the two metrics compared to previous tests. At $\{n_{ac} = 1, n_{ah} = 50\}$ there was a significant difference between the achieved bond throughput and the theoretical maximum throughput which were measured at 10.46 Mbps and 28.93 Mbps respectively. As the applied load was increased towards $\{n_{ac} = 100, n_{ah} = 50\}$ the achieved bond throughput was reduced to a minimum of 0.3091 Mbps and the maximum theoretical mean bond throughput to 0.3446 Mbps.

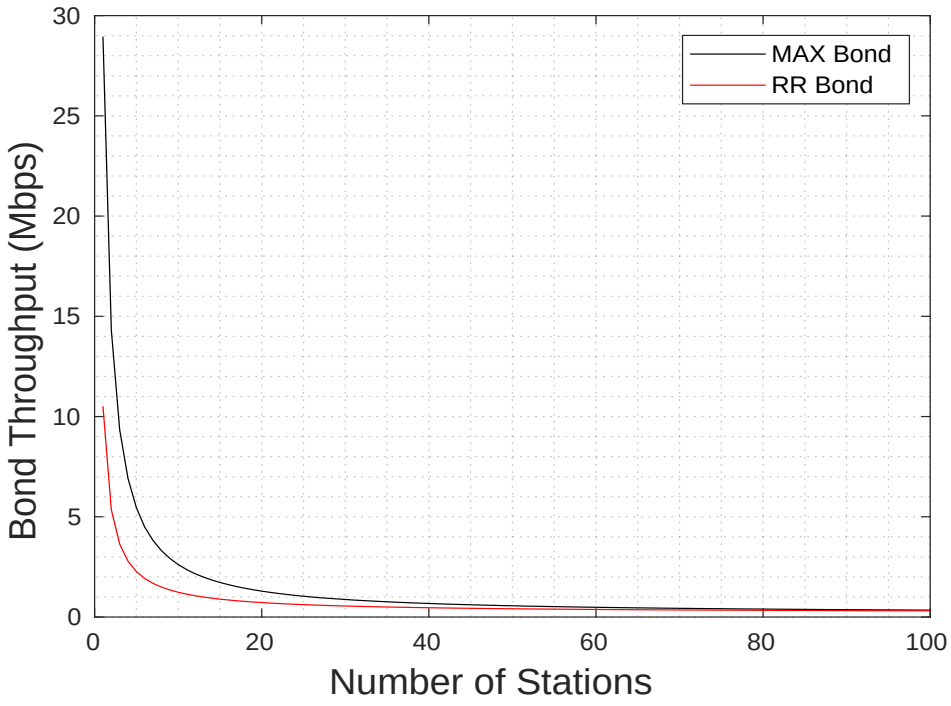

Figure 3.69: Test 2C Static Round Robin Bond Throughput

The link throughput difference results for the unbalanced load scenario in Test 2C are plotted in Figure 3.70 below. The overall mean link throughput difference for all applied loads used was 1.0529 Mbps. At the lightest load of $\{n_{ac} = 1, n_{ah} = 50\}$ the throughput difference was measured at a maximum of 28.52 Mbps. The output then underwent a steep exponential decrease as the load applied was raised. The throughput difference reached a minimum value of 0.001606 Mbps at $\{n_{ac} = 76, n_{ah} = 50\}$ before increasing again to 0.06761 Mbps at the heaviest considered load of $\{n_{ac} = 100, n_{ah} = 50\}$.

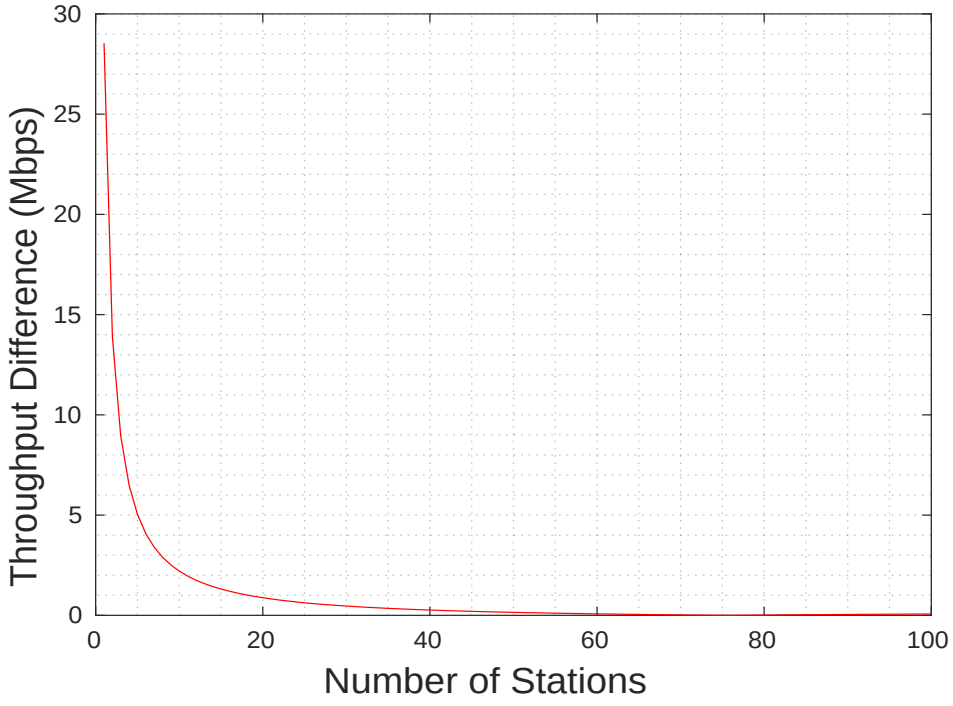

Figure 3.70: Test 2C Static Round Robin Link Throughput Difference

The link bandwidth utilisation results for the static round robin bond in Test 2C are shown in Figure 3.71 below. The overall mean bandwidth utilisation for all applied load combinations used was 70.479%. A non-linear increasing output was observed as the 11ac link load was increased from n = 1 to n = 100. At the initial load of $\{n_{ac} = 1, n_{ah} = 50\}$ the utilisation was measured at a minimum of value 36%. Then as the applied load was gradually increased towards $\{n_{ac} = 100, n_{ah} = 50\}$ the utilisation increased non-linearly to a maximum value of 89.72%.

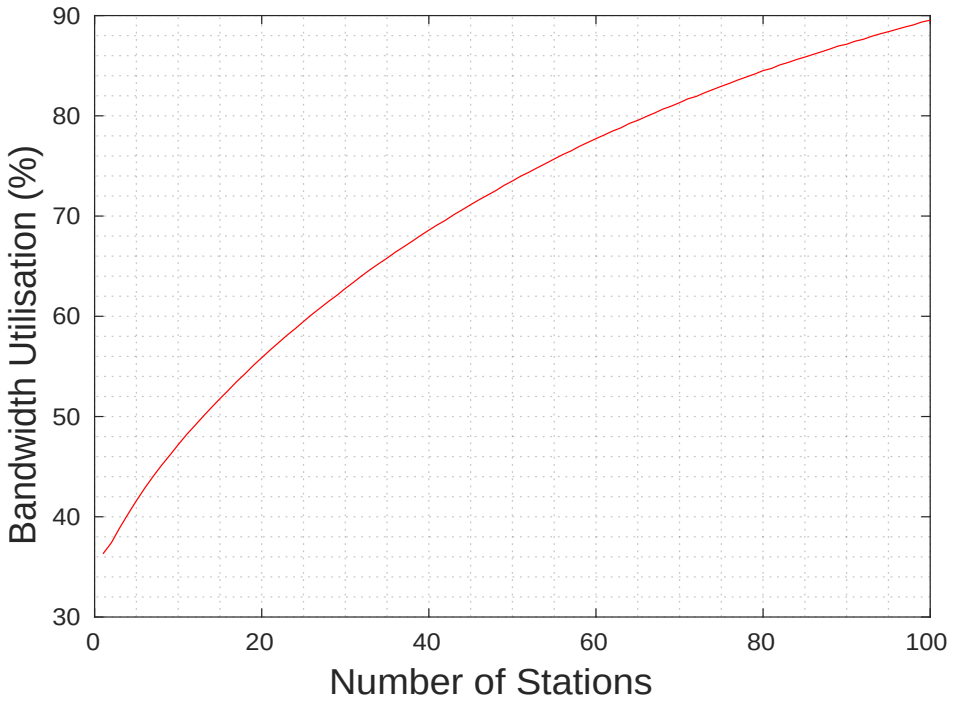

Figure 3.71: Test 2C Static Round Robin Link Bandwidth Utilisation

The bond delay results for the static round robin bond configuration and unbalanced scenario used in Test 2C are shown in Figure 3.72 below. The overall bond delay is plotted as the dotted black link, the 11ac delay as the the solid blue line, and the 11ah link delay as the dashed red line. The mean bond delay for all applied loads considered was 82975.3 μs. For applied loads where $\{n_{ac} =< 75, n_{ah} = 50\}$ the bond delay followed the 11ac delay and was constant at approximately 78070 μs while for $\{n_{ac} =>= 75, n_{ah} = 50\}$ it followed the 11ah delay and increased non-linearly to a maximum value of 117300 μs at the heaviest considered applied load of $\{n_{ac} = 100\, n_{ah} = 50\}$.

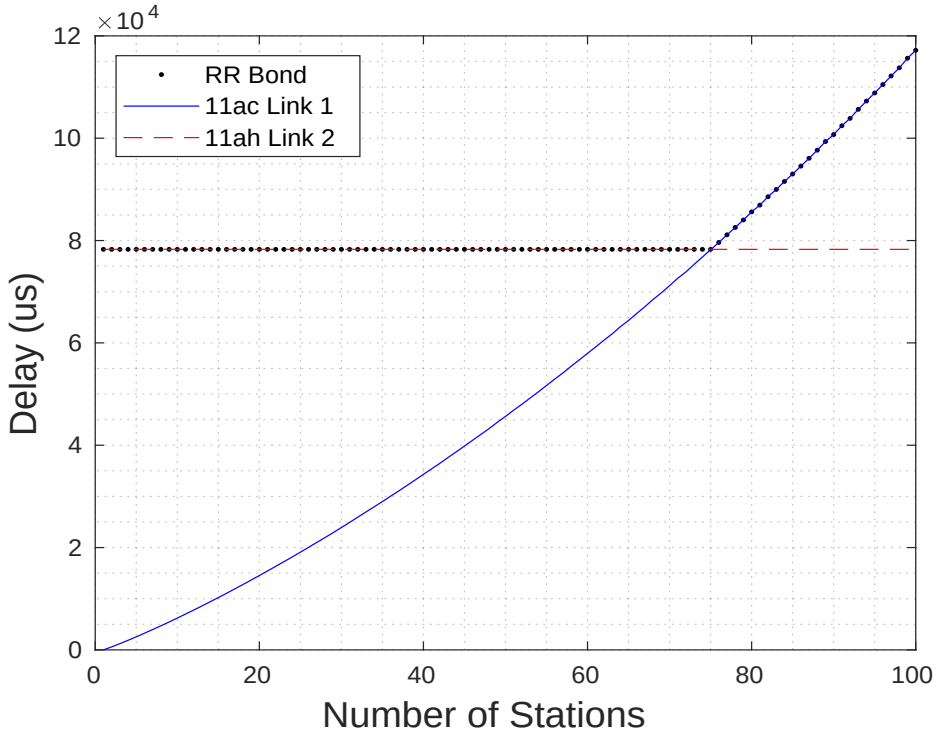

Figure 3.72: Test 2C Static Round Robin Bond Delay

The link delay difference results for the static round robin bond in Test 2C are shown in Figure 3.73 below. The overall mean link delay difference for all applied loads used was 37667.54 μs. At the initial load of $\{n_{ac} = 1, n_{ah} = 50\}$ the delay difference was measured at a maximum test value of 78070 μs. Then as the applied load was increased the simulation output decreased non-linearly to a minimum of 272 μs at $\{n_{ac} = 75, n_{ah} = 50\}$. As the load was raised further still towards $\{n_{ac} = 100, n_{ah} = 50\}$ the link throughput difference increased non-linearly to a final value of 39260 μs.

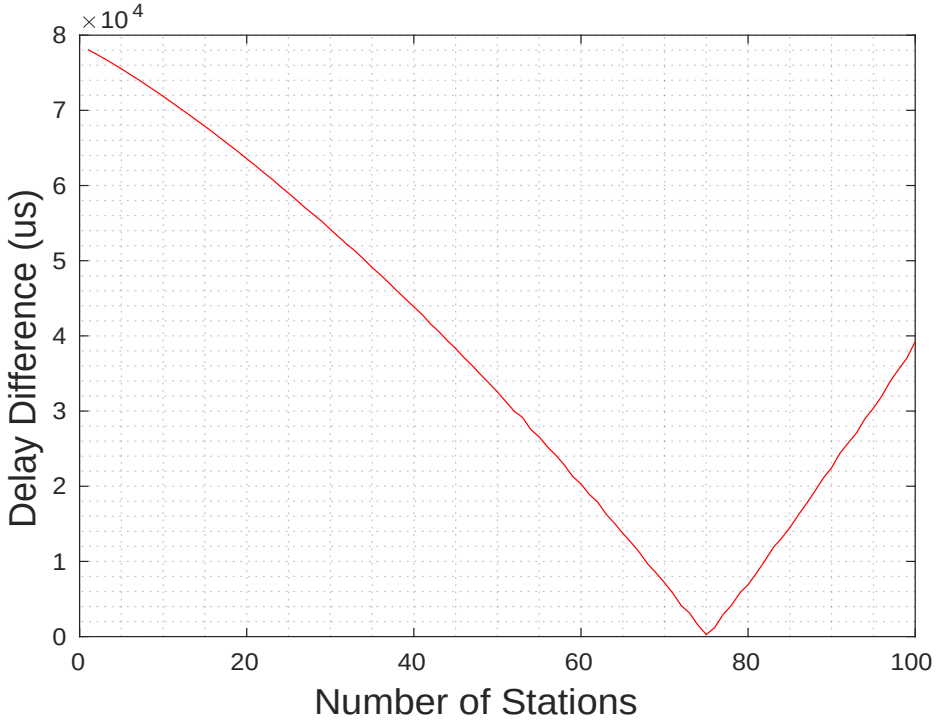

Figure 3.73: Test 2C Static Round Robin Link Delay Difference

3.5.2.6 RR Test 2C: Discussion

The main observation from Test 2C was that the achieved static round robin throughput for the heterogeneous 11ac-11ah bond was 29.5% lower than the theoretical maximum due to the impact of the throughput slow-down effect. The overall mean bond throughput for static round robin was 0.74 Mbps with a mean link bandwidth utilisation of 74.19%.

Test 2C was the second unbalanced load scenario designed to stress the heterogeneous 11ac-11ah bond throughput performance using a moderate static 11ah link load of n = 50. As seen in Figure 3.69 and now expected from the previous tests a non-linear decreasing output was observed for the achieved bond throughput as the applied load was increased. Throughout the test there was a significant difference between the achieved bond throughput and the maximum theoretical throughput caused by the disparity in the achieved performance of each bond link - at the lightest considered load the achieved bond throughput was approximately 10.5 Mbps while the maximum theoretical throughput was 28.9 Mbps.

From visual inspection of the throughput graph it appears that the difference between the achieved bond throughput and theoretical maximum became smaller as the bond load increased, however it is difficult to determine due to the effect of increased resource sharing as the number of nodes was increased. Therefore for we look to the link throughput difference and bandwidth utilisation results to properly quantify the impact of the slow-down effect. As seen from Figure 3.70 above as the load on the 11ac link was increased it reduced the performance disparity but did not remove it altogether. Likewise in Figure 3.71 an increasing link bandwidth utilisation was observed which never quite reached the maximum of 100% and instead stopped at 90% at the heaviest considered test load of $\{n_{11ac} = 100, n_{11ah} = 50\}$. The achieved throughput of the 11ac link eventually exceeds that of the 11ah link at an applied load of $\{n_{11ac} = 74, n_{11ah} = 50\}$. The change is reflected in 3.72 where for applied loads between $\{n_{11ac} = 1, n_{11ah} = 50\}$ and $\{n_{11ac} = 74, n_{11ah} = 50\}$ the bond delay followed the 11ah link delay and remained constant at approximately 78070 μs, while for $\{n_{11ac} >= 75, n_{11ah} = 50\}$ the bond delay followed the 11ac link delay and increased non-linearly. It is also reflected Figure 3.73 where initially the delay difference decreased non-linearly until reaching a minimum of 0 μs at $\{n_{11ac} = 74, n_{11ah} = 74\}$ and then increasing non-linearly for the remainder of the graph.

Another significant finding from Test 2C was that for applied bond loads where $\{n_{11ac} = 1, n_{11ah} = 1\}$ the achieved bond throughput of the static round robin configuration was substantially lower than the single fastest available link, which was in this case the 11ac Link 1 with the variably applied load. In this scenario therefore the use of bonding is counter productive to the general aim of increasing the overall throughput which can be seen from Figure 3.74 below.

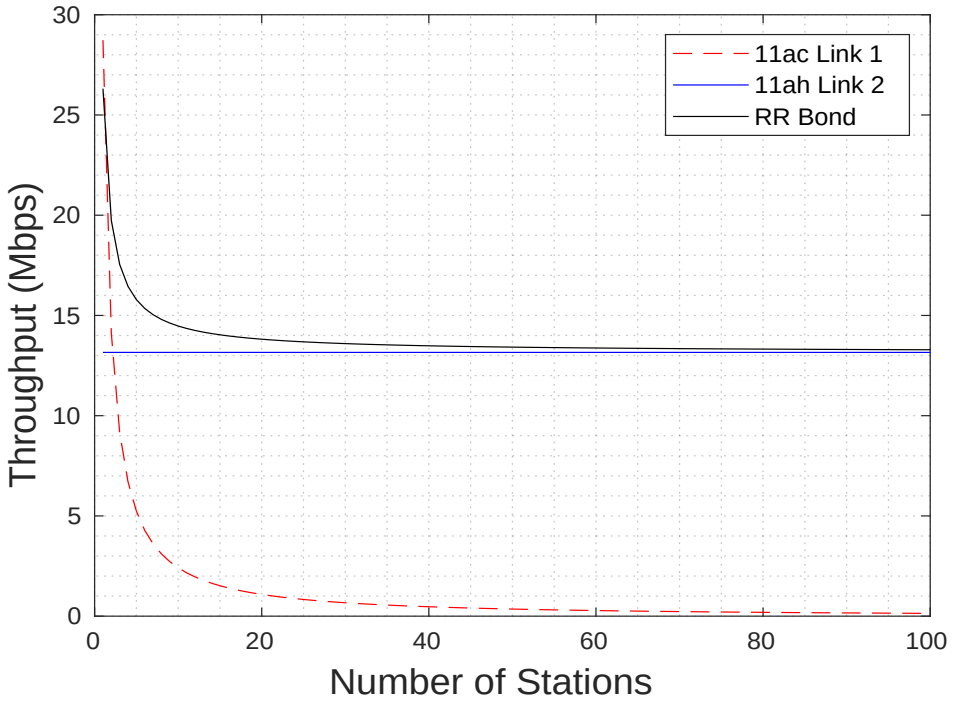

Figure 3.74: Test 2C Static Round Robin Link Throughput Comparison

Figure 3.75 below shows how the compound bond throughput was impacted by an increase in the raw PHY data rate of the 11ac link. Again for all data rate pairings the overall bond throughput was limited by the throughput of the slower 65Mbps 11ah device regardless of the capacity of the higher speed 11ac link.

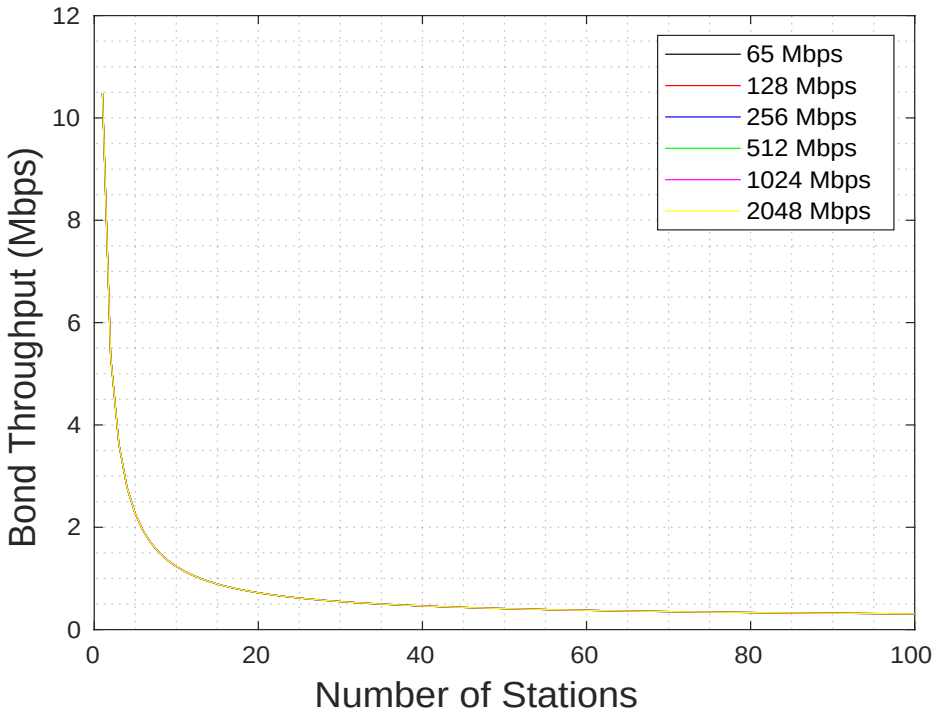

Figure 3.75: Test 2C Static Round Robin Throughput (Heterogeneous Data Rates)

Figure 3.76 shows how the link bandwidth utilisation was influenced by a similar increase in the raw PHY data rate of the 11ac link. Again as per all previous tests all device pairings following a similar exponentially increasing output and as seen from the graph the higher the data rate of the 11ac link the lower the achieved bandwidth utilisation.

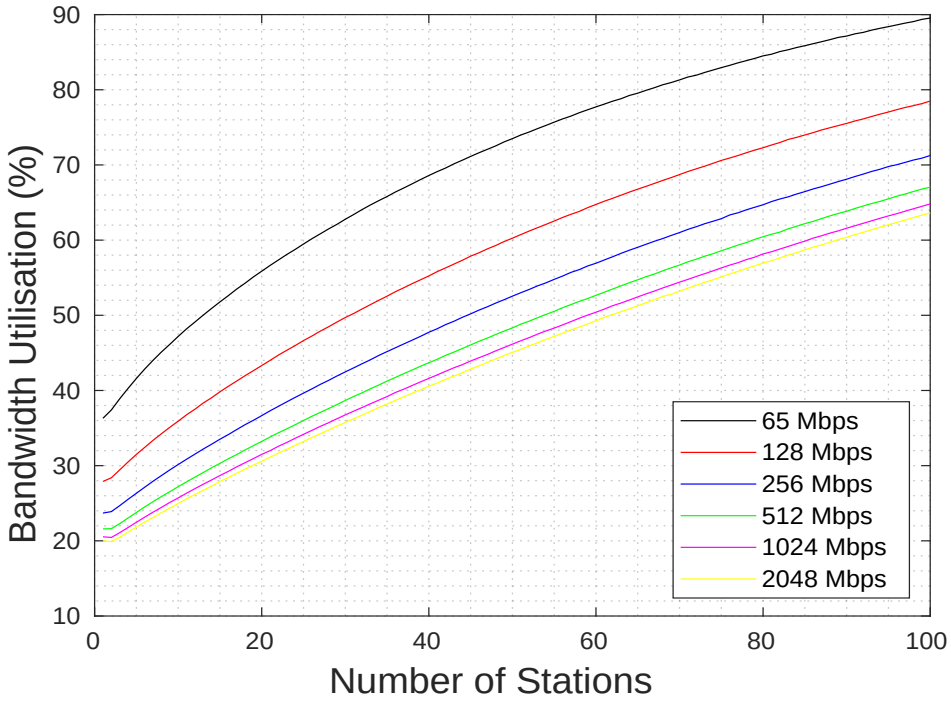

Figure 3.76: Test 2C Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.5.2.7 RR Test 2D: Results

The bond throughput from the static round robin evaluation in Test 2D is plotted in Figure 3.77 as the red line alongside the theoretical maximum bond throughput as the solid black line. The overall mean bond throughout for all applied load combinations considered was 0.5135 Mbps. A familiar exponentially decreasing curve was seen for both throughput metrics as the applied bond load was raised. At $\{n_{ac} = 1, n_{ah} = 100\}$ the maximum bond throughput was 28.82 Mbps while the achieved bond throughput was measured at 8.33 Mbps. Then as the load was gradually increased towards $\{n_{ac} = 100, n_{ah} = 100\}$ the achieved bond throughput was reduced non-linearly to 0.165 Mbps and the maximum theoretical bond throughput to 0.221 Mbps.

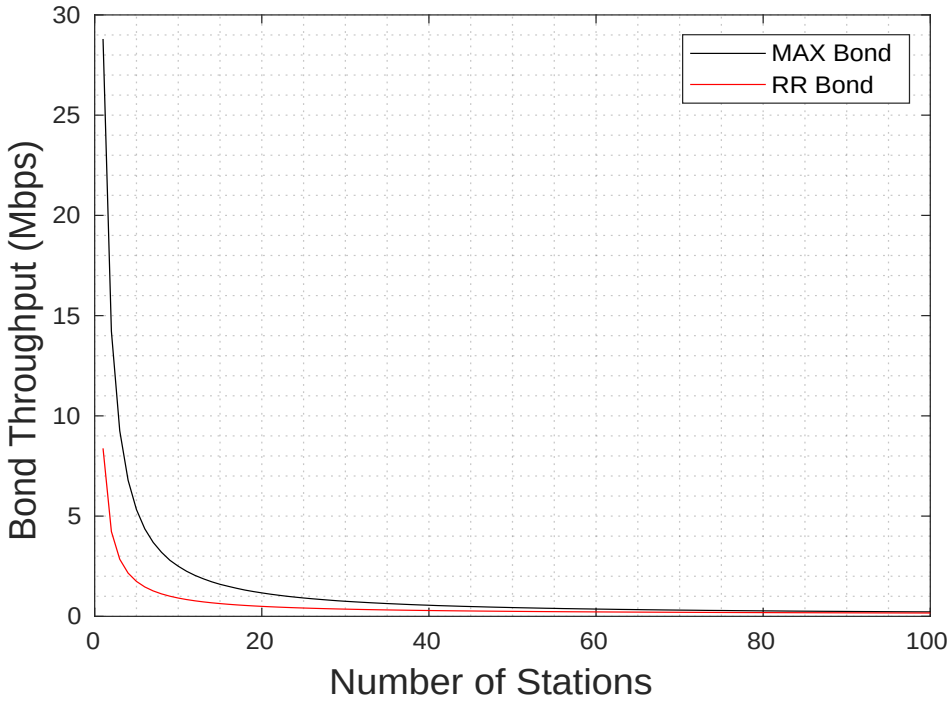

Figure 3.77: Test 2D Static Round Robin Bond Throughput

The link throughput difference results for the static round robin bond in Test 2D are given below in Figure 3.78. The overall mean link throughput difference for all applied load combinations used was 1.1571 Mbps. In the simulation output an exponentially decreasing curve was seen as the applied bond load was increased. At the lightest test load of $\{n_{ac} = 1, n_{ah} = 100\}$ the throughput difference was measured at a maximum value of 28.65 Mbps. Then as the applied load was gradually increased towards $\{n_{ac} = 100, n_{ah} = 100\}$ the throughput difference was reduced to a minimum value of 0.05593 Mbps.

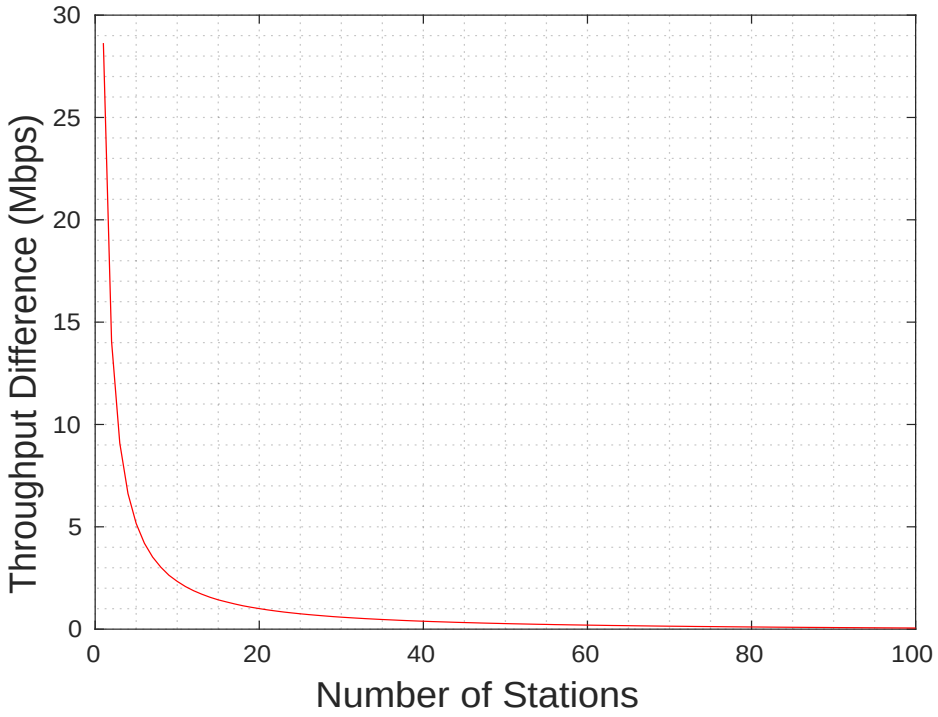

Figure 3.78: Test 2D Static Round Robin Link Throughput Difference Results

The link bandwidth utilisation results for the static round robin configuration in Test 2D are presented in Figure 3.79 below. The overall mean link utilisation for all applied load considered was 74.197%. Again a non-linear increasing curve was seen in the simulation output as the applied bond load increased. At the lightest test load of $\{n_{ac} = 1, n_{ah} = 100\}$ the utilisation was measured at a minimum value of 29.8%. Then as the applied load was raised towards $\{n_{ac} = 100, n_{ah} = 100\}$ the achieved utilisation increased non-linearly to a maximum value of 74.6%.

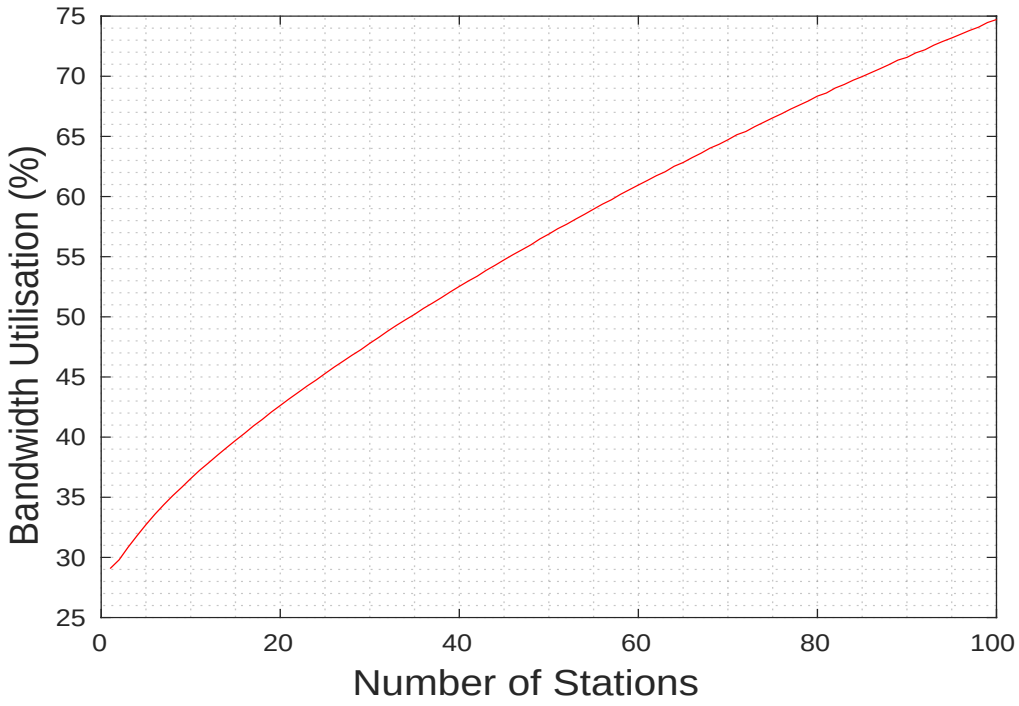

Figure 3.79: Test 2D Static Round Robin Link Bandwidth Utilisation

The bond delay results for the static round robin configuration in Test 2D are presented in Figure 3.80 below. The overall bond delay is plotted as the dotted black line, the 11ac link delay as the solid blue line, and the 11ah link delay as the dashed red line. For the entire range of applied loads considered from $\{n_{ac} = 1, n_{ah} = 100\}$ to $\{n_{ac} = 100, n_{ah} = 100\}$ the bond delay remained constant at 190200 μs following that of the slower 11ah device.

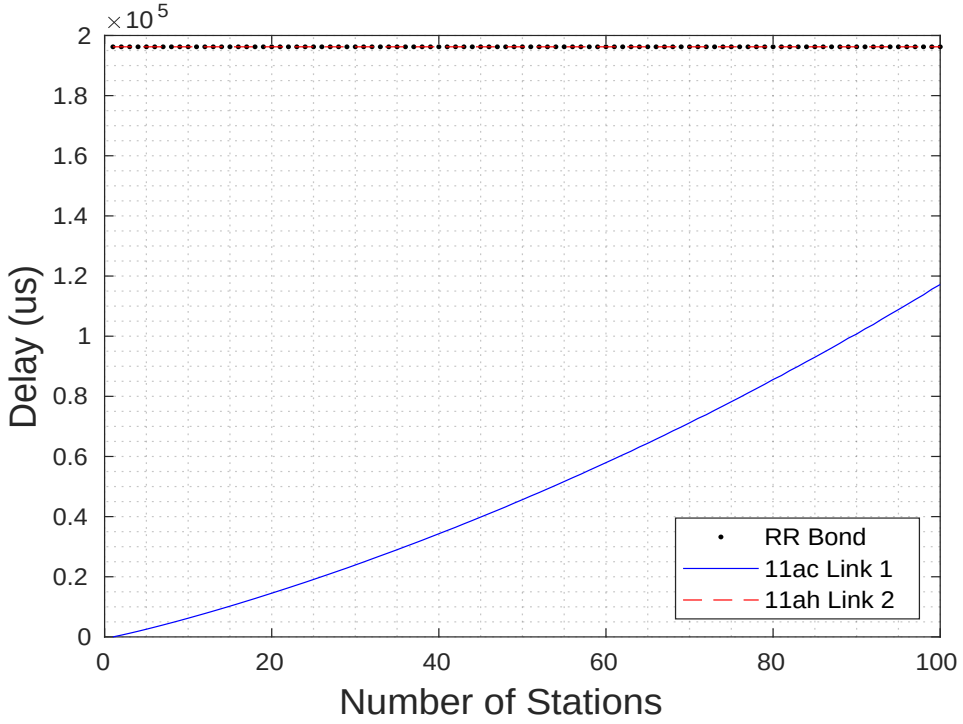

Figure 3.80: Test 2D Static Round Robin Bond Delay

The delay difference results for the static round robin and unbalanced load scenario in Test 2D are shown in Figure 3.81. The overall mean delay difference for all applied loads considered was 145166.49 μs. As seen below a non-linear decreasing curve was seen in the simulation output as the applied bond load was increased. At $\{n_{ac} = 1, n_{ah} = 100\}$ the delay difference was measured at a maximum of 190200 μs and as the applied load was gradually increased towards $\{n_{ac} = 100, n_{ah} = 100\}$ the delay difference decreased non-linearly to a minimum value of 78450 μs.

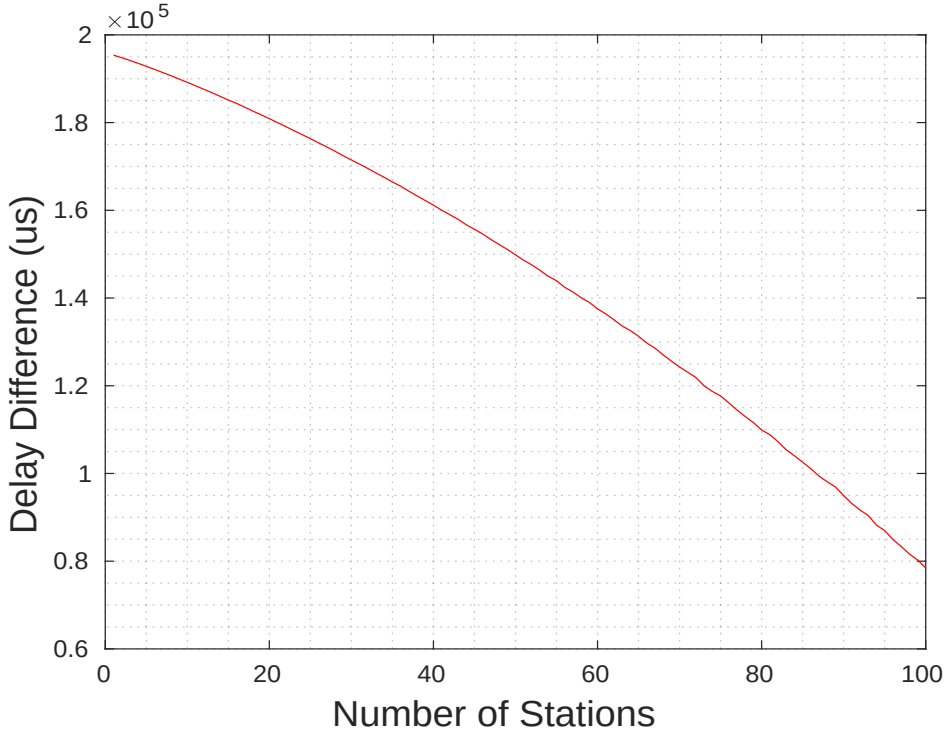

Figure 3.81: Test 2D Static Round Robin Link Delay Difference

3.5.2.8 RR Test 2D: Discussion

The main finding from Test 2D was that mean bond throughput achieved by static round robin was 25.8% lower than the theoretical maximum mean throughput due to the severe impact of the slow-down effect. The mean bond throughput for the test was 0.513 Mbps which was the lowest for the heterogeneous 11ac-11ah bond overall in Tests 2A-2D.

The applied loads in Test 2D were chosen to stress the bond throughput performance using a heavy 11ah load of n = 100 competing nodes. At the lightest load of $\{n_{11ac} = 1, n_{11ah} = 100\}$ the achieved bond throughput was just over 8 Mbps while the maximum throughput was much higher at 29 Mbps. As the load on the 11ac link was raised its achievable throughput reduced due to increased resource sharing on the DCF channel and the response of the CSMA/CA back-off algorithm. As seen in Figure 3.78 this reduced the throughput disparity and as a result the impact of the bond slow-down effect was gradually reduced as seen in Figure 3.79, where the link bandwidth utilisation increased non-linearly as the applied load was raised towards $\{n_{11ac} = 1, n_{11ah} = 100\}$.

The same behaviour was also reflected in the bond delay in Figure 3.80 which followed that of the slower 11ah link and was constant throughout at approximately 190200 μs, as well as the delay difference in 3.81 which decreases non-linearly as the applied load is increased.

Another important finding from the test was that for applied loads where $\{n_{11ac} = 1, n_{11ah} = 1\}$ the achieved throughput of the static round robin bond was lower than the single fastest available link, which was in this case 11ac Link 1 with the variably applied load. Here the use of bonding is again counter productive to the aim of increasing the throughput as seen from the graph in Figure 3.82 below.

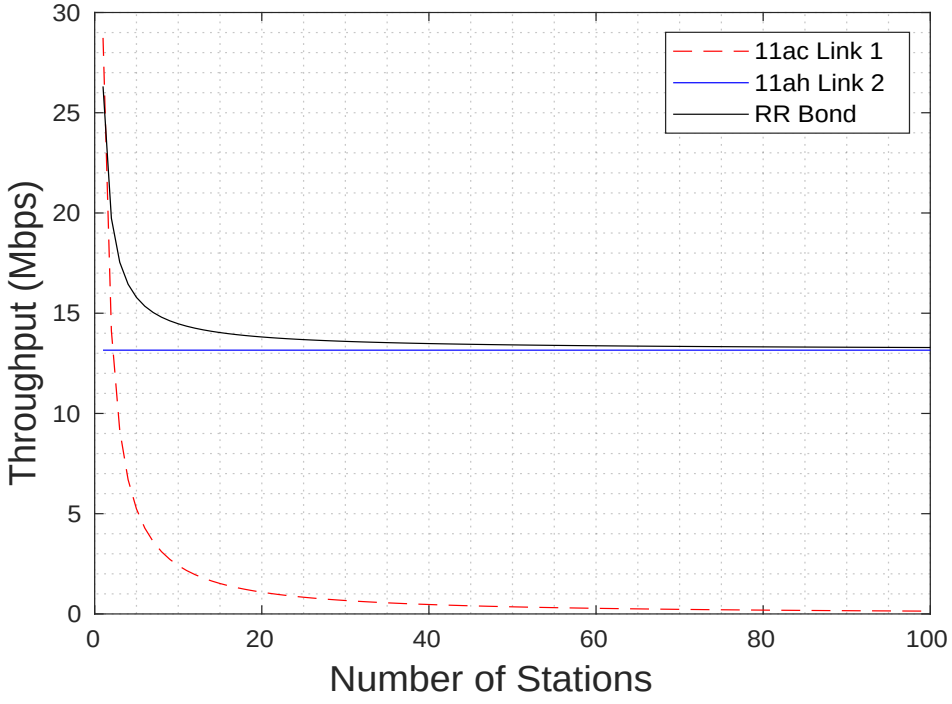

Figure 3.82: Test 2D Static Round Robin Link Throughput Comparison

Figure 3.83 below illustrates how the aggregate bond throughput was influenced by an increase in the 11ac link PHY data rate. As seen throughout all tests and now anticipated, for all data rate pairings the overall bond throughput was *always* limited by the throughput of the slower 65Mbps 11ah device regardless of the capacity of the higher speed 11ac link.

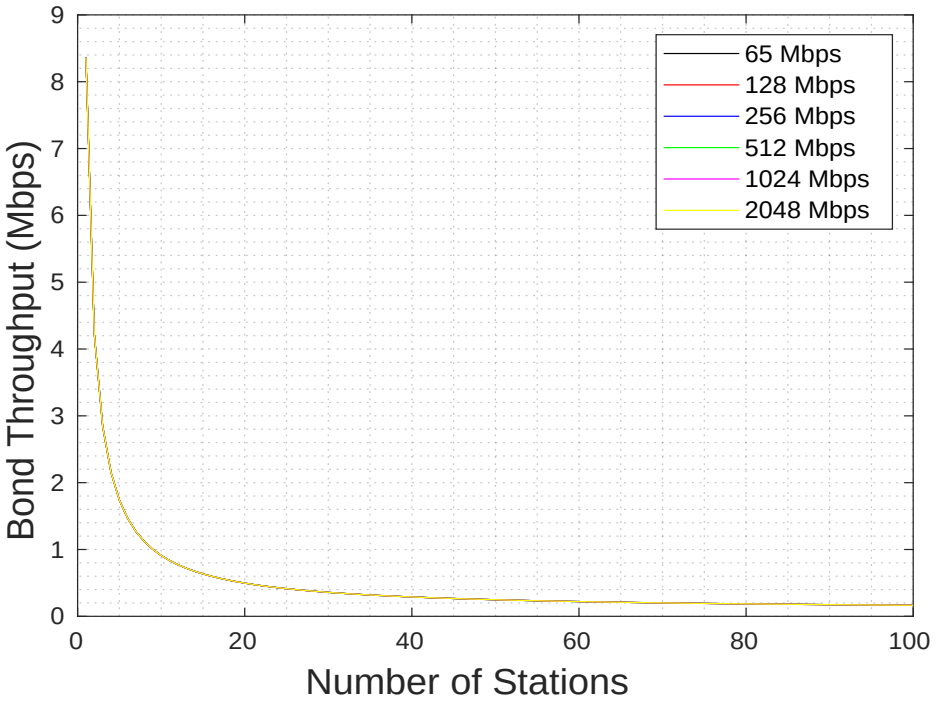

Figure 3.83: Test 2D Static Round Robin Bond Throughput (Heterogeneous Data Rates)

Similarly graph in Figure 3.84 shows how the link bandwidth utilisation was influenced by the same increase in the PHY data rate of the variably loaded first 11ac link. Again as per the previous tests all device pairings followed the same exponentially increasing output and in general the higher the data rate of the 11ac link the lower the resulting link bandwidth utilisation.

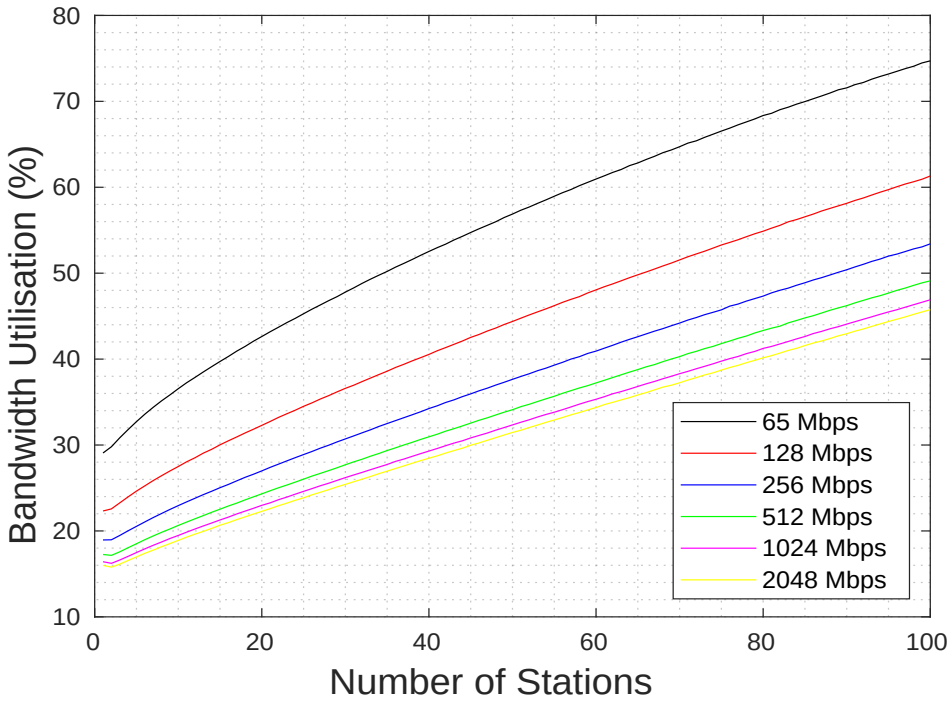

Figure 3.84: Test 2D Static Round Robin Bandwidth Utilisation (Heterogeneous Data Rates)

3.6 Results Evaluation

This section provide a detailed evaluation and comparison of the results from the static round robin testing presented in the previous section. The results for the homogeneous 11ac-11ac bond are evaluated in Section 3.6.1 and the heterogeneous 11ac-11ah bond results are evaluated in Section 3.6.2. The performance of the homogeneous and heterogeneous 802.11 bond types are compared later in Section 3.6.3.

3.6.1 Homogeneous 11ac-11ac Bond

The throughput results for the homogeneous 11ac-11ac bond in Tests 1A-1D demonstrate the inefficiency of the traditional static approach to link selection where there is a disparity in the individual link performance as would be the case in most practical scenarios.

The achieved mean 11ac-11ac bond throughput for all four tests was 6.085 Mbps with a mean link bandwidth utilisation of 84.62%. This meant that during the test over 15% of the total available link bandwidth was wasted due to the throughput slow-down effect and the inability to adapt dynamically to changes in link state.

	Test 1A	**Test 1B**	**Test 1C**	**Test 1D**
Bond T.put (Mbps)	2.4799 ±0.1038	19.7604 ±0.88	1.2422 ±0.04	0.85805 ±0.09
T.put Diff. (Mbps)	0.0018 ±5 * 10^{-6}	2.449 ±0.07	1.0223 ±0.013	1.101 ±0.0133
B.width Util. (%)	99.902 ±0.0082	65.03 ±0.0092	91.94 ±0.0085	81.61 ±0.0089
Bond Delay (us)	50264 ±610	50266.553 ±1588	62596 ±674	117134 ±1274
Delay Diff. (us)	95.3797 ±28	50266.551 ±1533	22425 ±348	66917 ±682

Table 3.3: Summary of Static Round Robin Results for Homogeneous 11ac-11ac Bond

The mean simulation results are summarised in Table 3.3 along with the 99.9% confidence interval. The general output behaviour in each case is summarised in Table 3.4 on the opposite page. The results for each individual test scenario are presented in detail in the previous chapter section.

For the current study the main performance metric of interest is the link bandwidth utilisation which provides a measure of the overall bond efficiency and the main method of quantifying the impact of the throughput slow-down effect which is used throughout the thesis research.

Looking at the bond throughput alone can be misleading, for example the highest throughput performance for round robin is seen in Test 1B but this was the worse overall in terms of the link bandwidth utilisation with a mean of just 65.03%. The best utilisation was seen in the benchmark case in Test 1A with 99.902% but this had only the second highest mean bond throughput of 2.4799 Mbps. Here the achieved throughputs of both links reduced exponentially as the applied loads were increased simultaneously and so the overall mean bond throughput was lower compared to the unbalanced scenarios in Tests 1B-1C. In the unbalanced tests the achieved

mean bond throughput was mostly related to the throughput of the statically loaded second 11ac link. In Test 1B the applied load on the second 11ac link was a just single node therefore the overall achieved mean bond throughput was relatively high at 19.76 Mbps. By contrast in Test 1D the static load on the second link was one hundred nodes achieving a mean bond throughput of 0.85805 Mbps.

Likewise the mean bond delay in the unbalanced load scenarios in Tests 1B-1D increased according to the size of the static load applied in each particular case. Note that link throughput and link delay difference metrics are absolute and do not consider the relative proportional size difference.

	Test 1A	**Test 1B**	**Test 1C**	**Test 1D**
Bond Throughput	non-linear, decreasing throughout	non-linear, decreasing throughout	non-linear, decreasing throughout	non-linear, decreasing throughout
Throughput Difference	flat output, < 0.1 kbps throughout	non-linear, increasing throughout	decreasing for $\{n_1 < 0\}$ increasing for $\{n_1 > 50\}$	non-linear, decreasing throughout
Throughput Efficiency	flat output 100% or near throughout	non-linear, decreasing throughout	increasing for $\{n_1 < 50\}$ decreasing for $\{n_1 > 50\}$	non-linear, increasing throughout
Bond Delay	non-linear, increasing, link 1 and 2 delay equal	non-linear, increasing, follows Link 1	constant for $\{n_1 < 50\}$ increasing for $\{n_1 > 50\}$	flat output follows Link 2
Delay Difference	mostly flat output and < 500 μs throughout	non-linear, increasing throughout	decreasing for $\{n_1 < 50\}$ increasing for $\{n_1 > 50\}$	non-linear, decreasing throughout

Table 3.4: Static Round Robin Output Behaviour for Homogeneous 11ac-11ac Bond

Identified by comparing the results of Tests 1B-2D against the benchmark case in Test 1A there are three conditions (within the constraints of the implemented framework) that must be satisfied in order to obtain a fully balanced 802.11 interface bond: (1) Homogeneous PHY Data Rates, (2) Identical DCF Parameters, and (3) Balanced Applied Load. When all three requirements are met the aggregate throughput of the statically configured round robin bond is maximised and no slow-down effect occurs, i.e. the achieved bond throughput is equal to the sum of the throughputs of the individual links as was the case throughout in Test 1A.

As shown by the further analysis for Test 1A and the results of Tests 1B-1D, when any of the

above conditions are not satisfied a non-negligible throughput slow-down effect is observed. In order to quantify the impact of the effect in each case the key results from the evaluation of the static bond configuration are summarised as follows:

- Test 1A: In the benchmark test the achieved static round robin bond throughput was within 0.01% of the theoretical maximum mean bond throughput of 2.4803 Mbps. The near-optimal performance was due to the homogeneous 11ac-11ac and 65Mbps-65Mbps bond and balanced applied load used, i.e. all three of the above conditions were satisfied such that no or very little throughput slow-down was observed due to the in-order delivery requirement.

- Test 1B: In the first unbalanced test only conditions (1) and (2) above were met and the mean 11ac-11ac bond throughput under static round robin was 19.76 Mbps with a mean link bandwidth utilisation of 65.06% such that over one third of the available capacity was wasted due to the throughput slow-down effect.

- Test 1C: In the second unbalanced load test again only conditions (1) and (2) were met but the overall performance was better compared to Test 1B. The mean static round robin 11ac-11ac bond throughput was 1.2422 Mbps with a link bandwidth utilisation of 99.19% such that less than 1% of the total available link capacity was wasted due to the throughput lag effect.

- Test 1D: In the final test for the homogeneous 11ac-11ac test only conditions (1) and (2) were satisfied and the mean round robin bond throughput was 0.858 Mbps with a mean throughput efficiency of 81.61% such that over 18% of the total available link capacity was wasted due to the throughput slow-down effect.

Another significant finding in the context of the main thesis research was that on numerous occasions the impact of the inherent slow-down effect was so severe that the achieved throughput of the statically configured round robin bond was reduced below that of the single fastest available link, therefore rendering the use of interface bonding counter productive to the aim of increasing the overall link speed. For the homogeneous 11ac-11ac interface bond considered in Tests 1A to 1D this particularly damaging scenario occurred at the applied loads shown in Table 3.5 below.

Load Test	**Problematic Loads**
RR Test 1A	Not Applicable
RR Test 1B	$\{n_1 >= 8, n_2 = 100\}$
RR Test 1C	$\{n_1 <= 14, n_2 = 100\}$
RR Test 1D	$\{n_1 <= 38, n_2 = 100\}$

Table 3.5: Problematic Applied Loads for Homogeneous 11ac-11ac Bond

3.6.2 Heterogeneous 11ac-11ac Bond

The round robin throughput results for Tests 2A-2D also demonstrated the inefficiency of the traditional static approach to link selection when there is an unbalanced load applied to a *heterogeneous* 11ac-11ah interface bond. The main finding was that in all four tests the achieved mean bond throughput was 4.1 Mbps with an overall mean link bandwidth utilisation of 78.75% such that over 20% of the entire available link bandwidth was wasted due to the throughput slow-down effect and the inability of the statically configured bond to adapt dynamically to changes in link state.

	Test 2A	**Test 2B**	**Test 2C**	**Test 2D**
Bond T.put (Mbps)	1.2999 ±0.0174	13.8537 ±0.8	0.7406 ±0.026	0.5135 ±0.021
T.put Diff. (Mbps)	0.59016 ±0.0064	12.2602 ±0.648	1.0529 ±0.015	1.1571 ±0.019
B.width Util. (%)	72.98 ±0.0079	95.363 ±0.0082	70.479 ±0.01	74.197 ±0.012
Bond Delay (us)	85313 ±1598	50197.75 ±1562	82975 ±1642	195383 ±2221
Delay Diff. (us)	35113 ±592	50197.74 ±1541	37667 ±554	145166 ±1891

Table 3.6: Summary of Static Round Robin Results for Heterogeneous 11ac-11ah Bond

The mean bond throughput, delay, and utilisation results statistics are summarised in Table 3.6 above along with the 99.9% confidence interval, which allows us to say that for 99.9% of simulations run, the true mean is likely to fall within the specified range. The output behaviour in each case is summarised in Table 3.7 below. The results for each individual applied load tests are presented in the previous chapter section. A summary of the main findings from the evaluation is given in Section 3.6.2 on the following page.

	Test 2A	**Test 2B**	**Test 2C**	**Test 2D**
Bond Throughput	non-linear, decreasing throughout	non-linear, decreasing throughout	non-linear, decreasing throughout	non-linear, decreasing throughout
Throughput Difference	non-linear, decreasing throughout	non-linear, decreasing throughout	decreasing for $\{n_{11ac} < 0\}$ increasing for $\{n_{11ac} > 50\}$	non-linear, decreasing throughout
Throughput Efficiency	non-linear, increasing throughout	non-linear, increasing throughout	increasing for $\{n_{11ac} < 50,\}$ decreasing for $\{n_{11ac} > 50\}$	non-linear, increasing throughout
Bond Delay	non-linear, increasing, follows 11ah	non-linear, increasing, follows 11ac	constant for $\{n_{11ac} < 75\}$ increasing for $\{n_{11ac} > 75\}$	flat output throughout, follows 11ac
Delay Difference	non-linear, increasing throughout	non-linear, increasing throughout	decreasing for $\{n_{11ac} < 75\}$ increasing for $\{n_{11ac} > 75\}$	non-linear, decreasing throughout

Table 3.7: Heterogeneous 11ac-11ac Bond Output Behaviour

To quantify the throughput slow-down effect for the heterogeneous 11ac-11ah bond using the static round robin configuration the key results from the evaluation are summarised below:

- RR Test 2A: In the balanced applied load scenario for the heterogeneous 11ac-11ah bond the static round robin configuration achieved an overall mean throughput of 1.299 Mbps with a mean link bandwidth utilisation of 72.91%. Despite the balanced applied load and identical 65Mbps-65Mbps data rates used the heterogeneous DCF parameters between the two 802.11 types resulted in a significant performance disparity and an overall throughput wastage of over 27%.

- RR Test 2B: In the first of the unbalanced applied load scenarios for the heterogeneous 11ac-11ah bond, the static round robin configuration achieved an overall mean bond throughput of 13.85 Mbps with a mean link bandwidth utilisation of 97.36% such that 2.5% of the total available capacity was wasted due to the throughput slow-down and the inability of the interface bond to adapt dynamically to changes in the link state.

- RR Test 2C: In the second unbalanced load test for the heterogeneous 11ac-11ah bond the achieved mean bond throughput under round robin was 0.7406 Mbps with a mean link bandwidth utilisation of 70.05%. This was such that nearly two fifths of the available link capacity was wasted due to the throughput slow-down effect and the static link configuration used by the bond.

- RR Test 2D: In the third and final unbalanced load scenario for the homogeneous 11ac-11ah bond the overall mean throughput for static round robin was 0.513 Mbps with a mean link bandwidth utilisation of 74.19% such that just over one quarter of the total available link capacity was wasted due to the throughput slow-down effect and static link configuration.

Again on multiple occasions the impact of the inherent slow-down effect was such that the achieved aggregate throughput of the static round robin bond was reduced below that of the single fastest available link, therefore rendering the use of interface bonding counter productive to the goal of increasing link speed. For the heterogeneous 11ac-11ah interface bond used in Tests 2A to 2D this damaging event occurred occurred at the applied loads shown in Table 3.8 below.

Load Test	**Problematic Loads**
RR Test 2A	$\{n_{11ac} = 1, n_{11ah} = 1\}$
RR Test 2B	$\{n_{11ac} = 1, n_{11ah} = 1\}$
RR Test 2C	$\{n_{11ac} <= 40, n_{11ah} = 100\}$
RR Test 2D	$\{n_{11ac} <= 82, n_{11ah} = 100\}$

Table 3.8: Problematic Applied Loads for Heterogeneous 11ac-11ah Bond

3.6.3 11ac-11ac vs 11ac-11ah Bond Performance

This section analyses and compares the performance of the homogeneous 11ac-11ac bond and the heterogeneous 11ac-11ah bond in each of the individual applied load scenarios considered for the the static round robin configuration.

3.6.3.1 Balanced Load Scenarios in Tests 1A and 2A

In Tests 1A and 2A the two bonds were subjected to an identical balanced load where the number of competing stations in each individual 802.11 network was incremented simultaneously to allow a performance comparison at similar levels of applied load. The achieved static round robin throughput for both bonds are plotted on the same axis in Figure 3.85 below.

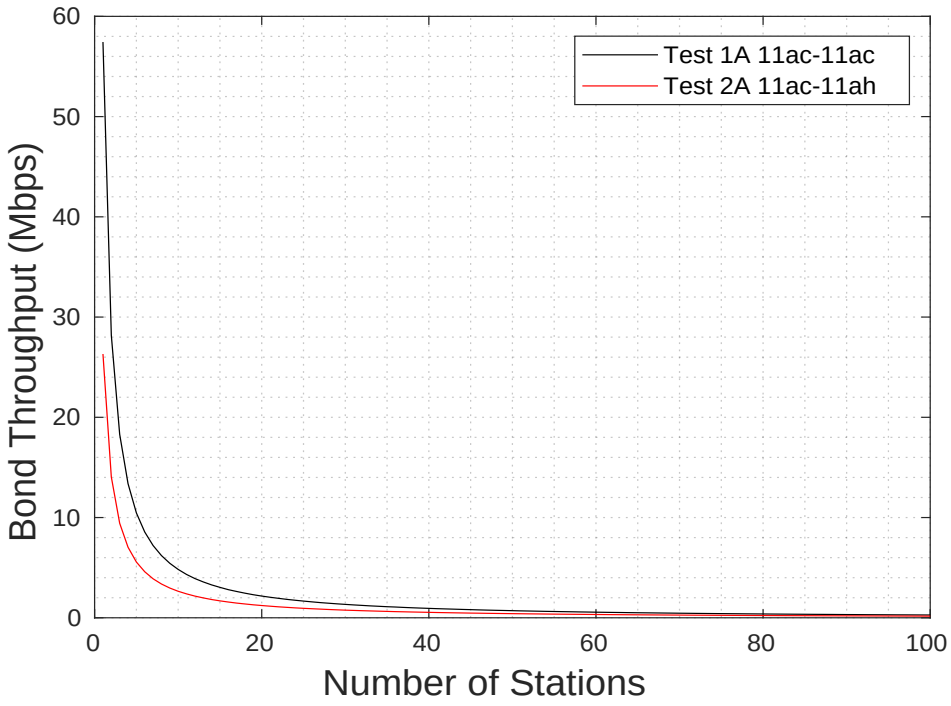

Figure 3.85: Test 1A vs 2A Round Robin Throughput

The main finding was that the mean link bandwidth utilisation for the homogeneous bond was substantially higher at 99.99% compared to just 72.81% for the heterogeneous bond despite the balanced load and homogeneous 65Mbps-65Mbps raw PHY date rates used. The overall achieved mean bond throughput was also higher for the homogeneous 11ac-11ac in Test 1A with 2.4799 Mbps while for the heterogeneous 11ac-11ah bond in Test 2A this was just 1.299 Mbps.

Test 1A gave the ideal conditions for the homogeneous 11ac-11ac bond throughput because as discussed in Section 3.5.1.2 the bond was fully balanced: (1) both 802.11 links used the same

PHY data rate of 65 Mbps, (2) both bond links used identical DCF parameters, and (3) both links always had the same applied load. This gave an identical throughput and delay response as the applied load was raised. For the homogeneous case throughout the entire test neither 11ac slave was forced to wait for the other to finish sending because on average they completed their transmissions at approximately the same time. Therefore no throughput slow-down effect was observed and the static round robin bond throughput efficiency was 100% or near for all applied loads considered.

By contrast in Test 2A the same behaviour was not replicated for the 11ac-11ah bond under the same balanced load scenario where a significant throughput slow-down effect was observed which occurred due the difference in DCF parameter values between the 11ac and 11ah devices and in particular the inter-frame spacing and slot length. This meant that conditions (1) and (3) above for a fully balanced bond are both satisfied but condition (2) was not met which led to a difference in the achieved access delays between the 11ac and 11ah links at similar applied loads. As seen from the delay difference plot in Figure 3.86 below at the initial load of $\{n_{11ac} = n_{11ah} = 1\}$ the 11ah link delay was approximately 15680 μs higher compared to the 11ac link. As the applied load was increased simultaneously the delay disparity became larger eventually reaching 75520 μs at the heaviest considered load of $\{n_{11ac} = n_{11ah} = 100\}$. By contrast the delay difference for the 11ac-11ac bond was negligible at all applied loads considered.

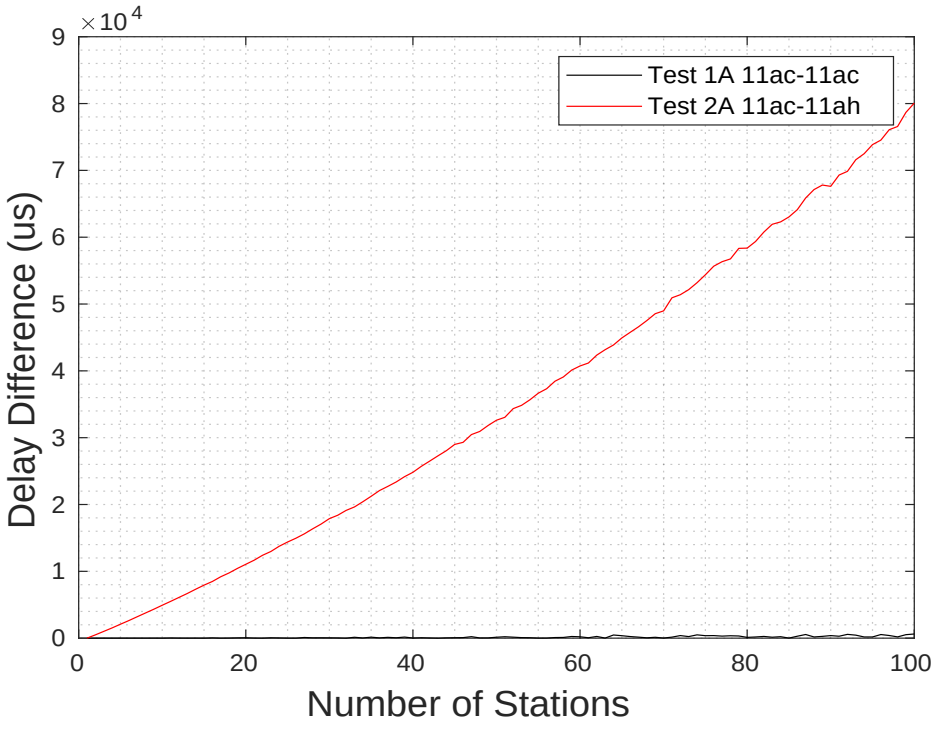

Figure 3.86: Test 1A vs 2A Link Delay Difference

The high delay of the 11ah link had the knock effect of reducing the achieved throughput in comparison to the 11ac device. As seen in Figure 3.87 below although both devices had the same 65 Mbps data rate, at an applied load of $\{n_{11ac} = n_{11ah} = 1\}$ the 11ah link throughput was almost 16 Mbps slower due to the difference in DCF parameters. However as the applied load was raised on both links simultaneously the throughput of the 11ac link decreased at a faster rate which brought it closer in performance to that of the 11ah link thereby cancelling out the impact of the increasing delay difference.

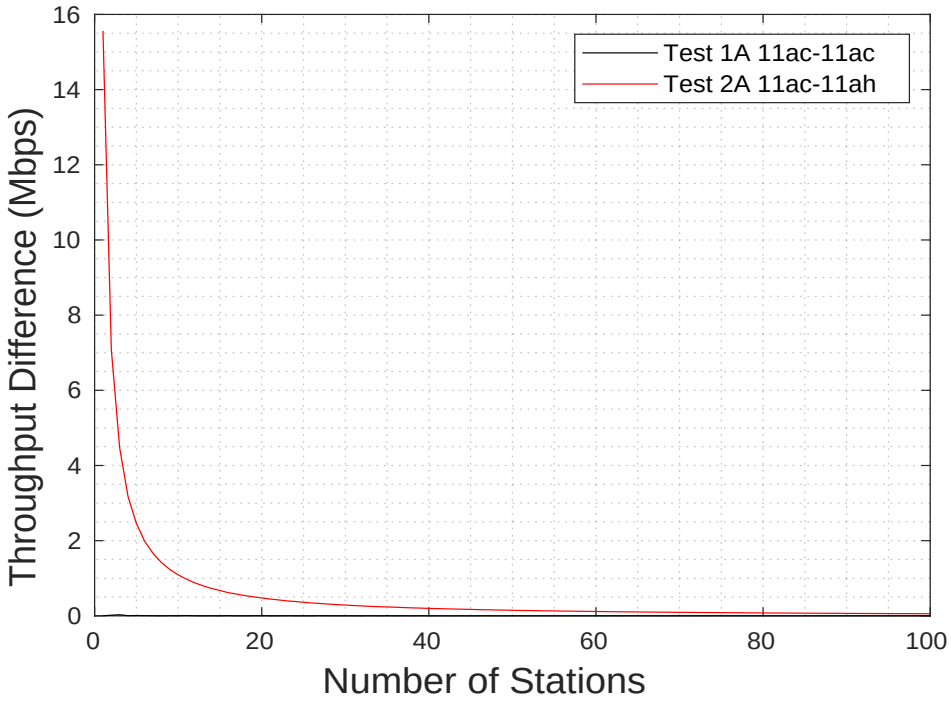

Figure 3.87: Test 1A vs 2A Round Robin Throughput Difference

This same behaviour was also reflected in the link bandwidth utilisation for the homogeneous and heterogeneous bonds plotted along the same axis in Figure 3.88 below. As observed the 11ac-11ac bond bandwidth utilisation was equal to 100% or near for all applied load combinations considered. By contrast for the 11ac-11ah bond as the applied load was raised the utilisation increased non-linearly from an minimum value of 63% to a maximum of 74% at the highest considered load despite the delay difference increasing over the same range. The overall mean link bandwidth utilisation for the heterogeneous 11ac-11ah bond was 72.9% compared to 99.99% for the homogeneous bond.

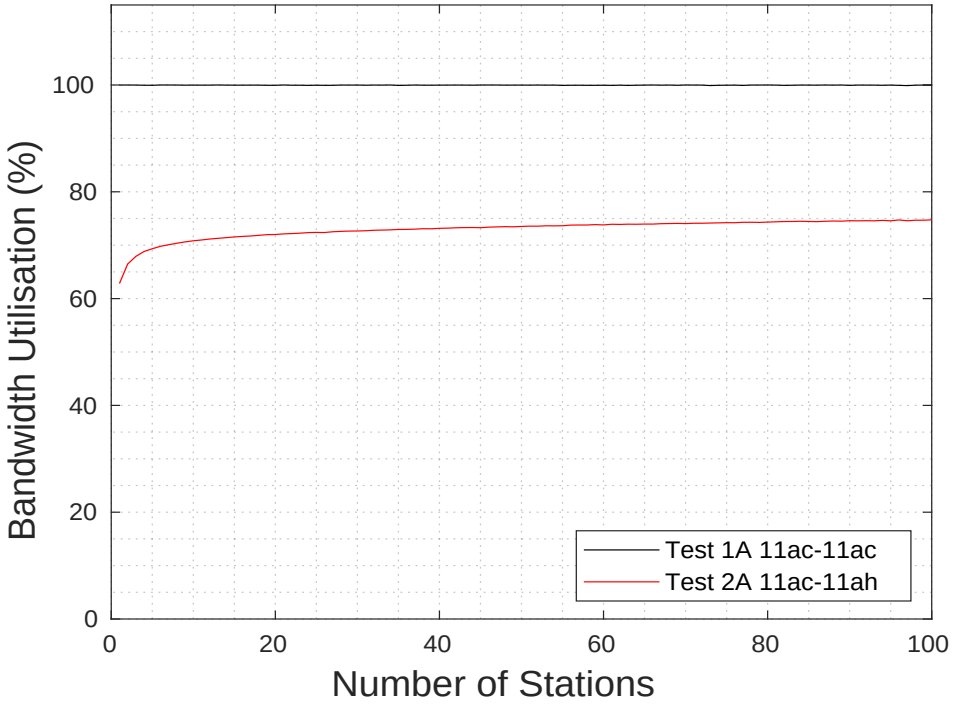

Figure 3.88: Test 1A vs 2A Round Robin Link Bandwidth Utilisation

3.6.3.2 Unbalanced Load Scenarios in Tests 1B and 2B

In Tests 1B and 2B the homogeneous and heterogeneous 802.11 interface bonds were subjected to an identical *unbalanced* load in order to stress the bond throughput performance and evaluate the slow-down as the difference in applied load increases. The second bond link was lightly loaded with a static load of n = 1 while the load on the first link was varied from n = 1 to n = 100.

The main finding was that the homogeneous 11ac-11ac bond outperformed the heterogeneous 11ac-11ah bond in terms of mean link utilisation by 22.3% and the mean bond throughput by 42.64%. As seen in Figure 3.89 for both bond types the achieved aggregate throughput displayed a non-linear decreasing output as the applied load was raised.

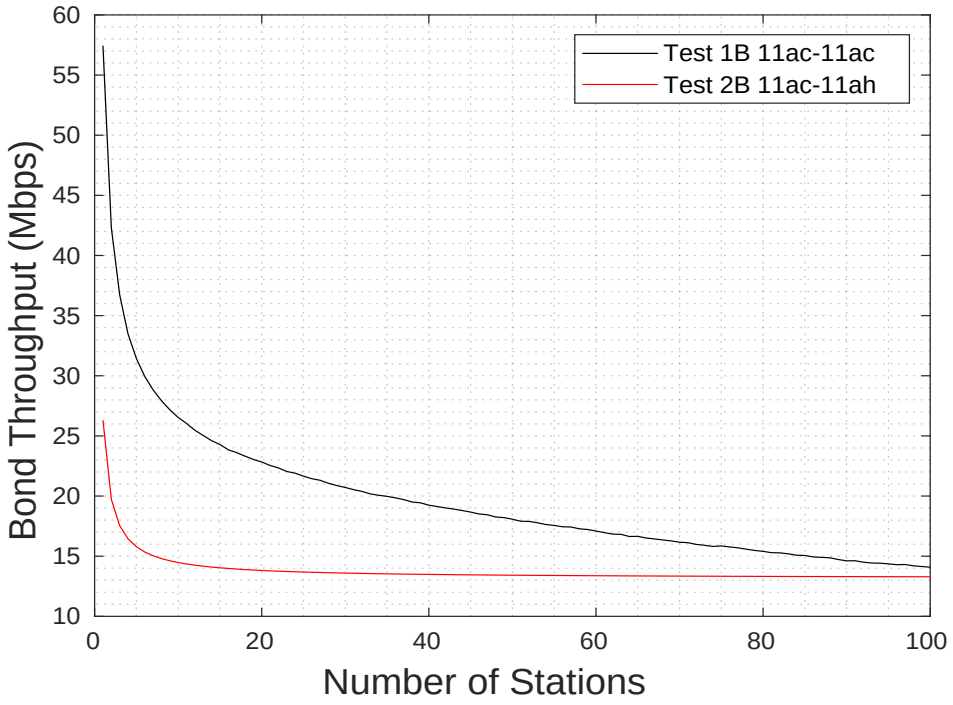

Figure 3.89: Test 1B vs Test 2B Round Robin Throughput

However it is difficult to determine the general behaviour by examination of the achieved bond throughput results alone, so for this we look to the link throughput difference and link bandwidth utilisation results for further insight. For the homogeneous 11ac-11ac bond in Test 1B the difference in achieved throughput increased exponentially as the applied load was raised from a minimum value of 0 Mbps to a maximum of approximately 28.5 Mbps at the heaviest considered load of $\{n_1 = 1, n_2 = 100\}$.

For the heterogeneous 11ac-11ah bond the throughput of the two links was not equal at the same level of applied load despite both using the same raw PHY data rate of 65 Mbps. The throughput of the 11ac link was approximately 16 Mbps higher at the same load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ due to the higher access delay which reduced its achieved throughput. The performance disparity between links decreased as the 11ac link load was raised thereby gradually increasing the link bandwidth utilisation of the bond as shown in the following page.

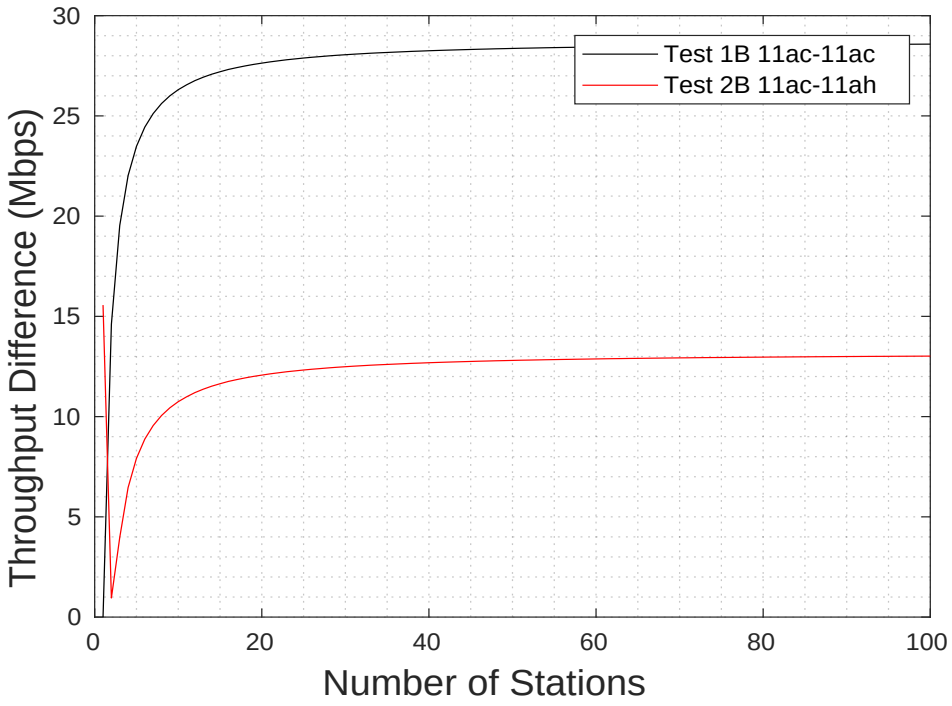

Figure 3.90: Test 1B vs Test 2B Round Robin Throughput Difference

The link bandwidth utilisation for both bond types is plotted in Figure 3.91 below. While for the homogeneous 11ac-11ac case in Test 1B the utilisation decreased non-linearly as the load was raised for the heterogeneous 11ac-11ah bond in Test 2B it *increased*. In Test 1B the decrease occurred as the 11ac Link 2 load was raised thereby gradually increasing the throughput difference which also increased the impact of the slow-down effect due to the in-order delivery assumption. By contrast for the heterogeneous 11ac-11ah we saw an interesting situation where the efficiency increased as the applied was raised. This occurred because as the 11ac load was raised its achieved throughput decreased towards that of the slower 11ah link thereby reducing the performance disparity and increasing the link bandwidth utilisation.

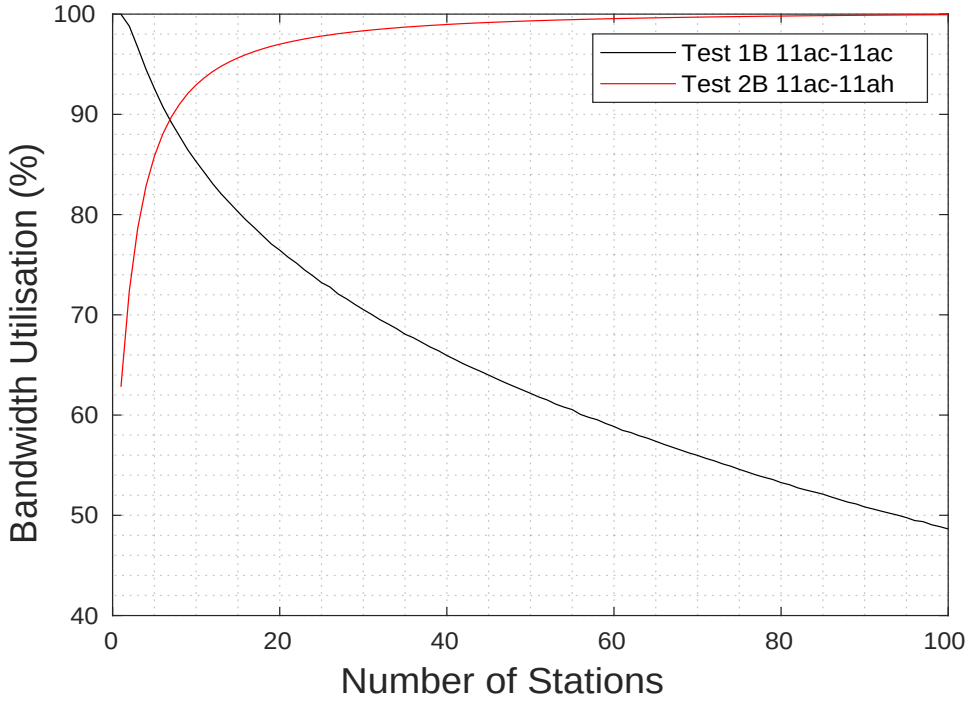

Figure 3.91: Test 1B vs Test 2B Round Robin Link Bandwidth Utilisation

3.6.3.3 Unbalanced Load Scenarios in Tests 1C and 2C

In Tests 1C and 2C the homogeneous and heterogeneous bonds were again subjected to the same unbalanced load in order to stress the bond throughput and evaluate the impact of the slow-down effect. The load on the first link of the bond was varied from n = 1 to n = 100 while the load on the second link was held constant at the moderate number of n = 50.

The main result from the comparison was that the homogeneous 11ac-11ac bond outperformed the heterogeneous 11ac-11ah bond in terms of the mean bandwidth utilisation by 21% and outperformed the mean bond throughput by 63.73%. The mean link bandwidth utilisation for the 11ac-11ac bond was 91.19% with a mean bond throughput of 1.2422 Mbps while for the 11ac-11ah bond the utilisation was 70.47% with a mean throughput of 0.7405 Mbps. Again the poor throughput performance seen was due to the difference in DCF parameters between the two 802.11 types used in the interface bond.

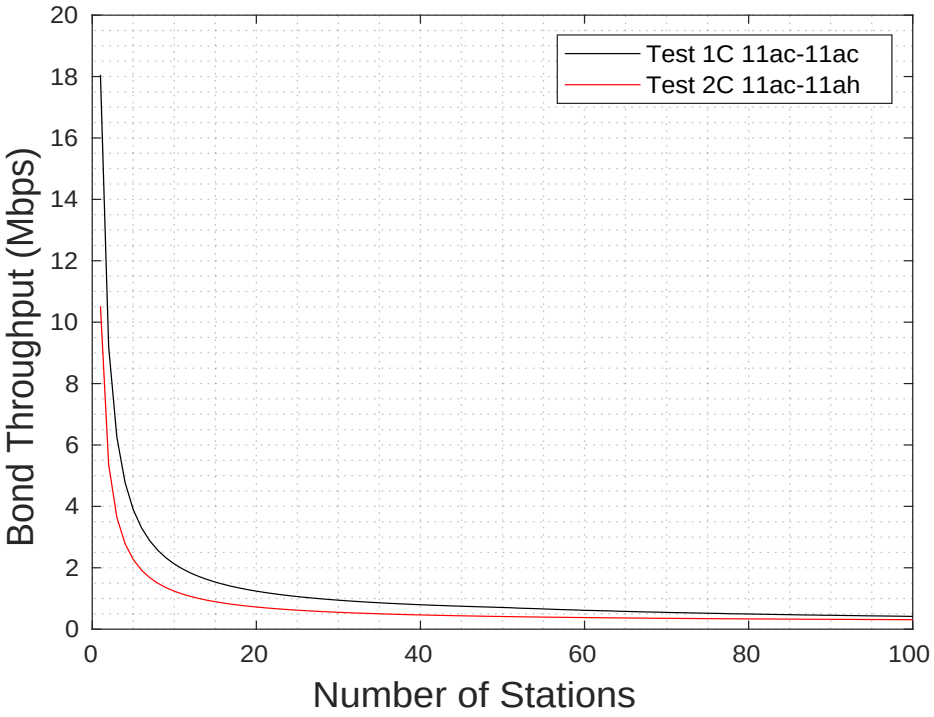

Figure 3.92: Test 1C vs Test 2C Round Robin Throughput

The bond link bandwidth utilisation in Figure 3.93 further illustrates the difference in behaviour between the two bond types. The utilisation of the homogeneous bond increased non-linearly as the applied load was raised from $\{n_{11ac} = 1, n_{11ah} = 50\}$ to $\{n_{11ac} = 49, n_{11ah} = 50\}$ and reached a maximum 100% at $\{n_{11ac} = 50, n_{11ah} = 50\}$. When the load was further increased towards $\{n_{11ac} = 100, n_{11ah} = 50\}$ the utilisation then decreased non-linearly. By contrast the utilisation of the heterogeneous 11ac-11ah bond increased non-linearly throughput the test as the applied load was raised but never reached the maximum possible value of 100%.

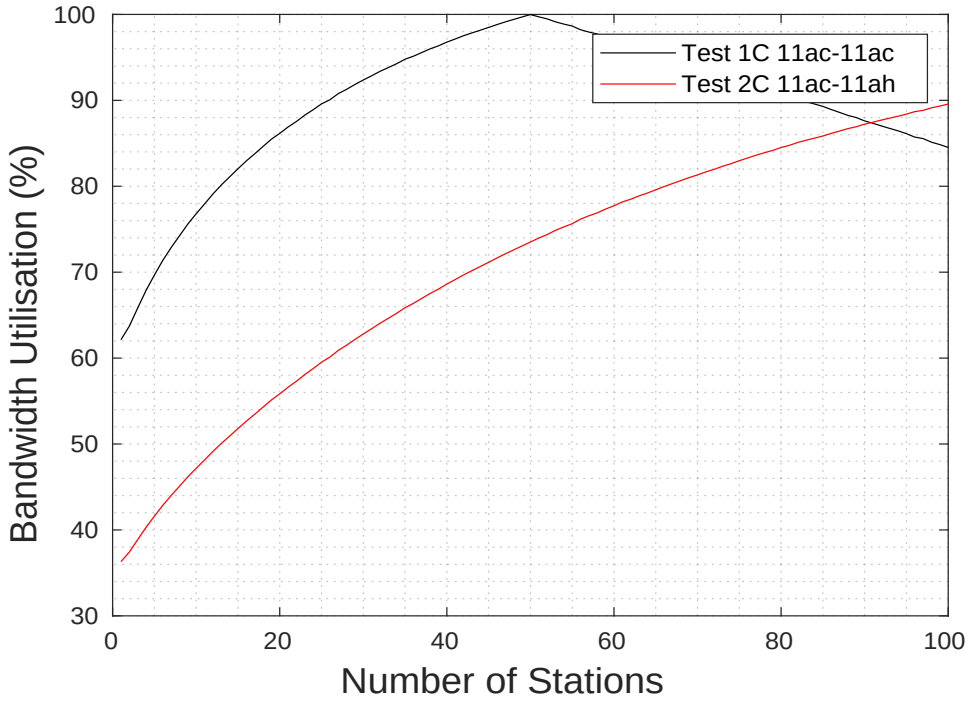

Figure 3.93: Test 1C vs Test 2C Round Robin Link Bandwidth Utilisation

As difference in the observed behaviour between the two bond types is due to the difference in DCF parameters and delay response between the 11ac and 11ah links as the applied load was raised. By virtue of its design for IoT the clocked-down PHY of the 11ah has a much higher access delay due to the difference in DCF parameters at the MAC layer. As a consequence the 11ah link throughput performance is significantly reduced in comparison to the 11ac link. The throughput difference between links then leads to a reduction in the overall mean bond throughput and link bandwidth utilisation due to the in-order delivery requirement and associated slow-down effect.

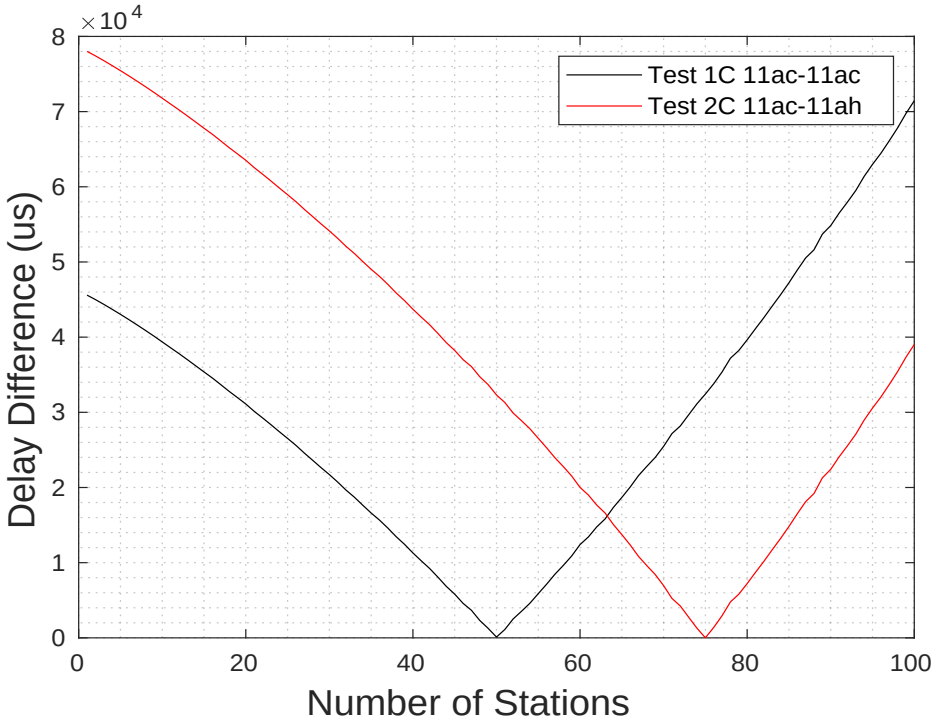

Figure 3.94: Test 1C vs Test 2C Round Robin Link Delay Difference

3.6.3.4 Unbalanced Load Scenarios in Tests 1D and 2D

In Tests 1D and 2D both bond types were again subjected to a similar unbalanced applied load scenario in order to evaluate the impact of the slow-down effect. The load on the first link was varied from n = 1 to n = 100 while the load on the second link was maintained at the maximum number of 100 nodes.

The main result from the comparison was that the homogeneous 11ac-11ac bond outperformed the heterogeneous 11ac-11ah bond in terms of the mean bond throughput by 67.1% and also outperformed the overall mean link bandwidth utilisation by 5.5%. The mean link utilisation for the 11ac-11ac bond was 81.61% with a mean bond throughput of 0.858 Mbps compared to 74.19% and 0.513 Mbps respectively for the 11ac-11ah bond.

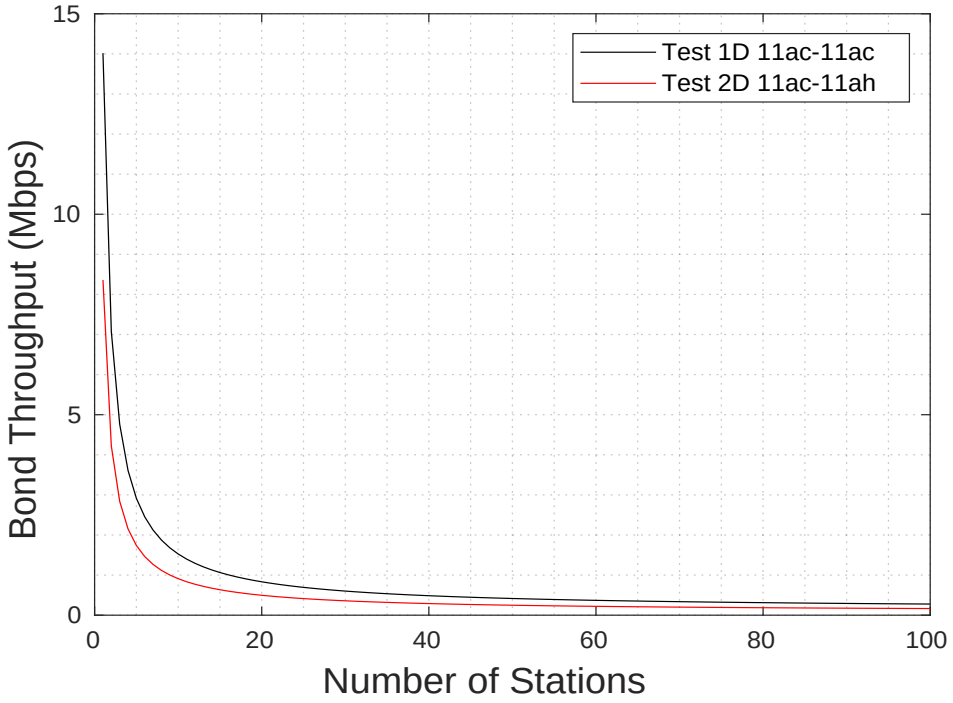

Figure 3.95: Test 1D vs Test 2D Round Robin Throughput

The achieved throughput for both bond types is shown in the graph above in Figure 3.95. As seen the homogeneous 11ac-11ac bond and the heterogeneous 11ac-11ah bond displayed a similar exponentially decreasing throughput as the applied load was increased. However the achieved 11ac-11ah bond throughput was considerably lower throughout compared to the 11ac-11ac bond due to the performance disparity between links (caused by the difference in DCF parameters) and the resulting slow-down effect caused by the in-order delivery requirement.

The difference in behaviour is also reflected in the link bandwidth utilisation results for each bond as shown in Figure 3.96 below. As seen both bond types displayed a non-linear increasing output as the applied bond load was increased. However the 11ac-11ac bandwidth utilisation was considerably higher than 11ac-11ah for all applied load combinations considered.

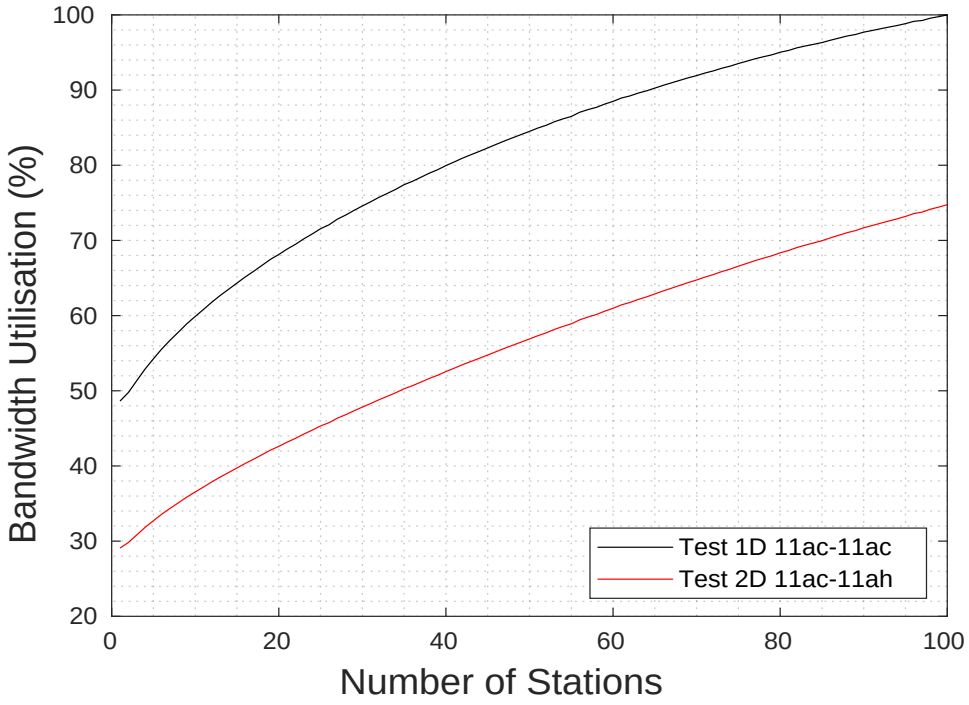

Figure 3.96: Test 1D vs Test 2D Round Robin Link Bandwidth Utilisation

The performance disparity between the two bond types is further illustrated by the delay difference results in Figure 3.97. For both the homogeneous 11ac-11ac bond and the heterogeneous 11ac-11ah bond a similar exponentially decreasing simulation output was seen. However the 11ac-11ah bond delay difference was approximately 75000 μs higher compared to the 11ac-11ac bond for all applied load combinations considered.

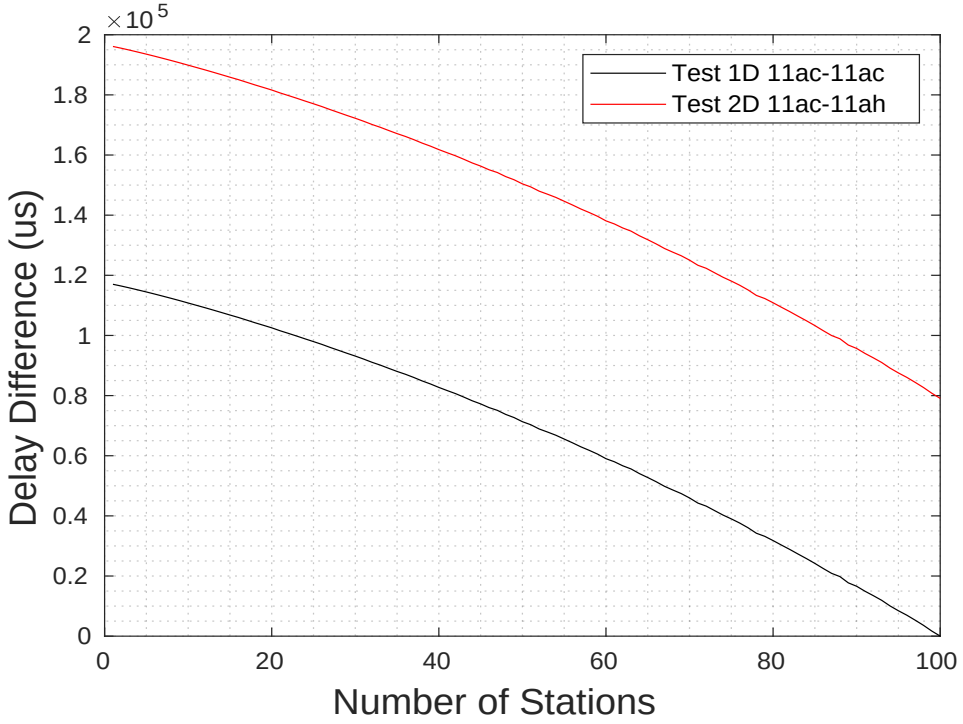

Figure 3.97: Test 1D vs Test 2D Round Robin Link Delay Difference

3.7 Chapter Conclusion

Chapter 3 evaluated the long-term throughput performance of the traditional static approach to round robin interface bonding at the Link Layer using dual 802.11 links.

During the experiments presented, the bond throughput and link bandwidth utilisation were measured for a dual-11ac interface bond and a heterogeneous 11ac-11ah bond under different combinations of applied link load. A significant degradation was observed in the achieved bond throughput due to the inherent slow-down effect when there was a disparity in the individual 802.11 link performance. In the worse case scenario seen during the simulation testing, the overall mean bandwidth utilisation was just 65%, such that over one third of the entire available link capacity was wasted.

Furthermore the condition where the aggregate bond throughput was reduced to below that of the single fastest available link, as seen in multiple experimental works, was found to be a common occurrence due the relative performance between the individual slave 802.11 links at the MAC layer. Notably, with respect to the previous related works, the above condition was present even with ideal RF conditions and not considering the degradation due to instability effects on the upper layer protocols such as TCP, as shown by Jayasuriya in [92].

We also saw how the difference in DCF parameters at the MAC layer between individual 802.11 slave links can adversely impact the overall bond throughput and bandwidth utilisation performance. In addition the simulation performance results provide a mapping between the individual single-link statistics, which is then used to predict the bond throughput based on the current state in the dynamic interface selection study in the following chapter.

CHAPTER 4

TOWARDS DYNAMIC 802.11 INTERFACE BONDING USING SDN

As the architectural principles of SDN solidify, research attention turns to more advanced and experimental applications of the technique. A particularly interesting application is the leveraging of SDN towards efficient dynamic 802.11 interface bonding. Multiple proof of concepts are published demonstrating the potential performance benefits of the technique but are limited to small-scale hardware test-beds. This chapter presents the first rigorous analysis of the performance of such systems for a large number of individual link throughputs, and presents a novel, predicted-throughput-based decision-making algorithm designed for efficient and robust dynamic link selection for heterogeneous 802.11 devices. The evaluation results show that when compared to the contemporary approach by the authors Nam et al, the proposed algorithm increases the achieved mean bond throughput by 0.79% and the link bandwidth utilisation by 6.16%, while protecting against the counter-productive and wasteful use of bonding at all times.

4.1 Chapter Research Overview

This main research chapter investigates efficient dynamic interface bonding for heterogeneous 802.11 devices by levering the principles of Software Defined Networking.

SDN significantly improved the management of high-performance data centre networks, and since the release of the OpenFlow protocol in late 2009, the paradigm has been widely adopted and has also found application in the enterprise WAN, cellular WWAN, and 802.11 WLANs.

Several small-scale, hardware-based proof of concepts have been presented that demonstrate the benefits of the SDN-assisted approach, including the main thesis reference work, the dynamic interface selection framework for MPTCP interface bonding over heterogeneous 802.11 devices by the authors Nam et al. in [130]. As discussed in more detail later in Section 4.2.3, the approach involves the centralisation of bond functions such as link state monitoring, dynamic link selection,

and dynamic bond scheduling, in the SDN controller. However, there are several design challenges introduced related to the dynamic wireless channel, heterogeneous link performance, and the inherent bond throughput slow-down effect for in-order delivery. There are several open research questions related to the centralised decision-making algorithms and related performance metrics used, particularly with respect to dynamic link selection.

Towards the efficient and robust dynamic interface bonding of heterogeneous 802.11 devices for the purposes of boosting wireless access at Layer 2 using SDN, the work presented in this chapter independently evaluates the performance of the key related work Nam et al, and then proposes a novel, predicted- throughput-based dynamic link selection algorithm designed to give specific protection against the counter productive scenario where the bond throughput is reduced below that of the fastest available 802.11 link.

4.1.1 Chapter Research Objectives

The main goals of the chapter research work are summarised as follows:

1. **Evaluation of contemporary approaches for dynamic 802.11 link selection using software defined networking:** The first aim of the chapter research is the investigation of efficient state of the art decision making algorithms and related performance metrics for dynamic bond interface selection using heterogeneous 802.11 devices assisted by Software Defined Networking. Multiple small-scale, hardware-based proof of concepts have already been proposed that demonstrate the potential application of SDN for the dynamic 802.11 interface bonding use case. As discussed previously the main related work is the SDN-based MPTCP solution by Nam in [130], which demonstrates the benefit of the approach using a dual 11g-11n interface bond. However, only a small number of individual link throughputs were tested in the experiments, and therefore in order to provide a more rigorous analysis, in the current study a much higher number of possible individual link loads and throughputs are considered.

2. **Exploration of novel algorithms for the efficient and dynamic bonding of heterogeneous 802.11 devices:** Informed by the above evaluation of the state of the art, the second research goal of the chapter work is to develop and evaluate novel algorithms for efficient and robust dynamic interface selection of heterogeneous 802.11 device types. In particular, one of the key design requirements of the developed algorithm is to provide protection at all times against the counter productive scenario revealed by the static round robin evaluation in the previous chapter, whereby the impact of the inherent slow-down effect for in-order delivery is so severe that the achieved bond throughput is reduced below that of the single fastest available 802.11 link, therefore rendering the use of the bonding technique pointless.

4.1.2 Chapter Research Contributions

Towards the fulfillment of the primary aims of the thesis and chapter research, the main contributions of the work presented are:

1. **Independent evaluation of the contemporary dynamic link selection algorithm by the authors Nam et al:** The first substantial research contribution of the chapter work is the independent evaluation of the state of the art approach to dynamic link selection algorithm for MPTCP by the authors Nam et al. as published in [130]. The test scenarios involve a range of applied link loads and PHY data rates for both homogeneous 11ac-11ac and heterogeneous 11ac-11ah interface bonds at Layer 2, in a directly connected point-to-point topology designed to reflect the case where link aggregation is used to boost access speeds over the highly contested wireless last-hop.

2. **Development and testing of a novel Maximum Throughput algorithm for dynamic 802.11 link selection:** The primary research contribution of the chapter/thesis work is the development and evaluation of a novel Maximum Throughput (MT) algorithm for dynamic 802.11 link selection, designed to match the Nam algorithm performance while also guarding against the counter productive scenarios identified in Chapter 3, where the inherent slow-down effect is so severe that the aggregate bond throughput is reduced below that of the fastest available single 802.11 link.

4.1.3 Summary of Chapter Findings

From the evaluation of the state of the art, throughput-gap-based approach to dynamic link selection for 802.11 bonding by Nam, the main research findings were:

- The dynamic slave selection algorithm by Nam was able to significantly increase the overall mean 802.11 bond throughput by 26.81% and the link bandwidth utilisation by 7.28% compared to the traditional static round robin approach. Although designed for use at the Transport Layer with end-to-end bonds, the throughput-gap-based dynamic link selection algorithm performed well in the evaluation for the point-to-point Link Layer bond considered. Furthermore the overall performance of the Nam algorithm was found to be slightly worse for the more heavily unbalanced heterogeneous 11ac-11ah interface bond.

- However, in terms of effective link selection decision-making, the performance was not robust for all applied loads combinations considered in the test. Specifically, the Nam algorithm was unable to guard against the damaging scenario where the bond throughput is reduced below that of the fastest available single link therefore rendering the use of bonding counter-productive to the aim of increasing throughput. During the Nam evaluation, this scenario was found to be a reasonably common event, occurring at various applied loads in 7 out of the 8 tests performed, and was observed on far more occasions for the heterogeneous 11ac-11ah bond when compared to the homogeneous 11ac-11ac bond.

The subsequent findings from the evaluation of the proposed predicted-throughput-based dynamic link selection were as follows:

- The novel MT slave selection algorithm was also found to significantly increase the mean 802.11 bond throughput when compared to the traditional static round robin configuration for the majority of the applied load scenarios considered. The MT algorithm at least marginally outperformed the Nam algorithm in every test with the exception of the balanced load scenario in Test 1A. Comparing against Nam in all eight individual tests the overall mean bond throughput was increased by 0.79% and the mean link bandwidth utilisation by 6.16%.

- As per the intended design and in contrast to the Nam approach, the overall decision-making of the MT algorithm was robust for *all* applied load combinations considered in the testing. In particular, protection was offered at all times against the scenario where the bond throughput is reduced below that of the fastest available single link. Notably, the predicted-throughput-based dynamic link selection algorithm was able to successfully guard against the above counter productive condition while at least matching the Nam algorithm performance at all other times.

4.1.4 Chapter Organisation

The remainder of the chapter is structured as follows. The relevant background information pertaining to SDN, wireless SDN, and SDN-assisted interface bonding is presented in Section 4.2. The independent evaluation of the Nam algorithm is presented in Sections 4.3 to 4.5. The novel, predicted-throughput-based algorithm for dynamic link selection is presented and evaluated in Sections 4.5 to 4.6, and finally, there is a short chapter conclusion and summary in Section 4.9.

4.2 Chapter Background Information

This section provides all the required chapter background information related to the subjects of Software Defined Networking (SDN) in Section 4.2.1, Wireless Software Defined Networking (wSDN) in Section 4.2.2, and the main thesis research theme of SDN-assisted Interface Bonding (SDNIB) in Section 4.2.3.

4.2.1 Software Defined Networking

This subsection provides an overview of SDN, an approach to Internet Protocol (IP) network implementation that decouples the control plane from the data plane and places the network intelligence inside a logical central controller. This is in contrast to traditional IP networks in which both planes reside in a single, vertically integrated network device. In an SDN architecture, the controller has a global view of network state and programs the forwarding behaviour of data plane devices using a southbound interface such as OpenFlow [44]. SDN applications running on the controller perform functions such as centralised routing, L3 addressing, virtual private networking, and finely-grained, multi-layer security.

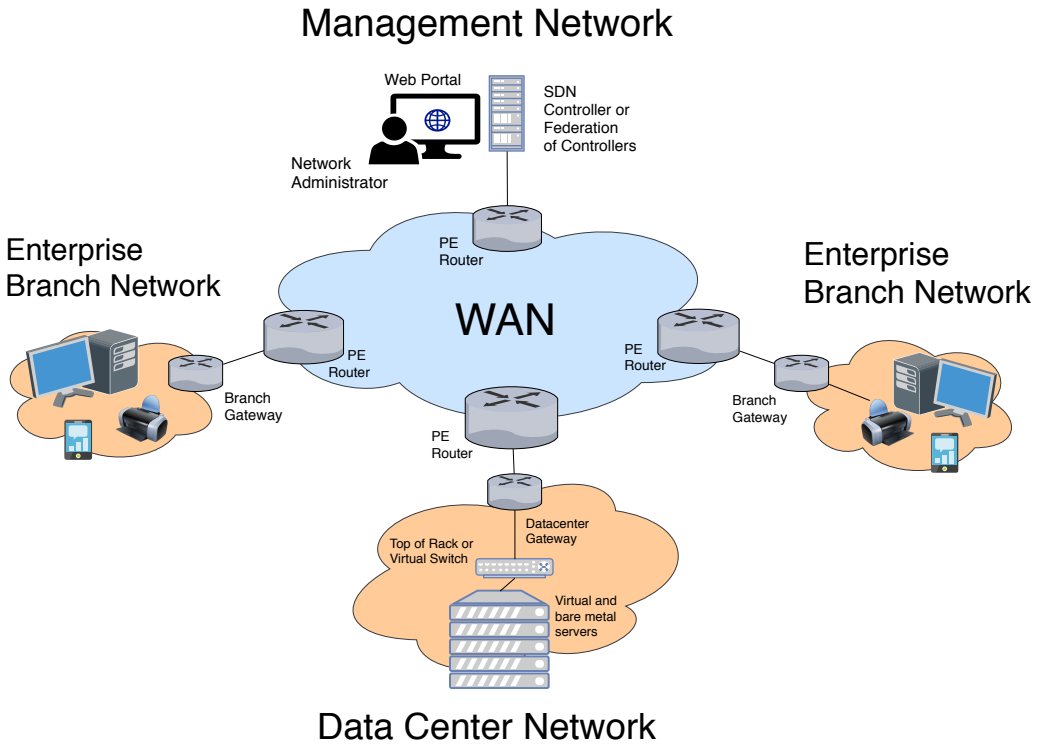

Figure 4.1: Example SDN Topology

4.2.1.1 Early Development

SDN was originally developed out of the need for a disaggregated, programmable, and flexible data plane in large-scale data centre networks. As server virtualisation become the new norm, the traditional integrated control and data plane design became limiting in several ways (see "Data Centre Networks are in my way" [85]). For example, while a new virtual server instance and applications could be spun up in minutes, creating the required network connectivity still required manual configuration, which could take hours or even days due to the strict administration and workflow processes used in such environments. In addition, if a virtual machine was migrated to another physical server or site, further manual configuration was needed. This impacted service delivery in several ways, for example by increasing the time needed to deploy new applications, increasing fail-over recovery time, and reducing the ability to scale offered services flexibly and efficiently. The introduction of software-defined control into data centres allowed the network to match the abilities of the IT infrastructure: SDN-enabled software switches and agent applications were installed on hypervisors and listened for events such as virtual machine creation and migration. These events were reported to the management plane software, which could then program the required configuration of the software switches through the SDN controller using a southbound interface, which was in most cases, the OpenFlow protocol. SDN therefore began as a means of providing a fast and scalable solution to immediate practical problems in data centre networks.

4.2.1.2 Definition of SDN

In a wide ranging literature survey published in 2015 [104], the authors Kreutz et al. define the four key pillars of an SDN architecture as:

1. Decoupled Control and Data Planes: The network control plane functions are removed from devices such as data centre gateways, leaves/spines, and virtual switches which become much simpler forwarding elements.

2. Flow-Based Packet Forwarding: Packets sent between two end points are represented as a flow consisting of a set of header values and corresponding actions for which all packets matched by the flow are performed.

3. Centralised Control Logic: Network control functionality resides in an external centralised entity usually running as a virtual machine on a commodity hypervisor. Commonly found protocols include include IP, VLAN/VXLAN, GRE, IPsec, MP-BGP, VPRN, and MPLS.

4. Programmable Network Logic: Software applications at the management layer provide user-friendly programming abstractions sent via the northbound API to the SDN controller, which translates and then configures the required data plane behaviour.

Speaking at the 2011 Open Networking Conference [170], Professor Scott Shenker of UC Berkeley defined SDN using three fundamental abstractions:

1. Forwarding Abstraction: This abstraction allows the controller to program any forwarding behaviour required while being shielded from network device hardware and software details. This is provided by a southbound API such as OpenFlow.

2. Distribution Abstraction: This abstraction shields user applications from the underlying distributed state of the network and provides a logically centralised point of control. This is achieved by means of the SDN controller.

3. Specification Abstraction: Allows a user application to express the desired network behaviour but not be responsible for the implementation. This can be achieved using virtualisation but may also be done using network programming languages that map the abstractions used into a physical network configuration on devices such as routers, switches, firewalls, or virtualised network functions.

4.2.1.3 Benefits of SDN

There are several benefits of the software-defined approach to IP networking:

- *Simplified Network Management:* Using a global view of network state, the SDN controller can implement automated, network-wide policy distribution and device configuration. Routing and security policies may be set by network administrators in a management plane portal, which then filter down through the centralised control plane to network devices in the data plane. SDN applications running on top of the controller translate the policies into the required forwarding table entries and switch configuration, and device state and statistics can be reported back up the management plane for monitoring. This automation removes the need to manually configure and physically inspect network devices such as routers and firewalls, which can greatly reduce the overall operational expenditure. Artificial intelligence and machine learning are view as key enabling technologies in SDN automation research, for example Intentional Networking [61].

- *More Efficient Networks:* SDN can improve network utilisation and efficiency in several ways. For example, by monitoring link usage, balancing traffic load across multiple paths, centralising resource allocation, rate limiting certain types of traffic, and automatically turning switch ports on and off to reduce energy consumption. In addition, data centres no longer have to use costly over-provisioning, as SDN allows the number of virtual servers to scale dynamically to meet the peak user traffic demands.

- *Improved Network Security:* The SDN controller has access to a wealth of Layer 2 to Layer 4 information, and is able to monitor application data traffic patterns and apply finely-grained security policy at a network-wide level. SDN overlay techniques provide automated per-user isolation through use of tunnelling and can also automate data privacy via encryption technologies such as IPsec [66, 98]. Furthermore, centralised security management leads to less human configuration errors that could otherwise leave the network vulnerable to attackers.

- *Faster Network Development:* The global visibility of the SDN controller and programming power of south bound interface such as OpenFlow, greatly simplifies the deployment of advanced network applications and services, providing faster network development and experimentation. In addition, by running in isolated network slices, each with its own dedicated routing and switching control plane, experimental functions can be tested in a controlled fashion on production networks without the risk of service impact.

4.2.1.4 SDN Controllers

The SDN controller is the logical brain of a software-defined network and represents the displaced network control plane element. It is responsible for programming the forwarding behaviour of devices in the data plane and maintaining a global map of network topology and state. The controller presents two interfaces: one southbound interface towards devices in the data plane, and another northbound interface towards SDN management applications. Depending on the number of controllers in the network there may also be a horizontal interface to facilitate inter-controller communication and synchronisation. OpenDaylight is the most advanced and well known of the open-source controller platforms and is a popular choice for research [89, 102]. The controller was created by an international consortium of industry partners, including Arista Networks, IBM, Juniper, Cisco, and Citrix. Other examples of well-known open-source SDN controllers include Ryu [23, 43, 148], ONOS [34, 203], and Floodlight [59, 122].

4.2.1.5 Southbound Interfaces

The southbound interface is the means by which the decoupled control plane and data plane elements communicate with each other over the network. In traditional switch architectures, this is typically implemented using a proprietary interface between the control module and the line cards. The predominant southbound interface is the OpenFlow protocol, which arose from work at Stanford University circa 2008, but others do exist, for example ForCES [84] and POF [190, 223]. OpenFlow is synonymous with SDN due to being one of the first de facto standards. The protocol uses a flow-based programming abstraction to define forwarding behaviour in OpenFlow-enabled switches. On packet arrival, the header fields are evaluated against a list of match criterion called flow entries. In the event of a match, counters for that entry are updated and a set of associated actions are performed, for example dropping the packet or rewriting the header. Entries may match against fields such as source and destination IP address, MAC address, Layer 4 ports, VLAN tags, and QoS information.

4.2.1.6 Software Switches

An SDN architecture also contains other components such as hypervisors, gateways, firewalls, and virtual switches. The virtual switch is a critical component in a virtualised SDN environment and most hypervisors offer at least basic virtual switching capabilities. Some offer more advanced functionality, such as the VMWare vSphere data centre solution [114, 169], which provides a fully featured virtual switch that can be logically distributed over multiple physical hypervisors and data centre sites. Open vSwitch [106, 218] is a open-source, distributed, multi-layer software switch designed for use in virtualised environments. The switch is fully SDN-enabled by support for the OpenFlow protocol and is comprised of two main components: *ovs-vswitchd*, the main user-space software component written in platform portable C, and the data-path module, which

provides fast packet processing in kernel space and is platform-dependent. The first packet of a flow always results in a table-miss in the data path, and the packet is then redirected to *ovs-vswitchd* in order to determine how the packet should be handled. The packet and forwarding information is then sent back to the kernel module, which caches the actions. The *ovs-vswitchd* user space component may be programmed by an SDN controller using the OpenFlow protocol, and it can also configure other settings such as the number of ports on the switch and QoS via the OVSDB protocol and the *ovsdb-server* component.

4.2.1.7 Network Hypervisors

Network hypervisors allow the partitioning or slicing of a single physical network into independent logical resources [154]. A network hypervisor forms an intermediate layer between the control and data plane and acts as a transparent proxy between the SDN controller and network devices. A popular choice for researchers is FlowVisor [112, 147, 161], which slices the network resources in terms of available topology, bandwidth, CPU rate, forwarding table quotas, and flows. FlowVisor works by intercepting and rewriting OpenFlow messages between the controller and data plane devices. A separate control plane is maintained for each network slice and in the data plane forwarding rules are applied to traffic on a per-slice basis. In this way, each device or slice believes it is communicating with its own, dedicated network.

4.2.1.8 SDN Challenges

There are still multiple ongoing research challenges with respect to software defined networking. A non-exhaustive list is provided below:

1. *Control Plane Scalability:* As the number of physical and virtual hosts in a network grows, the controller must manage an increasing number of routing and forwarding table entries, which can impact network performance and eventually lead to memory exhaustion and controller failure. One solution is to implement a controller cluster or federation over which to balance the control plane burden, in effect, redistributing the control plane across the network in a strategic way. Note that this requires the introduction of a horizontal interface between different controllers operating in a cluster.

2. *Control / Data Plane Availability:* The resilience of a software defined network is dependent on the high availability of the centralised SDN controller and suitable fault tolerance and detection in the data plane. Insufficient bandwidth or high latency in the control channel can also result in availability issues causing performance degradation and out-of-date information, and the problem is exacerbated considerably with greater scale.

3. *Programming Abstractions:* Southbound interfaces provide an open API for programming the data plane, but protocols such as OpenFlow are still relatively low-level and lack compiler checking and expressive power. Developing high-level abstractions for SDN programming is an ongoing research issue. Popular examples of high-level SDN languages proposed are Nettle [207], Frenetic [65], and Pyretic [153].

4. *Network Automation:* SDN provides a powerful network development platform using open APIs such as OpenFlow and expressive high-level languages such as Pyretic. However, developing a complete control logic that can be applied efficiently to all network types is still a daunting task, especially considering the size, diversity, and complexity of current state of the art deployments. SDN automation using artificial intelligence and machine learning has an extremely intriguing potential in this regard.

5. *SDN Security:* Although SDN can improve network security via network slicing and policy-based access control, the centralisation of intelligence in the controller provides a single attack surface for the entire network, making the architecture more vulnerable to DoS attacks in some respects. In addition, mutual authentication is still required between the controller and network devices during intitialisation in order to ensure that malicious SDN switches do not gain unauthorised access. Such networks are also susceptible to SDN malware and root-kits.

4.2.1.9 SDN Research Overview

This subsection provides a short survey of SDN related research works. A number of excellent, general SDN surveys can be found in the literature. Kreutz et al. [104] published a particularly influential and comprehensive review in 2015. Other good early examples include Jarraya [91] and Govindarajan [76].

4.2.1.9.1 Network Load Balancing SDN-assisted load balancing has been a popular research area in both wired and wireless networks. Load balancing configuration is traditionally end-point driven and configuration may be static, dynamic, or a combination of both. Dynamic, SDN-based methods can be more efficient due to the ability to adapt to changing network conditions, which can benefit the network as a whole by maximising scalability, minimising response time, and increasing throughput and avoiding situations where a single device is overloaded. Neghabi et al. [134] compares and categorises different SDN-assisted load balancing works according to deterministic or non-deterministic approaches. In general, deterministic approaches were found to provide better throughput and also had a better response time and lower overhead. Non-deterministic approaches provided greater utilisation and better delay compared to deterministic types, but suffered unacceptable levels of throughput. Both were seen to lead to unacceptable levels of packet loss and energy consumption. Other examples of SDN-assisted load balancing works include Raza et al. [152] and Zhong et al. [232].

4.2.1.9.2 Network Slicing and Tenant Isolation Several works examine the ability of SDN to slice the resources of the physical network infrastructure. Richart et al. [154] examine network slicing and resource allocation in software-defined wireless networks. In typical LTE-based solutions the network resources are allocated by assigning different PRBs to individual slices. Network resources are abstracted into a 3-dimensional grid comprised of time, frequency, and space, forming a central pool from which to allocate. In 802.11 WLANs there are several strategies, including traffic priority control using 802.11e QoS, slice-scheduling similar to TDM, and traffic shaping. Examples of other researches in network slicing include CellSlice, which targets WiMAX networks [103], Vi-Fi targeting Wi-Fi networks [78], and Zaki et al. [224] which targets LTE.

4.2.1.9.3 SDN Modelling Several authors evaluate different aspects of SDN performance, for example control plane reliability and scalability, using modelling techniques such as queuing theory and network calculus. Kang et al. [96] formally model and validate the OpenFlow protocol using the Algebra of Communicating Shared Resources. Asian and Matrawy [24] model SDN control applications using affine transformations in vector space. Petrov provides a model for predicting forwarding rule counter values in [144]. Nguyen [137] applies mathematical techniques to analyse the availability of SDN resources such as physical hosts, switches, routers, and virtual machines. Goto et al. [75] analyses the performance of an OpenFlow switch using queuing theory. Other queuing theory-based example include Sood [184], which models an SDN/OpenFlow switch as a single M/G/1 queue.

4.2.1.9.4 SDN Programming Languages Several works examine SDN programming including the different languages and abstractions used. While protocols such as OpenFlow provide a workable mechanism for communication between the control and data planes using its flow-based programming abstraction, it is still a relatively low-level and inflexible API for developers. Examples of flexible, high-level languages for programming SDN include Nettle [207], Frenetic [65], Merlin [185], and Pyretic [153]. Trios et al. [173] published a taxonomy of high level SDN languages in a survey which categorises three different levels of SDN programming:

1. *Low-Level Programming:* Where network devices are programmed directly via the southbound interface, for example using OpenFlow. Here, the authors compare the OpenFlow protocol to traditional computer assembly language.

2. *API-Based Programming:* This when SDN applications on the controller use the northbound API. However, this type of programming shares may of the same complexities and dependencies of the low-level languages.

3. *Domain Specific Programming:* These languages provide expressive programming power through the use of notations and abstractions, but require the use of a compiler framework to translate the high level abstractions into a format understandable by the northbound API of the controller.

4.2.1.9.5 Software-Defined WAN The SD-WAN has seen an increase in research popularity in recent years. Sadasivarao et al. highlights the need for a software-defined data centre WAN interconnect in [159]. Traditional interconnects are manually/statically configured with routing decisions made by a vertically integrated device at the network edge. The authors propose an SDN architecture that abstracts core optical transport nodes into virtual, programmable, OpenFlow enabled switches. A recurring theme for SD-WAN research is controller placement, which deals with the locations of SDN controllers and their associations with data plane devices, and is linked to control plane resiliency. For example, Tanha et al. [195] propose a solution to controller placement that takes into account the control plane latency and controller capacity. The authors Xiao et al. [213] use a spectral clustering controller placement algorithm that partitions the network into separate logical SDN domains. Gallegos-Segovia et al. [69] investigate the performance implications of hosting an SD-WAN controller in the cloud. Other SD-WAN works include Al-Sadeq et al. [18], which examines routing algorithm optimisation, Lara et al. [108] which considers WAN virtualisation and dynamic bandwidth provisioning, and Vdovin [205], which investigates network utilisation optimisation.

4.2.1.9.6 SDN Security SDN is viewed as a key enabling technology for securing next generation IP-based networks. Multiple research works have examined different aspects of SDN security. Tatang et al. [197] propose SDN-Guard, a system for protecting SDN controllers against root-kit malware. Tselios et al. [202] propose a blockchain-based SDN security solution for IoT deployments. Yan et al. [218] review DDoS attacks in cloud computing environments and identify SDN as a tool for mitigation. Siddiqui et al. [172] and Liang et al. [111] both propose SDN-based security solutions for 5G wireless networks. Ketel [99] examines ways to use SDN to improve security in BYOD 802.11 WLANs.

4.2.1.9.7 Internet of Things SDN is also regarded as an important enabling technology for the Internet of Things which will connect billions of different devices to the Internet with diverse traffic characteristics and QoS requirements. A software-defined approach will be key in managing such a large number of connected devices. An example SDN architecture for IoT is found in [32], in which the controller translates node service requirements into traffic flows using a network calculus based framework. In [198] the authors Thubert et al. propose a fully scheduled SDN solution designed for wireless IoT devices. The authors Nastic et al. were one of the first to apply a combination of SDN and NFV to IoT networks in [133].

4.2.1.9.8 Network Automation A recent research trend has been the application of artificial intelligence and machine learning in the control plane of software-defined networks. Intent-based networking is a term given to advanced network automation systems that rely on such techniques, and SDN provides the ideal set of statistics and programming interfaces. Xie et al. [214] categorise such algorithms into the following types: (1) Supervised Learning, in which

large, labelled data sets are fed to an algorithm in order to train it to perform a specific task, for example k-Nearest Neighbour, (2) Unsupervised Learning, in which an algorithm attempts to find patterns in data without a labelled input learning set, for example the k-Means algorithm, (3) Semi-Supervised Learning, which is combination of both previous types and uses both labelled and unlabelled data, for example Pseudo Labelling [109], and finally (4) Reinforcement learning, which involves learning entities called agents capable of inferring the best interaction with its environment to achieve the maximum reward, for example Reinforcement Learning Game Theory [132]. Other examples of intent-based networking include Szyrkowiec et al. [192], Han et al. [86], Subramanya et al. [189], and Chamania et al. [36].

4.2.2 Wireless SDN

The application of SDN in wireless networks is far removed from its original use in large and highly dynamic data centre environments.

The most notably difference is the vastly expanded data plane, which consists of such wireless functions as interference mitigation, channel allocation, handover management, and transmit power optimisation. While traditional wired SDN already works over wireless devices such as 802.11, it does not provide specific control over wireless data plane functions. Therefore, in order to achieve the control-data decoupling and remotely manage the wireless data plane devices, it is necessary for the controller to implement a new programming application interface that goes beyond the standard OpenFlow protocol used in the data centre network and WAN. Closely related to wireless SDN is Software Defined Radio (SDR), which provides for complete programmability of the wireless data plane in software using Digital Signal Processing (DSP) techniques. SDN and SDR are therefore complimentary, and many open research questions exist around their integration.

The physical wireless channel also presents a significant challenge for SDN in terms of network monitoring and optimisation. Difficulties such as slow and fast fading, Doppler effects, and hidden and exposed nodes, mean that wireless links are characterised by randomly fluctuating link quality and limited bandwidth. In addition due to high levels of mobility wireless topologies are more dynamic compared to their wired counterparts, and are subjected to frequent changes in the number of connected users through handovers and re-associations. Under these conditions efficient resource allocation and interference management become challenging given the highly dynamic nature of the wireless links. The monitoring of link state and the reporting of statistics is one of the key benefits of SDN, and in addition to device CPU and memory, for wireless devices the two main monitoring functions are: (1) the link state, e.g. the link delay, and packet loss rate, and (2) topology discovery. Here, the adverse effects of the wireless channel can be smoothed by filtering out the short term variations, but a trade-off must be made against the overall responsiveness of the system.

Because of the congestion and co-channel interference problems inherent in wireless networks of all types, the discovery of the network topology is an important first stage in the optimisation process. The single-hop topology discovery can easily be performed by scanning for beacons which also provide the operating channels and signal strength of the neighboring APs. In a multi-hop wireless topology this involves the re-broadcasting of received Hello messages. However, in more sparse networks there may not be enough density to successfully identify every individual AP. When wireless users at the limit of the AP range suffer connectivity issues, they may also be within range of another AP at the same time, and SDN allows multiple controllers to exchange reachability information and allow users to connect without regard to the provider to which the underlying network infrastructure belongs.

While the 802.11f standard does provide some function for AP collaboration, SDN goes further by allowing controllers to perform channel allocation that transcends operator boundaries. In addition power control between APs helps reduce interference by associating users with the AP that is closest in terms of physical proximity, thereby reducing the overall power consumption. For security purposes SDN can also assist in detecting and disabling attacks by rouge APs through the combined effort of the legitimate APs. Monitoring information collected from all nodes in the network can be analysed by a central SDN application, and the risk mitigation performed by the installation of a single rule to all proper APs.

The slicing of the network is also more challenging in the wireless domain. For example with 2.4 GHz 802.11, an instance of a FlowVisor must manage a limited number of 3 independent and non-overlapping channels, but this can lead to logistical issues in network planning for adjacent WLANs due to the co-channel interference. For optimal performance in large wireless networks, FlowVisor requires global network visibility in order to re-use frequency bands in different geographic locations. It is possible to use time division multiplexing between nearby APs, but this needs coordination and finely grained synchronisation between wireless nodes, which is not possible under the current 802.11 specification. However, as discussed in Section 2.2.1 of Chapter 2, this is now set to change with the development of the new 802.11be version of the standard, which provides a specification for inter-AP information sharing and coordinated channel access. Also note that frequency division multiplexing techniques can be wasteful due to the use of guard intervals between adjacent channels, and the inefficiency only increases as the number of sub channels are added. While long-term fairness is ensured in a flat topology, this may not be the case in a multi-hop network where hidden and exposed nodes may be present.

In the following sections, the various technological solutions and research work related to the application of the SDN in heterogeneous wireless networks is presented according to the network type targeted. There is a particular focus on the Software Defined Wireless LAN (SD-WLAN), with further sections on software-defined cellular networks and wireless sensor networks.

4.2.2.1 Software Defined Wireless LANs

This sub-section describes the various architectures and related research towards the application of SDN in 802.11 Wireless Local Area Networks (WLANs).

Although many of the technical challenges, for example mobility, slow/fast fading, and co-channel interference, are similar across both network types, there are several ways in which WLANs such as 802.11 are fundamentally different compared to their cellular counterparts in the WWAN. For example, residential 802.11 WLANs differ substantially in that their deployment is entirely unplanned and therefore unoptimised. Furthermore, the user traffic is predominantly media based, with applications such as high definition video and audio streaming, online gaming, video teleconferencing, and home automation.

In the following paragraphs, related wireless SDN solutions and research are presented according to the target research area, for example network virtualisation, data plane programmability, and centralised resource allocation.

4.2.2.1.1 Southbound Interfaces Because of the vastly expanded data plane and control plane functions in wireless software-defined networks, is it necessary to implement new programming interfaces in order to facilitate the centralised monitoring and management of 802.11 APs. For example, CAPWAP [124] allows a controller to securely connect via the Distributed Transport Layer Security (DTLS) protocol to an 802.11 AP for the purposes of exchanging network state information and control data. Two types of MAC operation are supported: (1) Local MAC Mode, which is the traditional form of operation whereby most MAC functionality is implemented in the 802.11 AP and the controller role is negligible, and (2) Split MAC Mode, where real-time MAC functions are executed by the AP, while operations such as beacon generation and probe responses are managed by the remote controller. Another example is CPE WAN [64], which provides a dynamic programming interface between Auto Configuration Servers and Customer Premises Equipment (CPE) such as modems, VoIP telephones, and 802.11 APs, and allows dynamic provisioning, auto-configuration, and software/firmware image management. Devices behind a NAT box are able to securely discover servers and obtain their configuration, and the configuration servers can also initiate connections to CPE devices. The CPE WAN API is considered as an important enabling technology for M2M, but does not include any provision for user mobility.

4.2.2.1.2 Network Virtualisation The virtualisation of the WLAN involves the use of multiple virtual APs mapped to one or more physical AP infrastructure devices. The following paragraphs describe the various approaches taken in the available research literature.

Instead of extending the OpenFlow protocol to include the required wireless data plane programming ability, the OpenRoads [113] architecture developed at Stanford University relies on the use of SNMP for managing and configuring the 802.11 APs. Slicing is performed via FlowVisor for the data path and SNMPVisor is used to slice the AP configuration channels as required.

Odin [191] uses a Light Virtual Access Point (LVAP) that maintains individual client association state separately from the underlying physical AP. Each LVAP consists of client IP address, BSSID and SSID, and a set of OpenFlow processing rules, and are used to track the logical state of the client connection. To perform handover, the LVAP is simply transferred from one AP to another which reduces the required overhead.

SplitAP [29] supports network virtualisation and the efficient management of client airtime for uplink traffic. Each AP broadcasts the beacons for the different virtual networks associated, which requires a client agent application to adjust traffic based on instructions received from the controller. If the usage of a given slice is above the specified threshold, a new maximum uplink airtime is calculated and broadcasted into the network.

The Virtualized Access Network [177] is an SDN architecture designed for the virtualisation and centralised control of residential networks. The architecture consists of domestic 802.11 APs, a central SDN controller, and content provided servers, for example Netflix and Amazon Prime Video. In the solution the controller exposes a northbound API which allows content providers to monitor traffic and define end-to-end QoS and QoE.

CloudMAC [50] virtualises the 802.11 APs by placing the SDN controller within a remote cloud computing infrastructure. The primary components of the architecture are the APs, VAPs, OpenFlow switch, and the SDN controller. The only MAC functions performed by the physical AP is the forwarding of MAC frames from client devices and time-sensitive operations such as re-transmissions and acknowledgements. All other functionality is managed by the remote VAP in the cloud which handles the generation and processing of AP-related MAC frames. The APs and VAPs are connected by means of a Layer 2 tunnel using an OpenFlow-enabled switch.

EmPOWER [156] uses the LVAP virtualisation concept in order to decrease the overhead associated with client mobility. The Energino add-on allows energy monitoring and a REST API enables administrators to selectively turn off and on lightly loaded APs in order to reduce power consumption. EmPOWER 2 is a more comprehensive architecture using a 3-dimensional pool of frequency, bandwidth, and time resources, with the aim of efficient interference mitigation and reduced energy usage.

In [123] the authors propose a new southbound interface called the Ethanol protocol, which is used alongside OpenFlow in wireless SDNs. QoS on the APs is supported by the Ethanol agent application using Hierarchical Token Bucket (HTB) scheduling for per-flow programmability based on user traffic classes. Ethanol follows the object-oriented programming paradigm and treats all entities as a physical or virtual object with its own set of parameters and executable functions.

SD-WLAN [231] also implements the 802.11 MAC layer functionality in a remote controller using an extended version of the OpenFlow protocol. BeHop [221] provides traffic monitoring capability and efficient handover management using VAPs and a modified OpenFlow protocol. Other closely related examples include CUWN [179], AeroFlux [164], RCWLAN [129], vBS [128], BIGAP [235], and OpenSDWN [163].

4.2.2.1.3 Network Densification To address the challenges of wireless network densification and observability, DenseAP is proposed in [125]. The architecture consists of DenseAPs and the DenseAP Controller, and a daemon application running on the AP monitors the 802.11 interface statistics, for example, RSSI values, user traffic patterns, and a list of new user clients waiting for connectivity. The collected state information is then passed to the DenseAP controller which makes control decisions, tunes the network parameters accordingly, and then informs and updates the DenseAP devices running in the data plane.

The Cisco Unified Wireless Network (CUWN) [125] provides security, QoS, and seamless mobility using either the Lightweight Access Point Protocol or CAPWAP as the southbound interface. The architecture supports both the local MAC mode and split MAC mode enabled by CAPWAP. For Radio Resource Management (RMM), CUWN chooses a local controller from the list of APs found using the Neighbor Discovery Protocol.

4.2.2.1.4 Programmability In terms of enabling device programmability, DIRAC [228] uses a southbound interface along with two primary components consisting of the router core (running on commodity silicon) and agent application running on the 802.11 APs. The software agents receive messages from the router core and forwards them to the device driver. The messages and interactions between the router core and the agent applications consists of events, statistics, and actions. The agents also send periodic AP statistics to the router core, including metrics such as the packet loss rate and the signal-to-noise ratio.

Trantor [127] extends DenseAP to provide software-defined control of wireless clients, in addition to 802.11 APs. The control information exchanged between STAs and the DenseAP controller include packet loss rate, channel utilisation, and RSSI. A set of APIs is provided in order to allow centralised control such aspects of the client as association, channel allocation, transmission rate, and transmission power. For example the function $associate(AP_i)$, which instructs a client to associate to the AP with index i. However, there are no network control mechanisms provided that can exploit these programming primitives.

Dyson [126] extends Trantor to enable two-way communication between the 802.11 STA devices and the DenseAP SDN controller. However, only Dyson-enabled clients can initiate communication with the controller, which uses such information as the sum of RSSI values for all nodes in the network, the number of packets transmitted during a given window, and the channel utilisation to create a global network usage map. The map includes such information as the geographical node location, connectivity information for each STA/AP, the achieved airtime utilisation, and a database of historical network measurements.

The work in [217] simplifies programmability by extending the OpenFlow control interfaces according to the specification: (1) Capabilities, these interfaces are used to inspect the abilities of AP devices, including the supported number of channels and transmission power etc, (2) Configuration, which are the interfaces used to facilitate the configuration of physical and logical ports, (3) Events, which are interfaces used to report occurrences such as probes, authentication, and association to the SDN controller, and (4) Statistics, which allow the controller to query various wireless statistics associated with the 802.11 AP and STAs.

ResFi [227] targets information collection and sharing in residential 802.11 networks. The ResFi agent application runs in user-space on the AP and provides the IP connectivity for the wired backhaul, and also controls the process in which APs discover each other and the subsequent connection establishment. The agent shares the information collected with the controller and other APs. Individual APs can subscribe to the listening socket of the neighbouring devices to receive their updated state/configuration. Because the solutions relies solely on the interfaces provided by the *hostapd* software, ResFi does not require any kernel or drive modification.

4.2.2.1.5 Resource Allocation As mentioned previously, the centralised resource allocation facilitated by wireless SDN allows for new and efficient methods of managing the co-channel interference, and is one of the main prospects in wireless SDN. In such systems, a WLAN may be represented as a conflict graph with the vertices denoted the APs, and the edges giving the interference level between APs. The objective of the resource assignment is to colour the graph using a minimum number of colours in order to improve channel reuse and the overall capacity.

CloudMAC allows multiple interfaces belonging to a physical AP to be mapped to a VAP with interfaces that operate on different radio channels. If the network monitoring finds that a given client is currently using a channel with a high level of interference, then the controller sends a 802.11h channel switch message to the client containing instructions for the node to switch to a channel with less interference. Because clients are already associated with a VAP, there is no need to re-authenticate, as it can simply connect to any AP through its existing VAP.

Odin also provides a heuristic channel allocation mechanism based on sampled RSSI values, which always assigns the channel with the smallest maximum/average RSSI for each AP. EmPOWER2 uses the provided API to traverse the network map and detect any uplink or downlink conflicts that exist between different LVAP pairs. The graph colouring algorithm presented in [160] is used in order to assign channels in a way that avoids conflicts.

A large number of other related works also target the centralised resource allocation problem in software-defined WLANs. For example, the work in [166] formulates the channel assignment task as a binary integer linear programming problem, the authors in [158] propose a fair algorithm for traffic-aware channel assignment based on the user traffic requirements, and [116] which implements a predictive interference model based on user traffic and signal strength. Note that resource allocation in wireless networks using SDN is a substantial and varied research subject in its own right, and a exhaustive survey of such techniques is beyond the current scope. For more information, the authors Dezfouli et al. provide a detailed literature review of related resource allocation techniques in [52].

4.2.2.1.6 Association Control In traditional 802.11 networks, the distributed association control mechanism is strongest signal first (SSF), where each client takes a local decision as to which AP to connect to based on the RSSI of the probe responses and beacon frames. Also used is Least Load First (LLF) whereby APs broadcast their current load (represented using metrics such as the total number of competing stations present) via beacons which are then used by the client devices to inform the association control decision making.

For client steering, CUWN uses Cisco Compatible Extensions (CCX) with centralised control of RSSI thresholds to facilitate seamless handover between 802.11 APs. When the RSSI threshold from the AP reduces below the scan threshold, the client quickens the rate at which it scans for other infrastructure APs to ensure the fastest possible hand-over. Three different types of roaming scenarios are accommodated: (1) Intra-controller roaming, whereby handovers are performed by switching the associated AP defined in the database (2) Inter-controller Layer 2 roaming, which occurs when a client associates with an AP managed by a different SDN controller for the same subnet, and (3) Inter-controller Layer 3 roaming, which occurs when a client associates to an AP managed by a different SDN controller for a different subnet.

The Odin Mobility Manager (OMM) provide seamless handover by enabling fast LVAP migration between APs. A persistent TCP connection is maintained for each Odin agent so that transferring between different agents does not require that the connection is reestablished. The re-association delay is equal approximately to the time it takes for two messages to be sent from the controller to the current and new APs in order to remove and add the LVAP association. For a successful exchange, this is equal to the total RTT between controller and APs, which is of course dependent on the size of the network being managed. Other notable example of client steering include EtherFlow which uses a predictive handoff strategy to reduce the computational and communication overheads, and BigAP, which utilises the 25 channels available in the 5 GHz Wi-Fi band to construct disjoint collision domains via the use of a separate monitoring interface and client channel switching.

4.2.2.1.7 User-based Management Some works including Yiakoumis et al. [222], allow an agent application running on the end user device to define the service types used by certain traffic classes, which can then be implemented by the operator with the help of its global network view. The role of the agent is to translate the high-level policy provided by the user into the required low-level network directives. Participatory Networking features a similar mechanism where a user facing API is used to communicate the desired configuration information to the SDN controller. One of the main aims of the solution is the resolution of conflicting requests for concurrent user access, which was also targeted by Kumar et al. in [67]. The architecture uses an OpenFlow enabled switch and network slicing at the home gateway and provides a graphical user front end for configuring application traffic settings. FlowQoS [165] overcomes a limitation in Open vSwitch where per-flow QoS is not supported. The architecture also provides a graphical front end for users where application traffic preferences are defined. This information is then passed to the SDN controller which is responsible for both traffic classification and policy enforcement within the network. Another similar example is P2PQoS [201], which targets peer-to-peer video and audio streaming using SDN. Hyunwoo et al. [131] proposed a SDN-based solution to improve QoS for high definition video streaming coordinated by the content provider. The SDN controller chooses the best delivery node based on the global view of network state with routing performed using the Constrained Shortest Path First algorithm with MPLS. Metrics such as link bandwidth, packet loss rate, and jitter, as well as QoE related metrics such as video start up delay and buffering rate, are continuously monitored and when issues are detected the next best content delivery node is dynamically reassigned.

4.2.2.1.8 Provider-based Management Network slicing allows the splitting of a physical home network to provide isolation between homes and a more efficient overall operation. Slice management can be delegated to a third party which can remotely control interference management, security, and QoS between multiple adjacent home networks. This responsibility may be transferred to content provider such as Netflix and Disney+ to provide end-to-end QoS and QoE for high definition video streaming. Examples include Home Slice [222], which assigns to each slice resources such as bandwidth, CPU and memory, and forwarding table entries using FlowVisor, with an SNMP manager used to dynamically configure the home network APs. A similar solution is proposed by Wang et al. in [211]. In a dense urban deployment, the high number of access points and adjacent networks results in significant interference issues. In the proposed architecture, users associate to a virtual AP which is maintained for all physical AP connections. Virtual APs may be switched off in dense deployments to control interference and manage network congestion. However, the solution does not provide any method of information sharing and coordination between APs. BeHop extends the above solution for better resource allocation and user association. The architecture consists of a centralised controller, data plane APs, and a network monitoring and data collection entity, with an API for configuring AP channel and power usage. The POX [97] SDN controller is used with dual-radio OpenWRT routers and Open vSwitch with Wi-Fi extensions. Several works focus on managing interference between APs belonging to different service providers. For example, the COAP [142] solution uses uses a cloud-based Floodlight controller to manage channel allocation and interference using a modified wireless version of OpenFlow, without assistance from end users or operators.

4.2.2.2 SDN in Cellular Networks

This section provides an overview of topics and research challenges related to the application of SDN in wireless cellular networks. A number of different SDN research areas and applications are targeted in the literature, including centralised resource allocation, control plane and data plane scalability, and protocol evolution and technology agnostic device management. However, the following sub-sections focus only on resource allocation and protocol heterogeneity.

In cellular networks the Radio Access Network (RAN) provides connectivity over the last hop for wireless users and is responsible for functions such as MAC scheduling and handover management. The Core Network is used to connect individual RANs and handles mobility, connection establishment, and Quality of Service (QoS). Base stations host two types of gateway used by subscriber devices to connect to the Internet: (1) the Serving Gateway (S-GW) and (2) the Packet Data Network Gateway (P-GW), with both devices containing both control plane and data plane functionality. Example data plane functions include traffic monitoring and network access control, and control plane functions include user mobility, dynamic routing, radio resource allocation, and connection establishment. However, the tight integration between control and data planes in the traditional cellular design creates issues for the scalability of the network and the efficient management of resources. In particular, the need to efficiently manage available network resources has significantly increased over the last decade due to an exponential growth in demand for wireless connectivity. As such a number of optimisation techniques have been developed in order to increase the overall capacity of the network, including cell splitting, power control, and advanced channel allocation algorithms. Existing architectures also make it difficult to exploit the availability of multiple connectivity options, for example LTE, 802.11, and WiMAX. Presently users must manually switch between networks without proper and informed consideration of the current network state/load in the decision making process.

By decoupling the control and data plane at the P-GW, both the overall scalability and manageability of the network are improved. The decoupling also allows global resource allocation and interference management, and the integration of heterogeneous wireless technologies by the use of multiple concurrent control planes and a hardware abstraction layer. Example SDN applications in cellular networks include Quality of Service (QoS), traffic monitoring and billing, network slicing, network function virtualisation, load balancing, and user mobility.

4.2.2.2.1 Resource Allocation One of the main prospects of the application of SDN to wireless is the centralised allocation of resources. Cell splitting allows greater spatial re-use of frequency resources by replacing a single large base station with multiple smaller stations transmitting at a relatively lower power. This results in a denser cellular topology and a higher overall bandwidth, but leads to a higher number of overlapping cells and greater interference, which complicates network management.

As defined by Haque et al. in [87], the different SDN-based solutions that are available to assist in this regard may be split into three categories: (1) RAN-based solutions, which target resource allocation at the cellular network edge, (2) Core Network solutions, which target allocation in the core of the network and the Internet boundary, and (3) RAN and Core Network approaches, where resource allocation is performed concurrently and in coordination across both RAN and core parts of the network.

A well known example of SDN-based resource allocation in the RAN is SoftRAN [77], which features a centralised radio access control plane that abstracts multiple base stations into a single virtual base station, similar to the big switch abstraction in traditional wired SDN. Near-optimal allocation is performed from a 3-dimensional grid of time, space, and frequency resources. However, the latency between the controller and the data plane devices is a particular issue for wireless. Therefore SoftRAN uses a process called task distribution whereby some of the responsibility for local decision making is redistributed back into the data plane devices, while the controller maintains overall control. For example, uplink transmission frequency, transmission power, and handover management are performed centrally, while uplink frequency allocation is performed at the base station. Note that the controller is informed during the decision-making process by accessing the information database (called the RIB), which contains various operator settings and preferences, interference maps, and flow statistics.

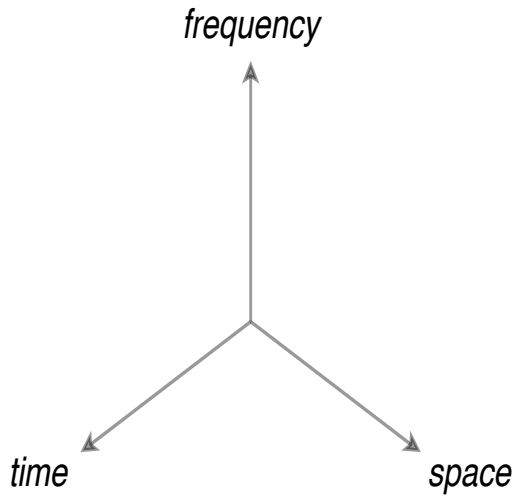

Figure 4.2: 3D Grid of Spectral, Temporal, and Spatial Resources

In the cellular core, SoftCell [94] decouples the control plane from the data plane at the Packet Data Network Gateway, and transfers the more complex functionality to access switches residing at the base stations. The main aim of the architecture is to connect User Equipments (UEs) to the Internet. The role of the controller is to implement the high-level policies as rules in firewalls and other middle-boxes in order to fulfil the user traffic requirements. Note that the controller implements the LTE signalling protocol between UE and the Mobility Management Entities in order to support entirely unmodified UEs. Packet classification and marking is done at the access switches using a local SDN controller and Open vSwitch. In order to increase the control plane and data plane scalability a multi-dimensional packet aggregation mechanism is employed which reduces forwarding table size.

Cellular SDN [110] provides unified RAN and Core Network resource allocation. Multiple application modules provide functions such as radio resource management, user mobility, subscriber tracking, and billing and policy enforcement. Furthermore the efficient sharing of resources is implemented using FlowVisor [161]. Local functions are delegated to data plane devices to remove the problems associated with the wireless control channel latency, for example functions such as traffic monitoring and changes in queue priority that do not benefit from controller involvement. Other local tasks including packet classification and compression. The authors in [115] extend Cellular SDN by installing an SDN agent application directly onto end user devices in order to improve QoE and enable direct device-to-device communication.

SoftAir [17] targets SDN in 5G cellular networks using a three-way virtualisation of the network, physical wireless medium, and access switches. Network virtualisation provides the proper isolation and insulation between network slices allowing resources to be shared effectively between multiple independent operators. Wireless virtualisation provides a low-level control over wireless resource sharing implemented by controlled scheduling. The switch-level virtualisation feature allows efficient bandwidth slicing and isolation using FlowVisor.

Other related solutions include MyNET [230], which provides a similar architecture to SoftAir with a hierarchical deployment of controllers designed to improve the overall responsiveness. CMaaS [220] provides a 4-tier controller hierarchy: (1) the UE controller which manages the user device and local attachments, (2) the Base Station controller which provides control over time-sensitive radio functions, (3) the RAN controller which manages a regional set of base stations, and (4) the Network controller which maintains a global view of network state and functions such as mobility, QoS, and routing. Note that in this architecture each controller is always constrained by the decisions of the upper controller.

4.2.2.2.2 Protocol Evolution Another branch of SDN works for cellular networks focuses on protocol heterogeneity and evolution.

OpenRadio [26] proposes a cellular core using modular protocol elements that allows the independent development of different control planes, which can easily be extended without modification or replacement of the underlying controller and data plane hardware. OpenRoads [113] uses SDN to bridge between heterogeneous link technologies such as LTE and 802.11 in order to increase the capacity and coverage, and also provide functions such as network slicing using FlowVisor, traffic isolation, and routing. OpenRAN [28] also relies on virtualisation to support multiple distinct wireless technologies and can also manage network storage and compute resources. SDWN is similar to OpenRAN and OpenRoads but also provides dynamic traffic configuration and RAN programmability. CROWD [21] also provides unified control of heterogeneous wireless technologies with efficient mobility and interference management through a hierarchical controller deployment. A regional controller performs long-term optimisations using information received from the local controller in the layer below.

4.2.2.3 SDN for Wireless Sensor Networks

The main challenge in wireless sensor networks is the limited resources of individual sensor devices, which is worsened at scale as the level of control traffic increases. Such networks usually consist of a group of constrained sensor devices with the common task of monitoring a particular phenomena, for example target tracking and habitat monitoring. Wireless sensor networks rely on a central node to monitor the network and collect the application data, and therefore the model used can be directly mapped to the controller-switch relationship in SDN.

The main benefit is that by freeing the sensor devices themselves from tasks such as routing and topology management, the overall power consumption and efficiency is improved. The controller assumes responsibility for such functionality which allows execution in a more efficient manner due to the globalised view of network state, while also acting as the sensor sink node in many implementations. The SDN approach also allows the management of heterogeneous hardware types under a single unified management platform. However, traffic patterns in wireless sensor networks differ significantly from traditional software defined networks, taking a reverse multi-path with aggregation at intermediate network nodes. In addition, wireless sensors are not equipped with a separate interface for the control connection, and therefore both data plane and control plane traffic is sent over the same unreliable wireless path.

4.2.2.3.1 Extending SDN for Wireless Sensor Networks Many research works extend the traditional SDN architecture to provide the various data structures and applications needed in wireless sensor networks. For example, Sensor OpenFlow [118] provides new classes of forwarding with reduced power consumption by virtue of a more efficient duty cycle control mechanism. While traditional SDN flows are address centric, Sensor OpenFlow provides support for low-power IP alternatives such as uIPv6 [57] while maintaining regular TCP/IP connectivity between devices and controller. To limit the amount of traffic over the combined control and data channel, the solution restricts the number of new flow requests that can be sent by the wireless device in response to dynamic changes in the network topology. After sending an initial request the device may not send another until the controller has responded or a timer elapses.

SDN-WISE [70] extends Sensor OpenFlow for stateful operation in wireless sensor networks supporting multiple controllers using a one-to-many controller proxy model. The SDN controllers and sensors are modified with the necessary data structures to manage the state of sensor devices performing local control tasks. For example routing is performed locally using the Topology Discovery protocol which is sent periodically to the controller to maintain the global network view. The controller also sends messages periodically back into the data plane to update sensors with the latest best next hop to the controller. Other related works include [146], which also uses a stateful approach when delegating tasks to sensors for wide-band spectrum monitoring. Controllers, collectors, and sensors form the three-tier architecture where the collectors assume the role of aggregating intermediate nodes.

4.2.2.3.2 Centralised Resource Allocation Another category of SDN work in wireless sensor networks is centralised resource allocation. An example is seen in SDWN [28] where the SDN controller is placed at the sink node. Special features include efficient duty cycling and flexible cross-layer routing optimisation. However, there is no attempt to mitigate the control plane traffic which is still required for establishing the end to end connectivity, and this is a particular problem for resource constrained wireless sensor devices. Smart [47] is a related architecture where the SDN controller is also placed at the sensor sink node, which is claimed to increase the overall energy efficiency, but again the mitigation of the control plane overhead is not considered.

4.2.2.3.3 Scalability The number of connected devices needed for many wireless sensor applications means that the use of a single controller becomes an issue as the network scales. Not only in terms of data collection, but also for the network control traffic. Multiple clusters of controllers may be used to alleviate the problem by splitting the control plane responsibility across multiple independent but communicating controllers and sink nodes. Examples of such works include Spotled [139] and Flow Sensor [149].

4.2.3 Overview of SDN-assisted Interface Bonding

This section provides an overview of the core concepts related to the main thesis research subject of SDN-assisted interface bonding, including the performance benefits, the main limitations, and open research challenges. In accordance with the main thesis research theme there is a particular emphasis on the bonding of wireless technologies such as 802.11.

4.2.3.1 Definition

SDN-assisted interface bonding is the technique of applying SDN principles to the wireless link aggregation problem. Such architectures centralise link aggregation functionality, for example bond scheduling and interface selection, in the SDN controller to exploit its global network state view and facilitate network-wide bond operation and automated policy configuration.

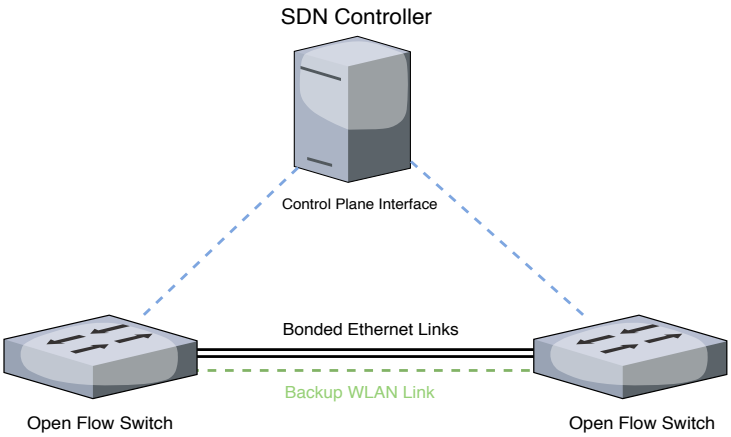

Figure 4.3: Centralised Bonding Functions in SDN-assisted Link Aggregation

4.2.3.2 Research Outlook

A number of SDN related interface bonding research works have been published in the last 4 years. Most of these target link aggregation using Multipath TCP at the Transport Layer using wired links such as Ethernet. Several open-source controllers, for example Ryu [43] and OpenDaylight [102], provide extended forms of the Link Aggregation Control Protocol for use in software-defined environments, but the functionality seen is still fairly simplistic, in particular there is no dynamic configuration of bond slaves according to the current achievable link performance. As a result there are many open research questions yet to be answered, especially at Layer 2 of the protocol stack and with respect to the scalability of such systems and their performance over highly contested and fast-fading wireless channels. A detailed literature of related SDN work is found later in the chapter.

4.2.3.3 Benefits

There are multiple benefits to an SDN-assisted approach to link aggregation which apply to both wired and wireless networks.

1. *Network-wide Bond Policy Configuration:* In an SDN-assisted bonding system, network administrators may express network-wide bonding policies in the management plane which filter down to switches and end-user devices in the data plane. For example, an administrator may forbid the use of bonding in certain parts of the network, prevent its use under certain traffic conditions, or otherwise limit its behaviour, such as allowing LACP or round robin mode only. This ensures consistent use of bonding across the network and avoids its problematic or wasteful application. Vendors may offer such functionality as a value added service which forms part of a larger SDN solution.

2. *Automated Bond Configuration:* Manual configuration of link aggregation between end hosts is time consuming, prone to error, and can lead to inefficient use of bonding. In SDN-assisted link aggregation environments, the SDN controller plays a key role in bond configuration by actively/passively acting in the bond creation process, for example by augmenting LACP, or by directly programming the bonding devices. The controller may be utilised to help select the most appropriate set of interfaces to be used based on its global view of network state, which greatly reduces the burden of manual configuration for engineers.

3. *Centralised Bond Scheduling:* Several recently published research works have demonstrated how centralised and dynamic bond scheduling can be used to increase overall bonding efficiency using MPTCP in wireless networks. For example, by calculating weighted round robin frame allocation ratios based on a global view of network state, the controller is able define the most appropriate scheduling to be used to maximise throughput, coordinate the use of bonding across the network, and facilitate advanced load balancing algorithms. Such architectures may greatly improve the operation of wireless interface bonding and increase the utilisation of available network resources.

4.2.3.4 Challenges

In addition to general challenges related to the individual research subjects of SDN and wireless link aggregation, there are several specific challenges with respect to SDN-assisted wireless interface bond as summarised in the points below:

1. *Dynamic Wireless Channel:* Due to the dynamic and shared nature of the wireless channel, the achievable data rates and delay of wireless network interfaces are in constant flux. The link monitoring and state information in the SDN controller that informs the interface selection unit and scheduler may quickly become outdated if the delay in the control path is too high or the wireless channel is too unreliable. Addressing this issue is key to a successful implementation of an SDN-assisted wireless interface bonding architecture.

2. *Control Plane Scalability:* The use of bonding leads to a higher number of flows traversing the network which creates an additional overhead at the controller in terms of the number of rules and flow-tables managed. This can lead to an increase in the overall control channel traffic and significant additional latency. In the worst case, this can result in memory exhaustion causing complete failure of the controller. How the performance of SDN-assisted wireless link aggregation solutions scale with respect to factors such as the flow load and the number of connected devices is still very much an open research question.

3. *Bond Interface Selection*: The current, static approaches to bond configuration can lead to inefficient and wasteful use of bonding, especially over wireless devices with heterogeneous delays. Multiple research questions exist around bond creation and slave selection such as the best combination of interfaces to use for a given set of links states, how often the bond configuration should be evaluated, and which performance metrics should be used in the decision-making processes.[1]

4. *SDN Bond Scheduling:* Traditional round robin scheduling performs poorly over links with heterogeneous delays, as demonstrated and analysed extensively in the preceding chapter research. The use of weighted round robin provides a possible alternative and the global view of the SDN controller is ideally situated to make scheduling decisions for bonding devices across the entire network. However, there are a number of unanswered research questions for SDN-assisted scheduling, such as how best to calculate the optimal frame allocation for a given set of states, whether scheduling updates should be periodic or event-based, and whether latency and scalability in the control plane is potentially a limiting factor in terms of the overall bond throughput performance.

[1]Note that addressing these challenges in particular is the main aim of the presented thesis research work.

4.2.3.5 Use Cases

There are several potential use cases and applications for SDN-assisted link aggregation in both wired and wireless networks, and a selection of which are described in the points below:

1. *Virtual Machine Migration:* In a software defined data-centre, link aggregation may used to facilitate very large data transfers, for example Big Data applications and database backups, but also to provide rapid virtual machine migration within or between physical data centre sites. The former may be of benefit during server fail-over by greatly reducing the time required for service restoration.

2. *Boosting Wireless Access:* In software-defined wireless environments, link aggregation may be used to increase data rates between a multi-homed mobile device and a multi-technology wireless AP by extending SDN to the communicating host, i.e. by installing Open vSwitch and other extensions directly onto the mobile device, which is of particular use when frequent bottlenecks occur at the wireless edge.[2]

3. *Re-deployable MPTCP Proxies:* SDN may be used in conjunction with Network Function Virtualistion (NFV) techniques to provide MPTCP proxies that are instantly deployable to any point in the network. The optimal proxy placement may be evaluated centrally by the SDN controller and the required virtual network functions deployed dynamically according to current network state.

[2]Note that this use case is the main focus of the thesis research work.

4.2.4 Related Work

This section provides a literature review of related SDN interface bonding research. Such works have largely focused on MPTCP, the multi-path enabled version of TCP, but some works also target link aggregation at Layer 2 of the stack using Open vSwitch. This survey provides a synthesis of currently published SDN related interface bonding research at all layers of the protocol stack categorised according to research application and considering both wired and wireless devices. To the best knowledge of the author, the following overview represents a first in the research literature.

4.2.4.1 MPTCP Link Aggregation in the Data Centre

One of the first research works exploring SDN-assisted link aggregation was published in 2015 by Duan et al. [55]. The work targeted large scale data transfers in data centre networks. An SDN-assisted application is proposed that dynamically assigns MPTCP sub-flows to intelligently selected network paths with the goal of avoiding potential packet re-ordering and loss. The system is comprised of a controller that performs sub-flow management calculations, and a monitor, responsible for dynamically adjusting the number of sub-flows. Simulations performed in ns-3 are used to demonstrate the value of the SDN-based proposition and results show a significant increase in the overall end-to-end MPTCP throughput. Note that the work does not actively prevent multiple MPTCP sub-flows traversing the same link - the proposed solution chooses which MPTCP sub-flows to add or remove based solely on the throughput demands made by the MPTCP end-point.

The authors Zannettou et al. explore path diversity in data centres using MPTCP-aware SDN [225]. As alluded to above, the use of MPTCP with Equal Cost Multipath Routing can lead to individual sub-flows utilising the same network path due to its random hash-based path selection. The SDN controller provides better utilisation over these paths by dynamically assigning sub-flows. The Floodlight-based controller consists of two components, a Topology Manager, responsible for obtaining the set of paths between hosts and presenting them to the second component, a Forwarding Module, which is responsible for selecting which path to assign to each individual sub-flow. The system was evaluated using the Mininet emulator with Linux MPTCP and Open vSwitch was used to emulate switches in a 8-ary FatTree data centre topology consisting of 128 hosts and 80 fabric switches. Test results show that by providing strictly disjoint sub-flow paths, the SDN controller was able to improve the MPTCP throughput up to 25%.

4.2.4.2 MPTCP Wireless Link Aggregation

Next, we provide an overview of published works closely related to the central thesis research subject. The main related work is the architecture and decision-making algorithm proposed by the authors Nam et al. described in detail in Section 4.2.4.2.1 below and independently evaluated later in the chapter. Other related SDN-assisted wireless bonding works are described in Section 4.2.4.2.2.

4.2.4.2.1 Nam et al. (2016) In the work presented in [130], the authors Nam et al. propose an SDN scheme for dynamic MPTCP path control over dual 802.11 WLAN links. A customised SDN agent application running on the MPTCP host communicates with an SDN-controller which tracks the capacity of available network links and dynamically chooses interfaces based on a global view of network state. The necessary flow-control, QoS, and routing configuration is then programmed by the SDN controller onto the host via the SDN agent application. The scheme uses a value called the bond throughput gap as the main decision making metric - if greater than 0.9 then a signal is sent to the controller to diagnose the network issue. If there is congestion in the local network, the controller may remove the offending sub-flow by sending a TCP RST segment. Sub-flows may be added using MPTCP MP_JOIN messages.

The solution was implemented using the POX SDN controller and tested on the Mininet network emulator. MPTCP version 0.9 was installed on the Ubuntu 14.04 operating system on both the MPTCP client and server. A hardware, TPLINK TL-WDR4300 802.11 AP was attached to the emulated network, and the MPTCP client was connected using two separate 802.11g and 802.11n wireless interfaces. The TL-WDR4300 used three dual-band antennas providing a theoretical maximum throughput of 300 Mbps over 802.11g and 450 Mbps over 802.11n.

During the first evaluation test, the behaviour of MPTCP was analysed under protocol-stress conditions by manipulating the 802.11 latency and capacity using Linux Traffic Control with Network Emulation and compared against the performance of the proposed scheme. The maximum achievable data rate of 802.11n was limited to 5 Mbps with a 1% packet loss rate. The achievable performance of the 802.11g link changed every 20 seconds between a 50 kbps data rate with a 5% packet loss rate and 2.5 Mbps data rate with a 3% packet loss rate. A packet capture of MPTCP packets was performed while the MPTCP client downloaded a file of 100 MB in size from a web server utilising both 802.11 links simultaneously. The MPTCP re-ordering queue size was estimated using the mptcptrace tool. The evaluation results showed that the proposed system reduced the reordering queue size as well as the overall data transfer times when compared to traditional end-point based MPTCP link aggregation. The traditional MPTCP approach took a total of 210 seconds to download the 100 MB file, while the SDN-based approach took 190 seconds, thereby providing a non-negligible percentage change of 9.52%.

The second evaluation test analysed the performance of the proposed scheme when using MPEG-DASH ABR video streaming using the DASH JavaScript player in a HTML5 web browser. The video server played a total of 596 seconds of video using 20 different bit rates between 45 kbps and 3.9 Mbps. To analyse QoE, the authors compared the number of bit rate switching events and the length of the re-buffering period measured for the proposed scheme against the same when using traditional MPTCP. The proposed scheme was found to experience 46 bit rate switching events and 25 seconds of re-buffering, compared to 68 switching events and 55 seconds of re-buffering measured with traditional MPTCP.

The authors defines a metric called the bond throughput gap, which is used in the proposed solution as the primary basis for the dynamic link selection decision-making:

$$Tput_{\Delta} = \frac{Tput_{11n}^{Max} - Tput_{11g}^{Max}}{Max(Tput_{11n}^{Max}, Tput_{11g}^{Max})} \tag{4.1}$$

where $Tput_{11n}^{Max}$ denotes the maximum achievable throughput of the 802.11n link and $Tput_{11g}^{Max}$ denotes the maximum achievable throughput of the 802.11g link. The normalised throughput gap metric provides a measure of the difference between the maximum achievable throughputs of the two 802.11 links as a ratio of the throughput of the link with the highest throughput.

The decision-making process of the SDN-controller application proceeds as follows. If other flows are sharing the paths, then the controller actively measures the achievable throughput of the two 802.11 links to obtain $Tput_{802.11g}$ and $Tput_{802.11n}$. If no other flows are using the paths then the application calculates the theoretical expected capacity using the MCS and RSSI on both links to obtain $Tput_{802.11g}^{Exp}$ and $Tput_{802.11n}^{Exp}$, which is calculated as:

$$Tput_{802.11}^{Exp} = Min(Tput_{802.11}^{WAN}, Tput_{802.11}^{LAN}) \tag{4.2}$$

where $Tput_{802.11}^{LAN}$ is calculated as:

$$Tput_{802.11}^{LAN} = Tput_{802.11}^{Max} - \sum_{i=1}^{N} Tput_{802.11}^{User_i} \tag{4.3}$$

The bond throughput gap threshold is set to a value of 0.9, based on the empirical data obtained from the hardware-based testing performed. Above this value the MPTCP throughput was severely degraded due to an excessively large re-ordering queue size. If the calculated value of the metric is less than 0.9, then both 802.11 links are used. Otherwise only the single path with the highest capacity is used.

However, unlike bonding at Layer 2, MPTCP operates end-to-end at Layer 4, which also allows the SDN application to exploit the availability of Layer 3 information, i.e. the multiple available paths through the network between MPTCP client and the server. By doing so, the scheme is able to determine whether the congestion is caused by a bottleneck in the WAN or whether it is caused locally in the 802.11 WLAN. The SDN application is then able to react by re-assigning the congested flow to a less congested path, and re-directing the flow to another, less loaded, content server, or by removing it altogether. A simplified version of the main decision-making algorithm is illustrated in the pseudocode shown below.

Algorithm 2: Nam et al. Algorithm for Dynamic MPTCP Path Control using SDN

1. INPUT PARAMETERS:

FlowID and FlowDB = Application ID and Flow Database

$Tput_{path_i}...Tput_{path_n}$ = Throughputs of paths i to n measured at the MPTCP client

$Tput^{Exp}_{path_i}...Tput^{Exp}_{path_n}$ = Theoretical max capacity of paths i to n calculated by controller

$Tput_{max}$ = Maximum throughput among available paths i to n

$Tput^{path}_{\Delta} = \frac{Tput_{max} - Tput_{path}}{Tput_{max}}$

$Tput^{threshold}_{\Delta}$ = Normalised throughput gap threshold of 0.9

2. DECISION MAKING ROUTINE:

Get $Tput_{path}$ and $Tput^{Exp}_{path_i}$ for available paths i to n

For available paths i to n

  If $Tput^{path_i}_{\Delta} \geq Tput^{threshold}_{\Delta}$ **then**

    Set TCP Receive Window Size on path i to zero

    Set a TCP RST segment over path i

  Else

    Send MPTCP MP_JOIN option to add path i

End

4.2.4.2.2 Other Wireless SDN Bonding Works Xu et al. [215] propose an SDN and NFV-based, multi-path wireless network architecture called MP-SDWN which supports seamless mobility, flow-level QoS, and high throughput. The SDN controller consists of individual Topology Manager, Slice Manager, and Client Manager components. The AP Daemon runs on physical access points and virtualises the high-level MAC functions for each network slice. It also maintains the Wireless Flow Transmission Tables programmed by the controller and obtains statistics from a Multi-connection Virtual Access Point (MVAP). To facilitate user mobility, a separate MVAP is maintained per user/client. Both TCP and UDP throughput is shown to remain stable during handovers due to the soft, virtualised handover method employed. The MVAP is also used to for multipath transmission at Layer 2, and while link aggregation for increasing overall data rates is described in the work as being supported, it is not considered as one of the use cases.

The authors Nguyen et al. [136] present a resilient and scalable OpenFlow control channel for software defined wireless access networks. The proposed solution uses multiple MPTCP links to increase control path availability and relies on a network calculus-based framework. OpenFlow traffic is transmitted to the SDN controller in one of two ways, either it is striped across all available links, or messages are duplicated by sending them over all available paths. In both cases the controller decodes only the message with the earliest arrival time and discards further duplicates. After the initial TCP handshake between controller and SDN device, a check is made for available MPTCP options. If enabled then a MPTCP bond is created and OpenFlow messages are subsequently transmitted across both available paths. The additional MPTCP negotiation does result in a small delay overhead at the start of the connection. However, the results show that the overall path delay is reduced and throughput increases, which improves the scalability and reliability of the control channel, allowing more flows to be serviced.

In [48], the authors propose a SDN-based solution allowing centralised bond scheduling with a weighted distribution of frames. The weighted round robin algorithm is implemented using the MPTCP parameter num_segment, which defines the static number of segments to send per sub-flow in the traditional MPTCP implementation. The scheduler performance was evaluated between a dual-interface, Wi-Fi and 4G smartphone on the client side, and a Netgear AC1900 AP on the network side connected to a WAN gateway. The SDN controller was located approximately 160 miles from the control network. Initial testing on the default MPTCP scheduler produced a median throughput of 34.8 Mbps. Subsequent tests using weighted round robin show a throughput improvement of approximately 7.8%. However, the results also show that in some cases, the assigned link weights were not fully respected. For example, with a weighting of 2:200 (Wi-Fi:LTE), approximately 0.99% of segments should be transmitted over the LTE link, but experimentally the ratio was measured at 0.9%. This is due to MPTCP always selecting the Wi-Fi link first which means that it is likely the final cycle ends before the LTE sub-flow gets used.

4.2.4.3 Open vSwitch Link Aggregation

To the best knowledge of the author, only one research work has been published targeting Link Layer interface bonding using Open vSwitch in software-defined environments. The authors Rocha et al. [157] propose an architecture allowing automated, scalable, and self adaptive link aggregation using SDN. Automatic bond creation is implemented as follows: when the controller receives notification of a new switch, all links are checked against those of neighbouring switches. Link aggregation is assumed whenever more than a single link is identified. Failed links are automatically removed from the bond and added if or when they come back up. Three scheduling methods are presented; hash table, traffic analysis, and virtual round robin. The hash-table algorithm is similar to LACP and takes a source/destination MAC address and source/destination IP address and then calculates a hash used to define the interface over which traffic from that particular flow is sent. The traffic analysis algorithm balances load by favouring links that are more lightly loaded. The proposed link aggregation architecture was tested in a virtual environment using Mininet, the Ryu controller, Open vSwitch, and OpenFlow protocol version 1.3. The topology used was a traditional 3-tier data centre architecture with core, edge, and aggregation layers and all interfaces were configured with a speed of 100 Mbps. With respect to fairness between bond links, the algorithms show a varying level of performance. The traffic analysis approach performed poorly in all tests. However, both hash table and virtual round robin algorithms were found to divide traffic equally between links. The work successfully implements a dynamic, SDN-based link aggregation architecture but several aspects of the work raise further research questions. For example, bond creation is performed automatically so there is no way of managing which links to use in the bond, and neither is there an increase in data rates for a single communication session.

4.3 Evaluating the State of the Art

This section independently evaluates the main related contemporary research work, the SDN-assisted dynamic link selection algorithm designed for MPTCP link selection using dual wireless interfaces by the authors Nam et al. in [130].

The core link selection decision-making algorithm and related performance metrics of the Nam algorithm are implemented in MatLab on top of the 802.11 interface bonding framework presented in Chapter 3. The algorithm is then evaluated for the same series of balanced and unbalanced applied load scenarios as used for the static round robin configuration in the previous study. The MatLab based implementation of the Nam algorithm is described and illustrated using pseudocode in Section 4.3.1.1, and the various simulation input parameters and applied load scenarios used are detailed in Sections 4.3.1.2 to 4.3.1.4. The simulation results are given as a function of the applied bond load and are discussed in Section 4.4 in comparison to the traditional static bond configuration. A detailed evaluation and overview is given in Section 4.5, where the limitations of the throughput-gap-based approach by Nam are explained and presented alongside example output from the 802.11 bonding simulations conducted.

4.3.1 Experiment Setup

This section describes the experiment setup and methodology used for the independent evaluation of the dynamic link selection algorithm by the authors Nam et al. in [130].

4.3.1.1 Nam Algorithm MatLab Implementation

First we describes the MatLab implementation of the throughput-gap-based interface selection algorithm by Nam and its adaptation for use at the Link Layer over a dual-link, point-to-point 802.1 bond, in order to model the type of kernel-based, packet-oriented interface bonding used by solutions such as the native Linux bonding module and the Red Hat Teaming Driver, as studied in the three independent research works by authors Jarasuriya [92], Shidik [171], and So et al. [180].

As described in the previous related work section, the dynamic link selection algorithm used by Nam works by calculating a metric called the bond throughput gap, a normalised value representing the difference in achievable throughput performance between the two bond links in proportion to the throughput of the fastest link. The results provided by the authors showed that the aggregate 802.11 bond throughput performance was particularly degraded whenever the measured throughput gap between links was greater than a value 0.9. Using this empirically obtained value as a threshold for removing problematic links, the authors implement the following decision-making algorithm: whenever the measured bond throughput gap exceeds the threshold of 0.9, the slower of the two bond links is removed from the bond, thereby reverting to the use of the fastest available single link[3].

Note that the algorithm was originally designed for use with MPTCP and end-to-end interface bonds traversing multiple intermediate devices such as routers, switches, and firewalls. In the implemented framework the algorithm performance is evaluated at Layer 2.5 using a point-to-point 802.11 interface bond as per the pseudocode below in Algorithm 3. In addition, a secondary function of the implemented solution is the diagnosis of unstable network links, which can under some circumstances be fixed using the next preferable route and end-to-path. However, this functionality is not modelled in the current simulations where the focus is firmly on the efficiency of the core decision-making algorithm and related performance metrics in relation to the aggregate bond throughput when used over heterogeneous 802.11 devices.

[3]For the dual-link 802.11 interface bond assumed in the simulations

The main pseudocode used in the MatLab implementation is shown in Algorithm 3 below. The algorithm input parameters are the respective throughputs of each individual 802.11 slave link, S_1 and S_2, as measured during the simulation. Similarly the output of the algorithm, BondInterfaces[], is a vector representing the newly assigned state of each link of the bond where 0 represents an inactive interface and 1 represents an active interface.

Working through each line of the presented code, for all possible combinations of applied loads used in the simulation a separate decision is made in each case as to which interfaces should to be attached to the interface bond. To do so the measured throughput of each link is passed to the $calcS_{gap}$ subroutine which then calculates the bond throughput gap based on the current link state exactly as per the Nam algorithm. In the *if* conditional statement, the value of S_{gap} is checked to see whether it exceeds the specified threshold of 0.9. If the value of S_{gap} is greater than 0.9, then the max() and min() subroutines are used to find the fastest and slowest links and the output vector of assigned slave devices is adjusted according by removing the slower link. If the value of S_{gap} is less than the empirically obtained threshold of 0.9, then *both* bond links are utilised.

Algorithm 3: Throughput Gap Based Link Selection Algorithm Pseudo-code

```
Data: S1, S2
Result: BondInterfaces[]
Sgap;
for all possible combinations of nodes do
    Sgap = calcSgap(S1, S2);
    if (Sgap > 0.9) then
        fast = max(S1, S2)
        slow = min(S1, S2)
        BondInterfaces[fast] = 1;
        BondInterfaces[slow] = 0;
    else
        BondInterfaces[0] = 1; BondInterfaces[1] = 1;
```

4.3.1.2 Simulation Input Parameters

In order to allow a direct comparison, the simulation input parameters used, including 802.11 types (11ac and 11ah), DCF parameters, and MCS settings are identical to those used in the evaluation of the traditional static link selection approach in Section 3.4 of Chapter 3.

4.3.1.3 Test Scenarios for Homogeneous 11ac-11ac Bond

First we present the tests for the homogeneous 11ac-11ac interface bond.

4.3.1.3.1 Nam Test 1A: Balanced Load Scenario The first Nam test for the 11ac-11ac bond is the balanced load scenario shown in Figure 4.4 below, which is similar to RR Test 1A in Chapter 3. At each iteration M frames are sent and the number of competing stations in each WLAN is then incremented simultaneously from $n = 1$ to $n = 100$. At the start of the simulation the algorithm uses the individual link states to evaluate the bond throughput gap and if exceeding the 0.9 threshold then the problematic bond link is removed. Metrics such as the bond throughput and bandwidth utilisation are calculated and appended to a series of N-sized output vectors.

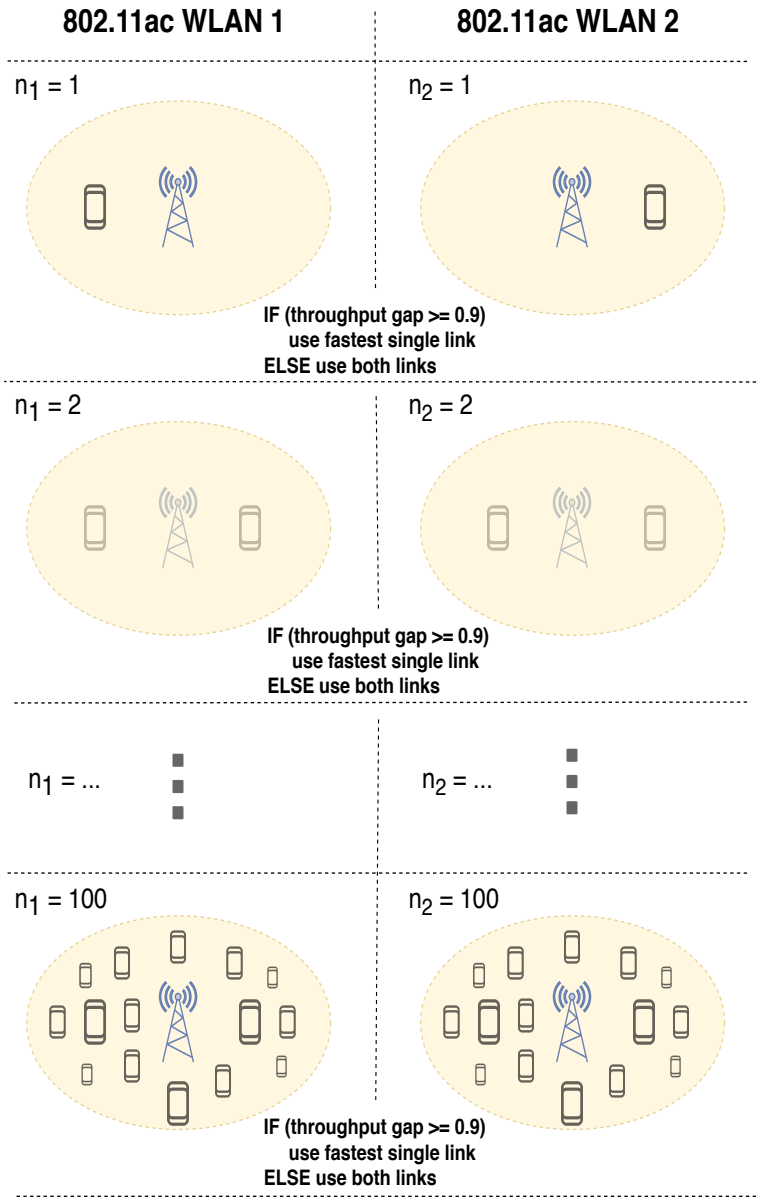

Figure 4.4: Nam Test 1A 11ac-11ac Bond Balanced Load Scenario

4.3.1.3.2 Nam Test 1B: Unbalanced Scenario (Low Load) The next Nam evaluation scenario is illustrated in Figure 4.5 below. The test is similar to RR Test 1B where in order to stress the aggregated bond throughput performance, a light load of $n = 1$ is applied on the second 11ac link while varying the load on the first 11ac link from $n = 1$ to $n = 100$. For each set of applied loads the Nam algorithm uses the individual link states to evaluate the throughput gap and if higher than the threshold of 0.9, then the slower 802.11 link is removed from the interface bond.

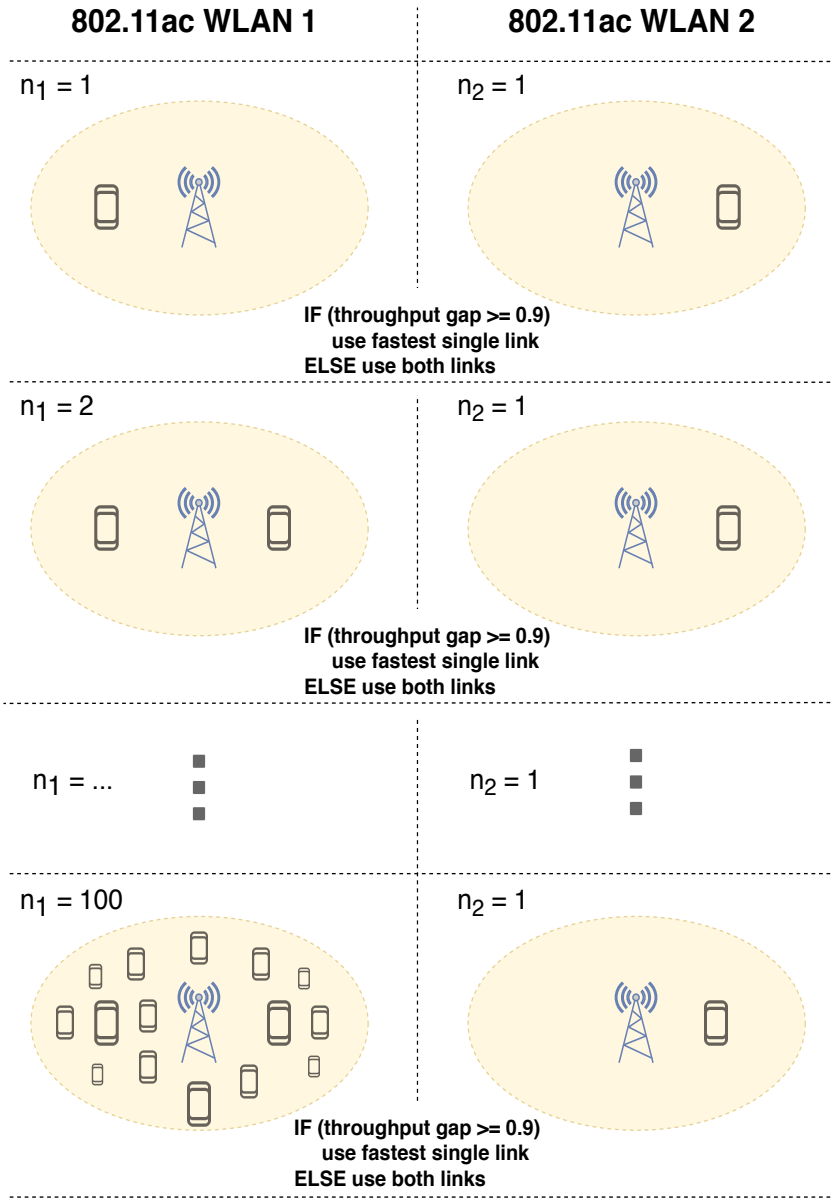

Figure 4.5: Nam Test 1B 11ac-11ac Bond Unbalanced Load Scenario (Low Load)

4.3.1.3.3 Nam Test 1C: Unbalanced Scenario (Moderate Load) The second unbalanced Nam test for the 11ac-11ac interface bond also stresses the bond throughput, this time with a moderate load on the second 11ac link of $n = 50$ while the load on the first 11ac link is incremented from $n = 1$ to $n = 100$. The test is shown in Figure 4.6 and is similar to RR Test 1C from Chapter 3. At each iteration M frames are sent and at the start of each test the algorithm evaluates the bond throughput gap based on the individual link states and a decision is made as to which links should be included in the interface bond.

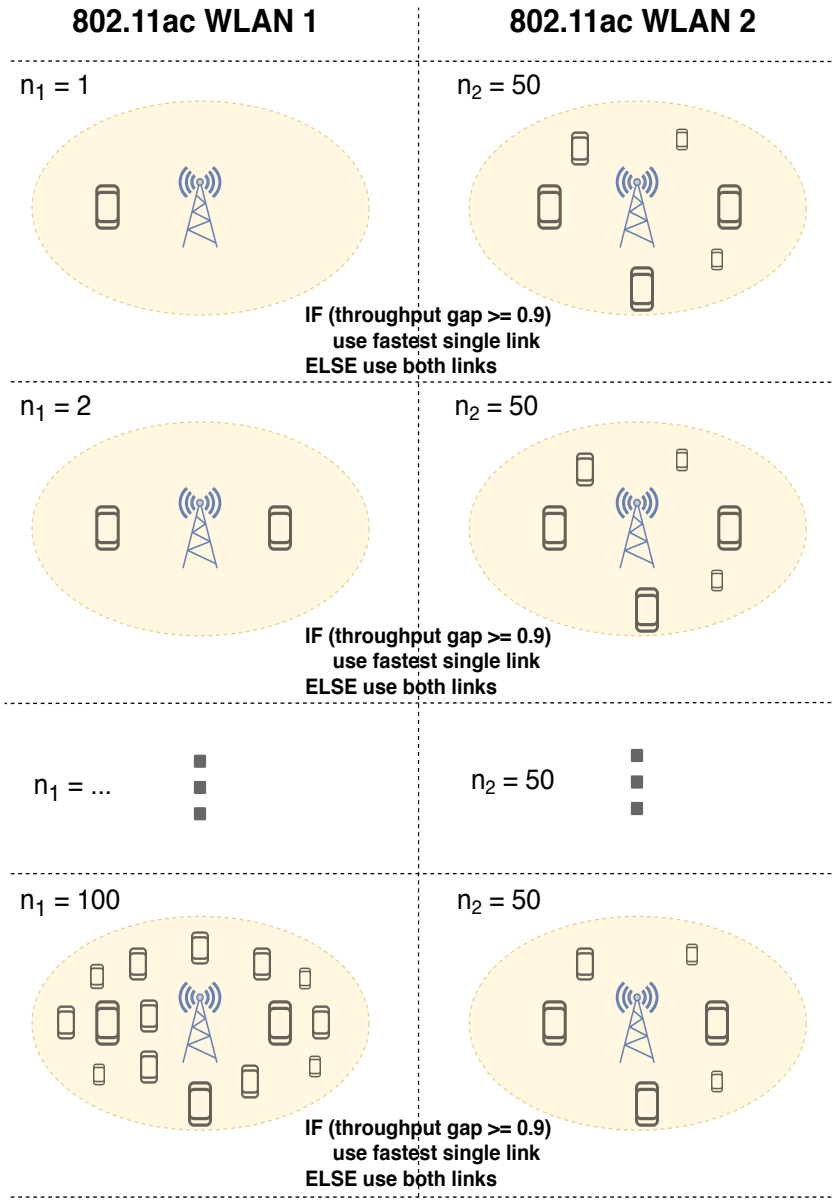

Figure 4.6: Nam Test 1C 11ac-11ac Bond Unbalanced Load Scenario (Moderate Load)

4.3.1.3.4 Nam Test 1D: Unbalanced Scenario (Heavy Load) The final unbalanced Nam test for the heterogeneous 11ac-11ac bond is illustrated in Figure 3.17 below. The static load on the second 11ac link is set to the maximum value of $n = 100$ competing nodes while increasing the load on the first 11ac link from $n = 1$ to $n = 100$. As for the other tests M total frames are sent and at the start of each sub-test the Nam algorithm evaluates the bond throughput gap using the individual link states and if greater than 0.9, then the problematic link is removed.

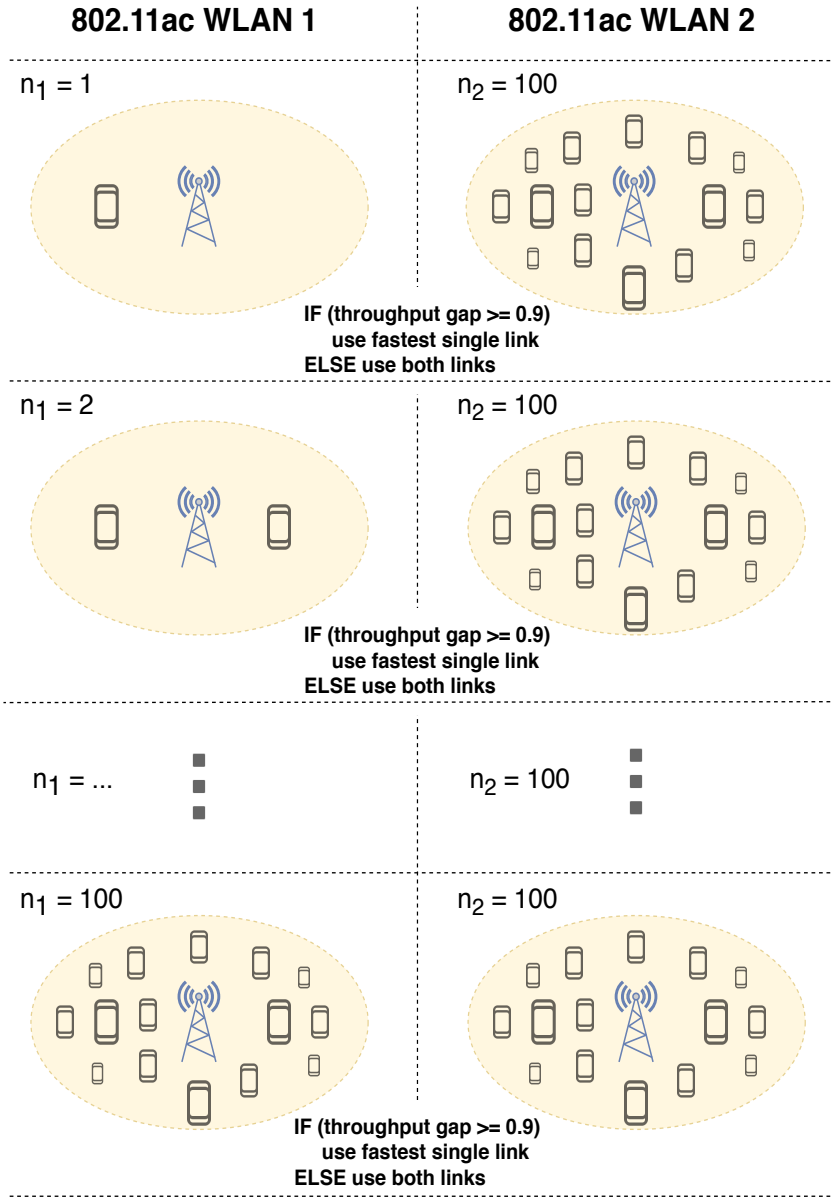

Figure 4.7: Nam Test 1D 11ac-11ac Bond Unbalanced Load Scenario (Heavy Load)

4.3.1.4 Test Scenarios for Heterogeneous 11ac-11ah Bond

Next we describe the similar tests used for the heterogeneous 11ac-11ah bond.

4.3.1.4.1 Nam Test 2A: Balanced Load Scenario The first Nam evaluation for the 11ac-11ah bond is the balanced applied load scenario shown in Figure 4.8 below. Again at each simulation the load on the 11ac and 11ah links is increased simultaneously by incrementing the number of nodes in each 802.11 WLAN from $n = 1$ to $n = 100$. For each combination of applied link load the bond throughput gap is evaluated and if greater than 0.9 the offending slave link is removed.

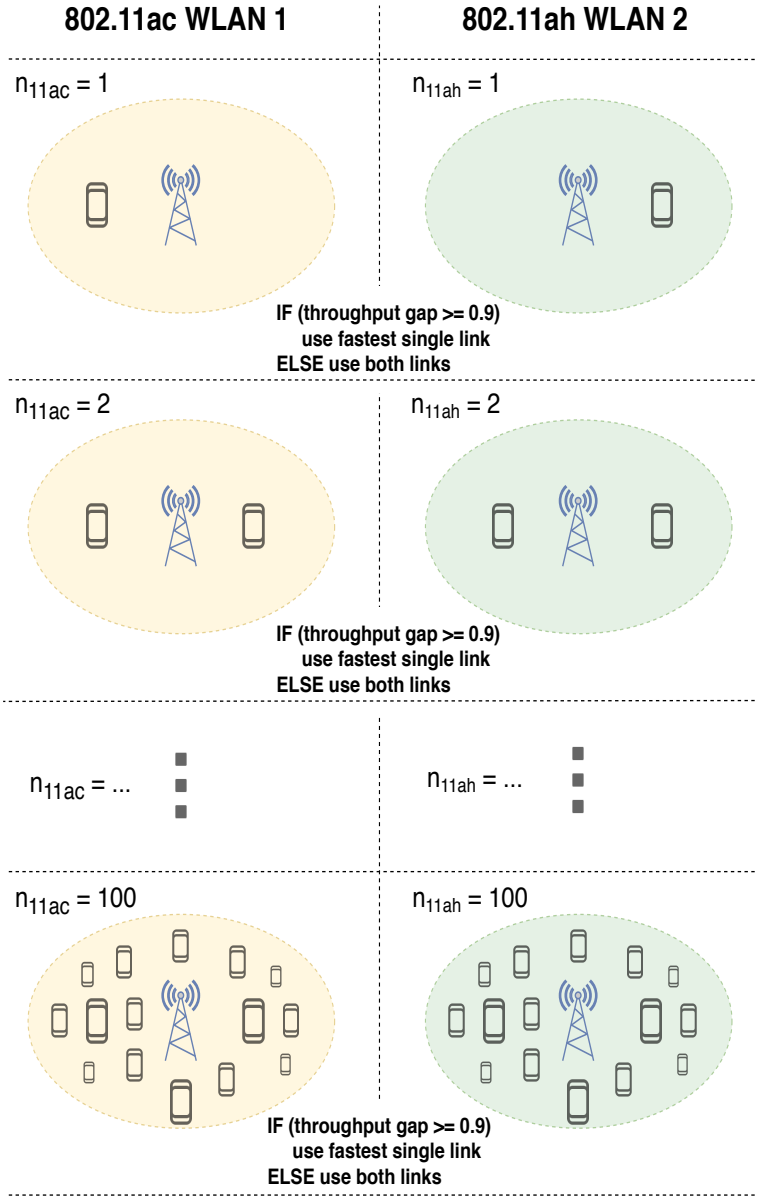

Figure 4.8: Nam Test 2A 11ac-11ah Bond Balanced Load Scenario

4.3.1.4.2 Nam Test 2B: Unbalanced Scenario (Low Load) In the first of the unbalanced load scenarios for the Nam algorithm and the heterogeneous 11ac-11ah interface bond, the applied load on the second 11ah link is kept constant at the low level of $n = 1$ nodes while the load on the first 11ac link is varied from $n = 1$ to $n = 100$. At every iteration M total number of frames are sent and at the start of each sub-test the Nam algorithm measures the bond throughput gap and if found to be higher than the 0.9 threshold then the problematic 802.11 link is removed.

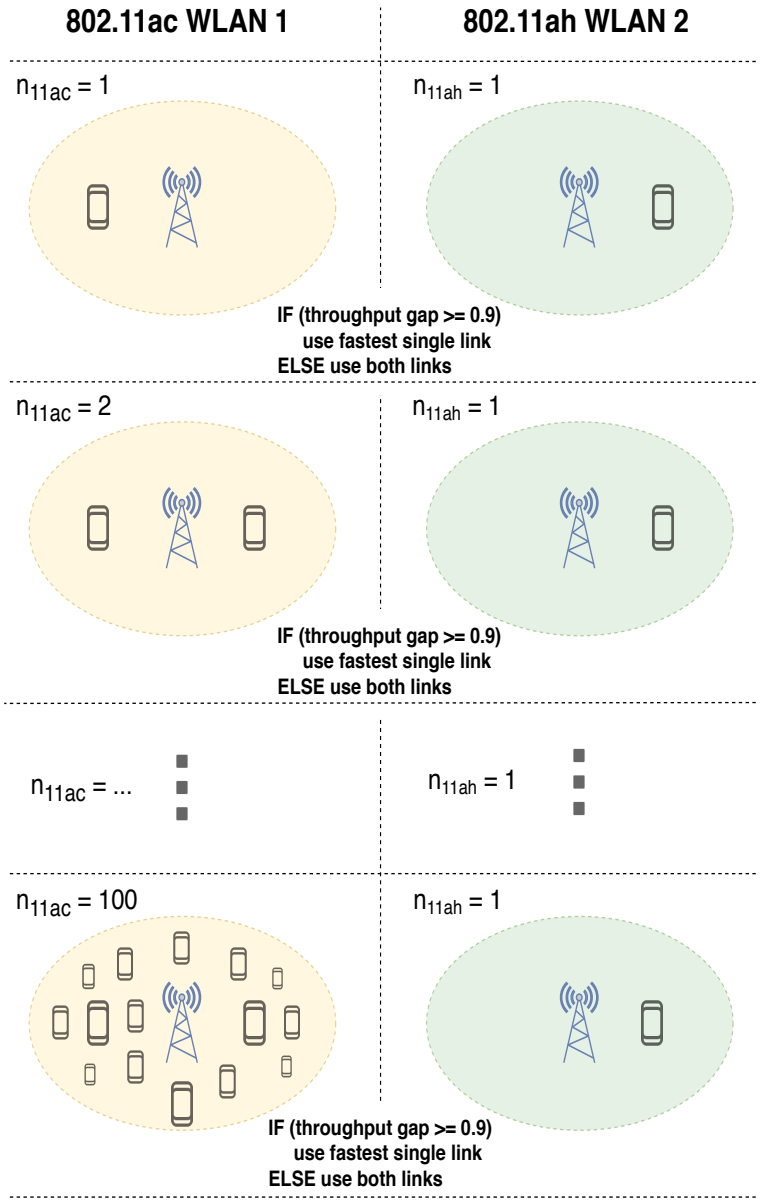

Figure 4.9: Nam Test 2B 11ac-11ah Bond Unbalanced Load Scenario (Low Load)

4.3.1.4.3 Nam Test 2C: Unbalanced Scenario (Moderate Load) The next Nam evaluation for the 11ac-11ah interface bond is the unbalanced load scenario shown in Figure 4.60 below. At each iteration the load on the 11ah link is kept constant at the moderate level of $n = 50$ while the load on the 11ac link is incremented from $n = 1$ to $n = 100$. Again for each applied load combination the Nam algorithm evaluates the bond throughput gap and if greater than 0.9, then the problematic link is removed.

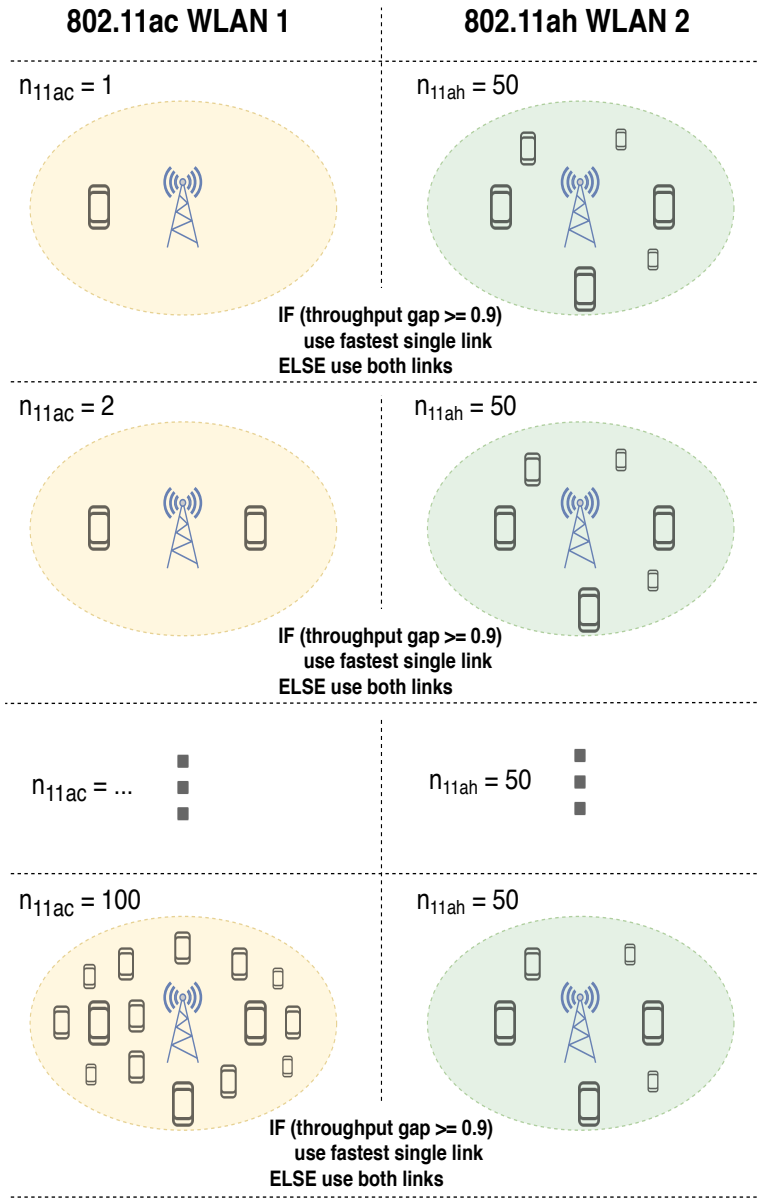

Figure 4.10: Nam Test 2C 11ac-11ah Bond Unbalanced Load Scenario (Moderate Load)

4.3.1.4.4 Nam Test 2D: Unbalanced Scenario (Heavy Load) The final Nam test for the heterogeneous 11ac-11ah bond is the unbalanced load scenario where the applied load on the 11ah link is kept constant at $n = 100$ competing nodes while the load on the 11ac link is varied from $n = 1$ to $n = 100$. The scenario is illustrated in Figure 4.7 below and as with the other tests for every applied load combination considered the throughput gap is measured and if higher than the 0.9 threshold the offending 802.11 slave link is removed from the interface bond.

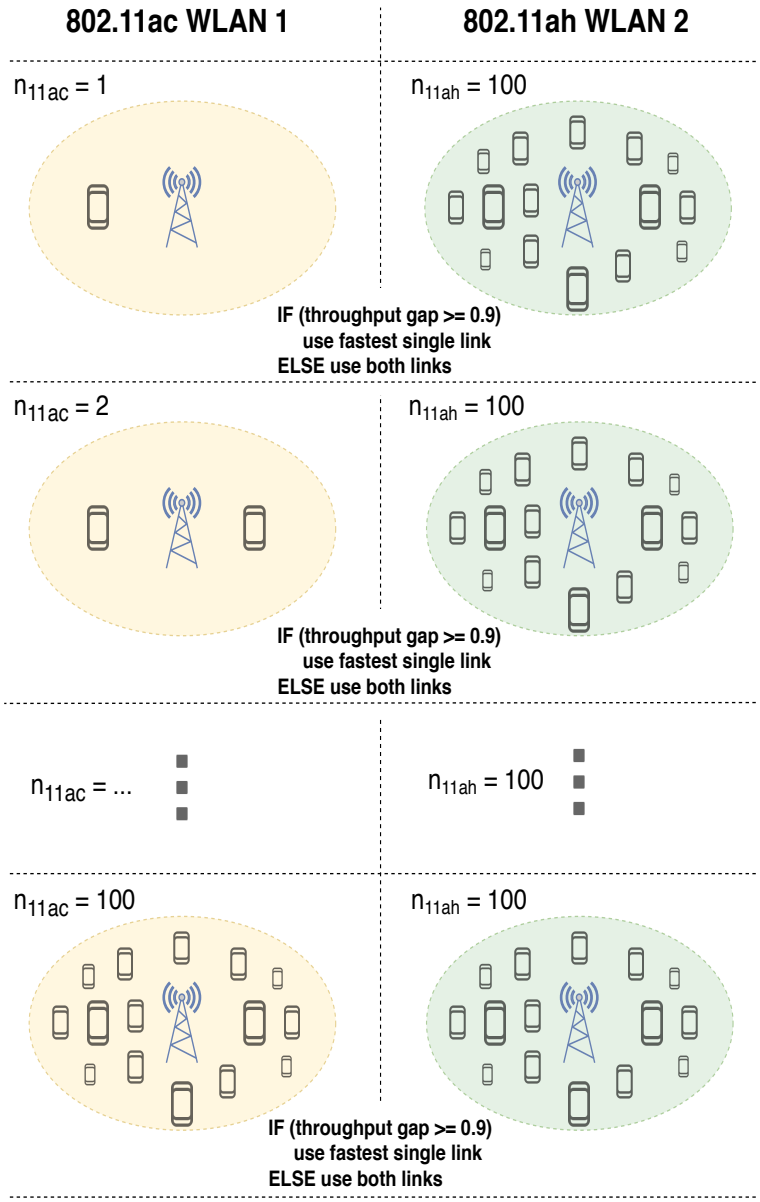

Figure 4.11: Nam Test 2D 11ac-11ah Bond Unbalanced Load Scenario (Heavy Load)

4.3.1.5 Comparison to Nam Methodology

In the benchmark work, Nam evaluates the aggregate throughput performance of an 11g-11n, round robin MPTCP interface bond using a small-scale hardware testbed where both 802.11 links are connected to a MPTCP server using a 100 Mbps Layer 2 Ethernet switch thereby creating a multi-hop, end-to-end interface bond at the Transport Layer.

Firstly, by virtue of being hardware-based the method used by the authors is fundamentally different compared to the MatLab based approach taken in the thesis research. In addition for the simulations presented both the homogeneous dual-11ac bond scenario and the heterogeneous 11ac-11ah bond scenarios are considered, while in the Nam work only the homogeneous 11g-11n case is considered. Furthermore the end-to-end Layer 4 bond used by the authors is in contrast to the single-hop, Layer 2 topology assumed in the developed framework. However in the Nam tests there is a single wired Ethernet segment with a speed of 100 Mbps that is four times the capacity of the fastest considered 802.11 throughput of 25 Mbps. This makes the 802.11 link the performance bottleneck and the main determining factor in the overall bond throughput performance over the end-to-end path.

Nam uses two separate 802.11g and 802.11n interfaces on the client device to connect to a multi-RAT, TP-LINK TL-WDR4300 access point capable of providing up to 300 Mbps and 450 Mbps respectively with signal strengths of -45 dBm and -38 dBm. MPTCP version 0.90 is used with Ubuntu 14.04 at both the client and the server end points to create the end-to-end interface bond. By contrast the bonding simulations presented in this chapter are MatLab based and agnostic with respect to the bonding protocol or technology used. This provides a greater focus on the impact of the 802.11 MAC mechanism on the aggregate bond throughput performance and allows an easier and more finely-grained control over factors such as channel contention levels and traffic generation during the simulation execution.

To generate the test traffic Nam placed in the server node a number of data files ranging in size from 10 KB to 5 MB, which were downloaded by the client node using both 802.11 links simultaneously. By contrast the MatLab simulations define a total number of M DCF frames to be sent over the each individual channel under the assumption of saturated arrival traffic, for which the individual tagged bond node transmits approximately M/N total frames. In the hardware-based tests by Nam the TCP server transmit buffer size is set to 10 KB (minimum), 85 KB (default), and 16 MB (maximum). The TCP client receive buffer size is set to 4 KB (minimum), 85 KB (default), and 6 MB (maximum). Separate bond throughput tests are then performed for each source-sink buffer size combination. In contrast the previous MatLab simulations do not model the link or bond transmit/receive buffers at the Transport Layer or anywhere else. Instead infinitely large buffer sizes at both sink and source nodes are assumed. In addition, the results of the test scenarios presented by Nam provide insight into the short-term performance of interface bonding using dual 802.11 links in small-scale network setting, while by contrast the MatLab simulation results give insight into the mean long-term behaviour of the interface bond at the

MAC layer in large networks consisting of a high number of competing wireless stations.

In a similar fashion to the MatLab experiments presented in the previous chapter, in order to stress the aggregate bond throughput performance the authors Nam et al. keep the achievable data rate on one bond link constant while the date rate on the other is varied. Instead of varying the achievable throughput by changing the number of competing DCF stations present as in the MatLab simulations, the desired unbalanced bond conditions are created by manipulating the 802.11 down-link throughput and latency artificially using Linux Traffic Control with Network Emulation. Three scenarios were considered:

- Scenario 1: The maximum throughput of the 802.11n link was kept constant at 5 Mbps, while the maximum throughput of the 802.11g link was varied between 2 Mbps, 1.5 Mbps, 1 Mbps, 0.5 Mbps, and 0.05 Mbps.
- Scenario 2: The maximum throughput of the 802.11n link was kept constant at 10 Mbps, while the maximum throughput of the 802.11g link was varied between 4 Mbps, 3 Mbps, 2 Mbps, 1 Mbps, and 0.1 Mbps.
- Scenario 3: The maximum throughput of the 802.11n link was kept constant at 25 Mbps, while the maximum throughput of the 802.11g link was varied between 10 Mbps, 7.5 Mbps, 5 Mbps, 2.5 Mbps, and 0.25 Mbps.

In the current work there are four basic test scenarios each defined by the level of applied load on each 802.11 link and given as the number of competing nodes present under the assumption of saturated network conditions. The four scenarios listed below were used in Tests 1A to 1D for the homogeneous 11ac-11ac bond and were then repeated for the heterogeneous 11ac-11ah bond in Tests 2A to 2D:

- Test A: The channel contention on Link 1 and Link 2 of the bond was incremented simultaneously from n = 1 to n = 100 to produce a balanced load across the bonded links.
- Test B: The channel contention on Link 2 was held constant at n = 1, while the Link 1 contention was varied from n = 1 to n = 100 to produce an unbalanced load.
- Test C: The channel contention on Link 2 was held constant at n = 50, while the Link 1 contention was varied from n = 1 to n = 100 to produce an unbalanced load.
- Test D: The channel contention on Link 2 was held constant at n = 100, while the Link 1 contention was varied from n = 1 to n = 100 to produce an unbalanced load.

The key differences between the research presented in this thesis and the work published by the authors Nam et al. in [130] are summarised in Table 4.1 below.

	Nam et al.	**Current Work**
Bonding Topology	End-to-end	Point-to-point
Bonding Layer	Layer 4	Layer 2
Experiment Type	Small hardware testbed	MatLab simulation
Bonding Technology	Multipath TCP	Protocol Agnostic
802.11 Standards	11g and 11n	11ac and 11ah
Bond Function	Dynamic Interface Selection	Dynamic Interface Selection
Bond Scheduling	Static Round Robin	Static Round Robin
Link Manipulation	Linux Traffic Control	Varying # Competing Nodes

Table 4.1: Comparison of Nam Experiment Setup vs Current Work

4.4 Nam Results Presentation

This subsection presents both the quantitative and qualitative results from the evaluation of the throughput-gap-based dynamic link selection algorithm proposed by Nam.

4.4.1 Homogeneous 11ac-11ac Bond

First we present the results for the homogeneous 11ac-11ac bond in Tests 1A-1D.

4.4.1.1 Nam Test 1A: Results

The throughput results for the dynamic link selection algorithm by Nam in the benchmark scenario in Test 1A are shown in Figure 4.12. The maximum theoretical bond throughput is plotted as the dotted black line and the achieved Nam bond throughput as the dashed red line. The overall mean throughput for the dual-11ac bond for all applied loads used was 2.478 Mbps while the mean theoretical maximum bond throughput was 2.4801 Mbps. During the test both throughput metrics decreased non-linearly as the applied load was raised and the achieved bond throughput was approximately equal to the maximum bond throughput for all applied load combinations. At the lightest load of $\{n_1 = 1, n_2 = 1\}$ both throughput metrics were measured at approximately 57.5 Mbps, but as the load was increased towards $\{n_1 = 100, n_2 = 100\}$ the outputs gradually reduced to minimum of 0.266 Mbps. Note that the size of the throughput difference between the two 11ac bond links was negligible compared to the individual absolute link throughputs with an overall mean value of 1.8405 kbps.

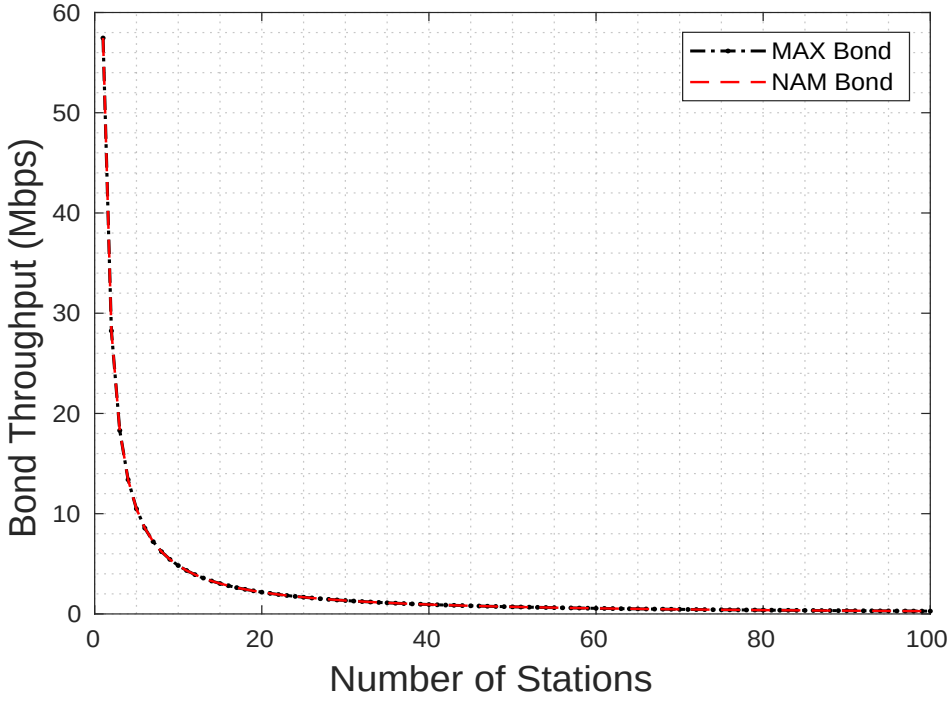

Figure 4.12: Test 1A Nam Algorithm Bond Throughput Results

The link bandwidth utilisation under the Nam algorithm for the benchmark case in Test 1A is given in Figure 4.13. The mean link utilisation for all applied load combinations considered was 99.898%. As seen below throughout the test the achieved utilisation under the dynamic link selection algorithm was close to 100% for all applied load combinations used with a minimum value of 99.1%.

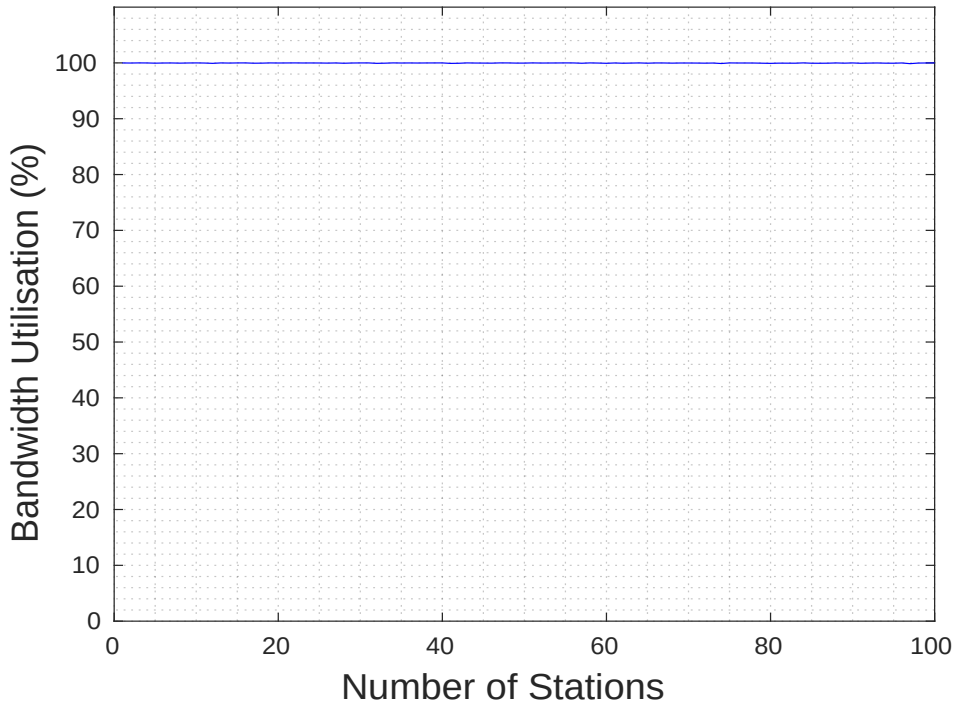

Figure 4.13: Test 1A Nam Algorithm Link Bandwidth Utilisation Results

The bond delay results under the Nam algorithm for the balanced load scenario in Test 1A are given in Figure 4.14. The overall mean 11ac-11ac bond delay was 50256 μs. The bond delay and the 11ac Link 1 delay were equal for all applied load combinations considered[4]. During the test the bond delay increased non-linearly from a minimum of 0.00086 μs at $\{n_1 = 1, n_2 = 1\}$ to a maximum of 117000 μs at the heaviest prescribed load of $\{n_1 = 100, n_2 = 100\}$. Note that the overall mean delay difference for all applied load combinations was 105.65 μs and therefore negligible compared to the absolute delay of each individual bond link.

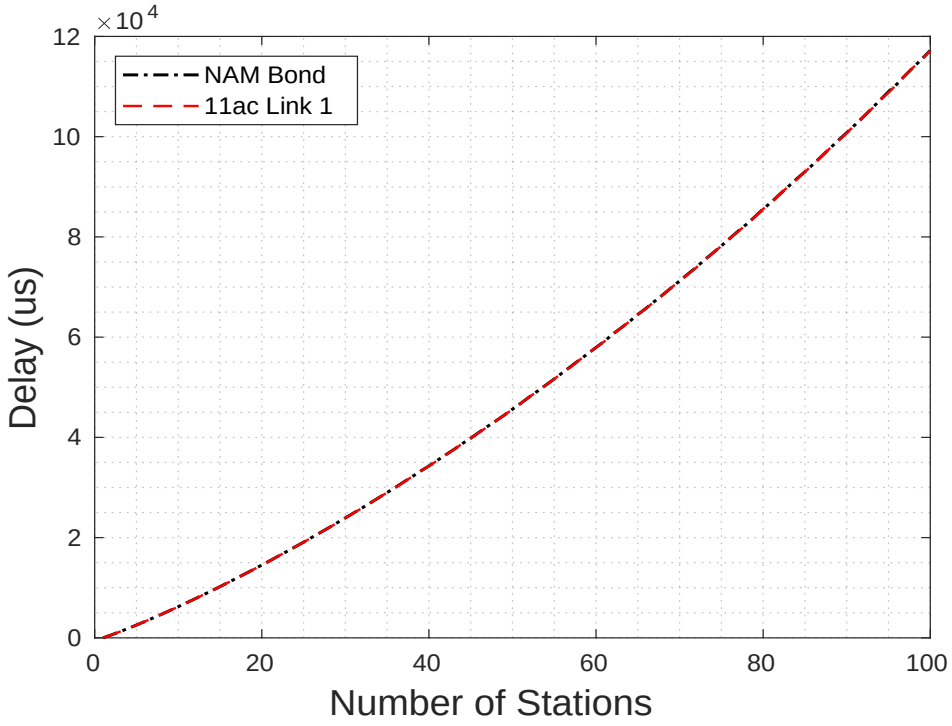

Figure 4.14: Test 1A Nam Algorithm Bond Delay Results

[4]The 11ac Link 2 delay was also equal to the 11ac Link 1 delay throughout but is not shown in the above graph for clarity

4.4.1.2 Nam Test 1A: Discussion

The main observation from the benchmark case in Test 1A was that the dynamic Nam algorithm throughput for the homogeneous 11ac-11ac bond used was equal or very close to the theoretical maximum bond throughput for all applied load combinations from $\{n_1 = 1, n_2 = 1\}$ $\{n_1 = 100, n_2 = 100\}$. In fact the performance of the dynamic link selection algorithm was identical to static round robin with both achieving an overall mean throughput of approximately 2.47 Mbps and mean bandwidth utilisation of 99.625%. The equality of performance is seen in Figure 4.15 below where the Nam throughput is plotted as the dotted black line and the static bond throughput as the solid red line.

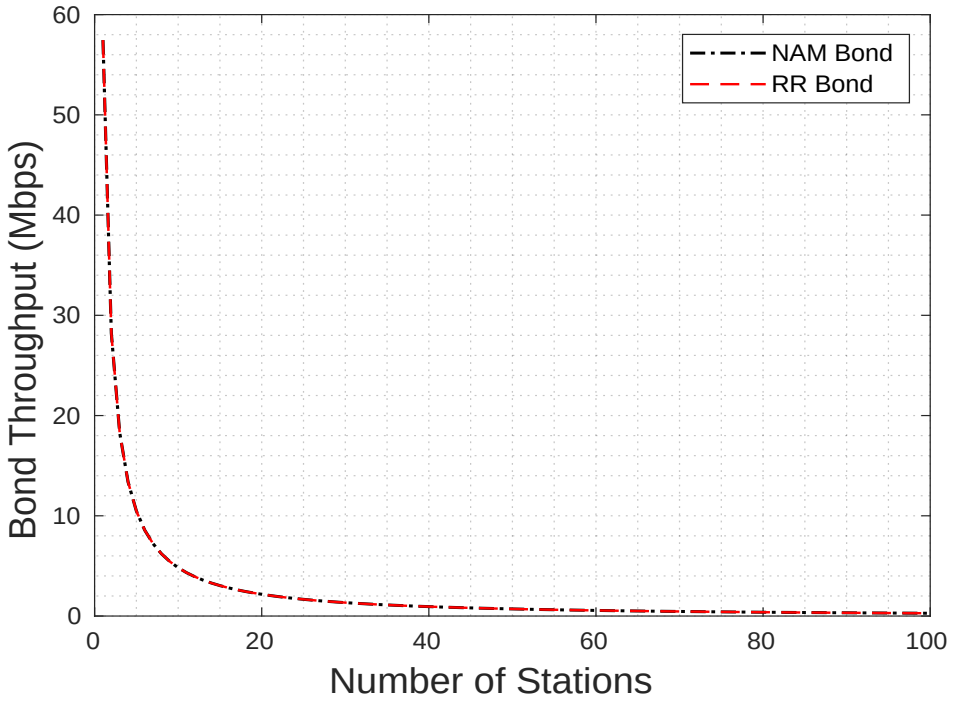

Figure 4.15: Test 1A Nam vs RR Bond Throughput Comparison

In Test 1A the Nam algorithm did not react under the balanced load conditions used in the test. The default state of the algorithm is to utilise both 11ac slaves in round robin mode unless the bond throughput gap is equal to or greater than the 0.9 threshold. As shown in Figure 4.12 and from the previous experiment in Chapter 3 the static round robin throughput is maximised in Test 1A due to the combination of homogeneous link data rates, identical DCF parameters, and balanced applied load resulting in a fully balanced interface bond at all times. Under this scenario the static round robin approach uses 100% of the available link bandwidth capacity, as also shown for the current test in Figure 4.13 above. The resulting bond throughput gap was therefore negligible and as desired the Nam algorithm selected both 11ac slave links at all times.

4.4.1.3 Nam Test 1B: Results

The bond throughput results for the Nam interface selection algorithm under the unbalanced load scenario in Test 1B are presented in Figure 4.16 below. The maximum theoretical bond throughput is plotted as the solid black line and the achieved Nam bond throughput as the dashed red line. The mean bond throughput for all applied bond loads considered was 29.3289 Mbps. Both throughput metrics decreased non-linearly as the applied bond load was raised. Furthermore the achieved bond throughput was lower than the maximum theoretical throughput for all applied load combinations considered with the exception of the initial applied load of $\{n_1 = 1, n_2 = 1\}$ where both were approximately equal at 57.45 Mbps. Between applied loads of $\{n_1 = 1, n_2 = 1\}$ and $\{n_1 = 8, n_2 = 1\}$ the achieved Nam bond throughput decreased to 27.99 Mbps and the maximum throughput to 31.84 Mbps. As the load was incremented to $\{n_1 = 9, n_2 = 1\}$ the achieved bond throughput changed from its previous downward trajectory and increased to 28.73 Mbps and remained constant as the load was further raised towards $\{n_1 = 100, n_2 = 1\}$. During the same range the the maximum theoretical bond throughput continued its previous non-linear downward trend to reach a minimum of of 28.98 Mbps. The overall mean link throughput difference for all applied load combinations considered was 1.5502 Mbps.

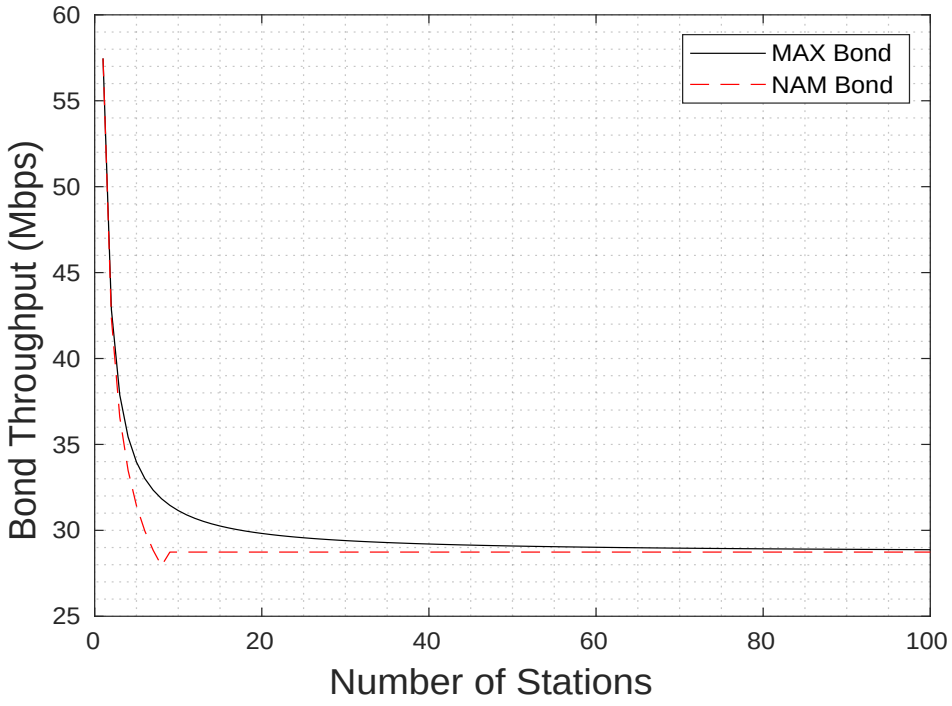

Figure 4.16: Test 1B Nam Algorithm Bond Throughput Results

The link bandwidth utilisation results for the dynamic link selection algorithm by Nam in Test 1B are shown in Figure 4.17 below. The mean link utilisation for all applied load combinations considered in the test was 99.507%. As seen from the graph for applied loads between $\{n_1 = 1, n_2 = 1\}$ and $\{n_1 = 8, n_2 = 1\}$ the resulting efficiency decreased non-linearly from an initial value of approximately 100% to a minimum of 87.91%. Then as the load was incremented to $\{n_1 = 9, n_2 = 1\}$ the output increased again suddenly to 100% where it remained as the load was further increased towards $\{n_1 = 100, n_2 = 1\}$.

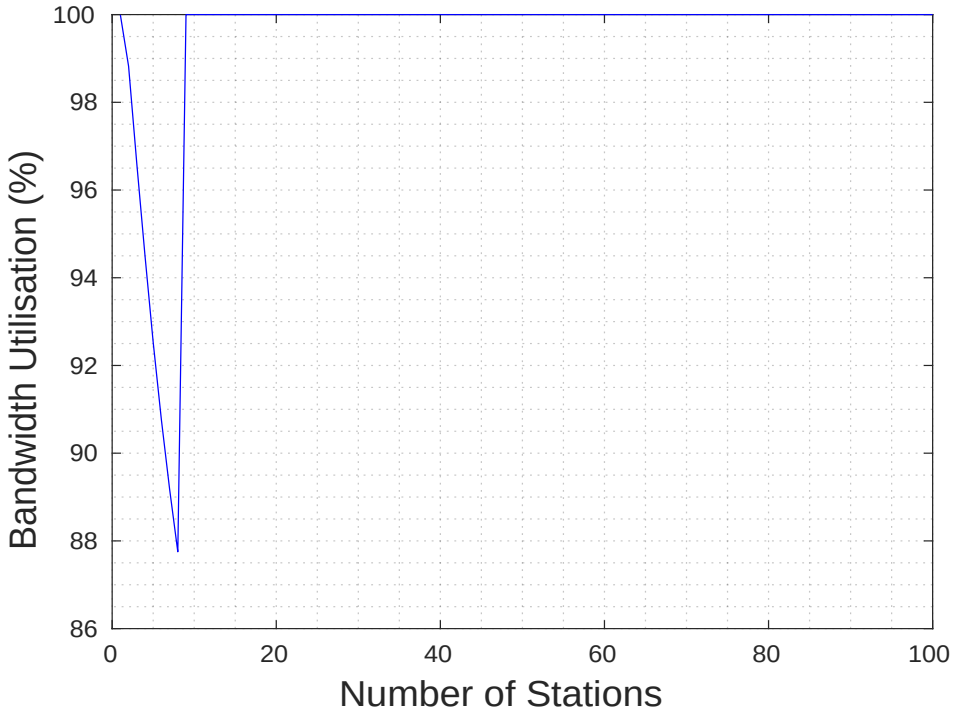

Figure 4.17: Test 1B Nam Bond Link Bandwidth Utilisation Results

The bond delay results for the Nam algorithm in Test 1B are shown in Figure 4.18 below. The overall mean bond delay for all applied load combinations used was 181.6635 μs. During the test the bond delay followed the Link 1 delay throughout and increased from an initial value of 0.0023 μs at $\{n_1 = 1, n_2 = 1\}$ to approximately 4696 μs at $\{n_1 = 8, n_2 = 1\}$. Then as the load was incremented to $\{n_1 = 9, n_2 = 1\}$ the delay output suddenly decreased to its previous value of 0.0023 μs where it remained as the load was further increased towards $\{n_1 = 100, n_2 = 1\}$. The mean delay difference for all applied load combinations used was 181.6612 μs.

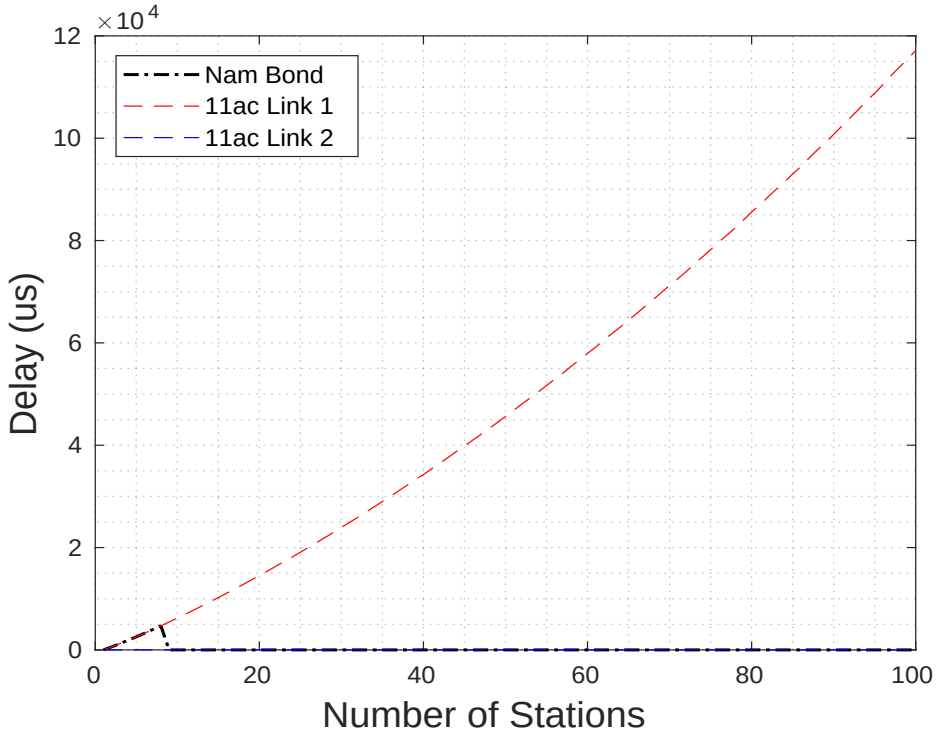

Figure 4.18: Test 1B Nam Bond Delay Results

4.4.1.4 Nam Test 1B: Discussion

The main result from Test 1B what that the dynamic Nam approach outperformed the traditional static round robin method by 23.93% in terms of mean achieved bond throughout and also increased link bandwidth utilisation by 53.02%. The overall mean bond throughput for static round robin was 19.75 Mbps while the achieved mean under the dynamic Nam algorithm was 29.3289 Mbps just 2.2% lower than the theoretical maximum mean bond throughput of 29.97 Mbps. The mean link throughput utilisation for Nam was 99.625% compared to just 65.03%. The achieved bond throughput under Nam is plotted alongside the static round robin throughput and the maximum theoretical bond throughput in Figure 4.19 below. The difference in behaviour between the dynamic and static link selection methods is clearly observed between applied loads of $\{n_1 = 9, n_2 = 1\}$ and $\{n_1 = 100, n_2 = 1\}$. As the applied load was increased within this range the round robin throughput reduced non-linearly to a minimum of approximately 14 Mbps while the Nam algorithm throughput remained constant at 28.73 Mbps.

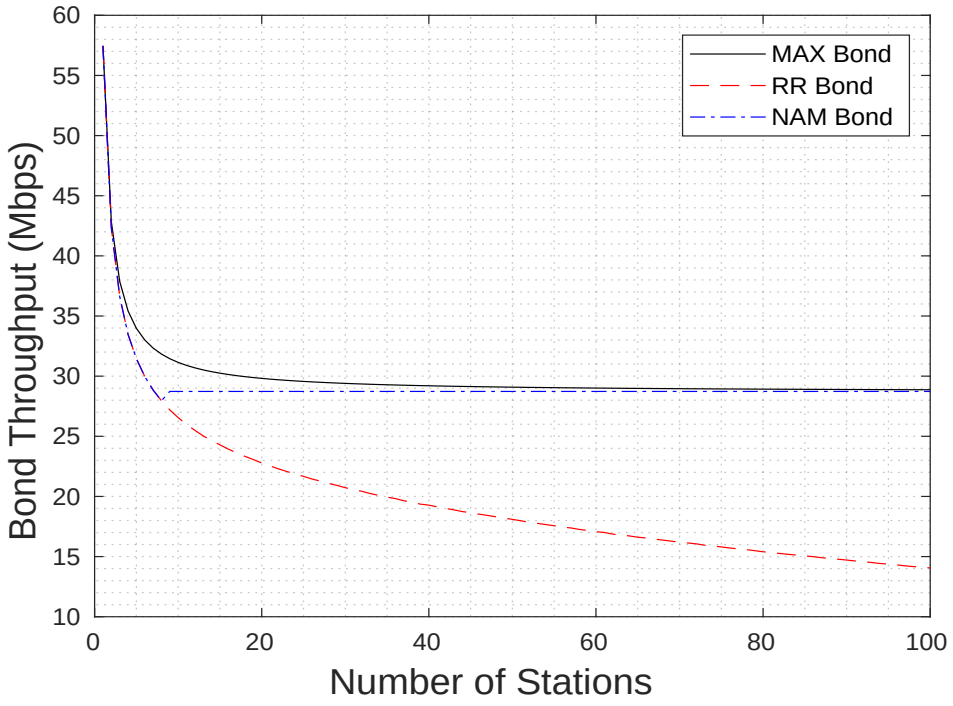

Figure 4.19: Test 1B Nam vs RR Bond Throughput Comparison

In Test 1B we saw the Nam algorithm react to the bond throughput gap threshold being exceeded by dynamically removing the slower of the two slave 802.11ac links. At $\{n_1 = 1, n_2 = 1\}$ both 11ac devices were in use and their individual throughput performance was approximately equal. When the applied load was increased the dual-11ac bond became increasingly unbalanced such that the bond throughput gap exceeded the 0.9 threshold at an applied load of exactly $\{n_1 = 9, n_2 = 1\}$ as seen in Figure 4.20 below. The dynamic selection algorithm responded to the change in link state by removing the slower 11ac slave in order to avoid the wasteful use of bonding and increase the mean bond throughput and mean link utilisation. As the applied load was increased towards the heaviest considered level of $\{n_1 = 100, n_2 = 1\}$ the throughput gap continued to increase to a maximum value of 0.9952 and did not reduce again below the specified threshold.

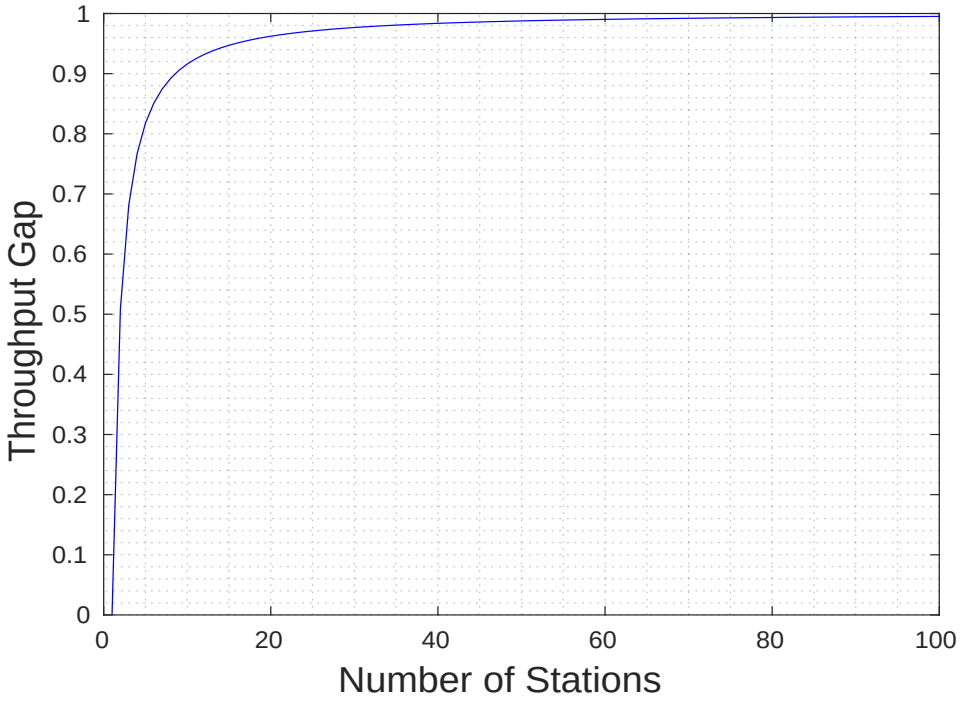

Figure 4.20: Test 1B Nam Algorithm Bond Throughput Gap Results

4.4.1.5 Nam Test 1C: Results

The throughput results for the dynamic link selection algorithm by Nam and the unbalanced scenario in Test 1C are shown in Figure 4.21. The maximum theoretical bond throughput is plotted as the solid black line and the achieved Nam bond throughput as the dashed red line. The overall mean throughput for all applied load combinations considered in the test was 1.4754 Mbps. Both metrics displayed a non-linear decreasing output as the applied bond load was increased. At the initial load of $\{n_1 = 1, n_2 = 50\}$ the maximum throughput was measured at 29.05 Mbps and the achieved Nam throughput at 28.7 Mbps. As the load increased towards $\{n_1 = 7, n_2 = 50\}$ both metrics decreased non-linearly while the difference between them remained approximately the same. Notably as the load incremented to $\{n_1 = 8, n_2 = 50\}$ the maximum throughput decreased to 3.473 Mbps while the achieved throughput reduced more significantly to 2.56 Mbps. As the load was further increased the difference between the achieved throughput and the maximum throughput became smaller and eventually reduced to 0 Mbps at $\{n_1 = 50, n_2 = 50\}$ where both metrics were measured at approximately 0.7 Mbps. Then as the applied load was increased further still the difference between the two outputs started to increased again while both continued their general downward trends. At the heaviest considered load of $\{n_1 = 100, n_2 = 50\}$ the achieved Nam bond throughput was decreased to 0.411 Mbps and the maximum theoretical bond throughput to 0.49 Mbps. The overall mean link throughput difference for all applied load combinations used was 0.32843 Mbps.

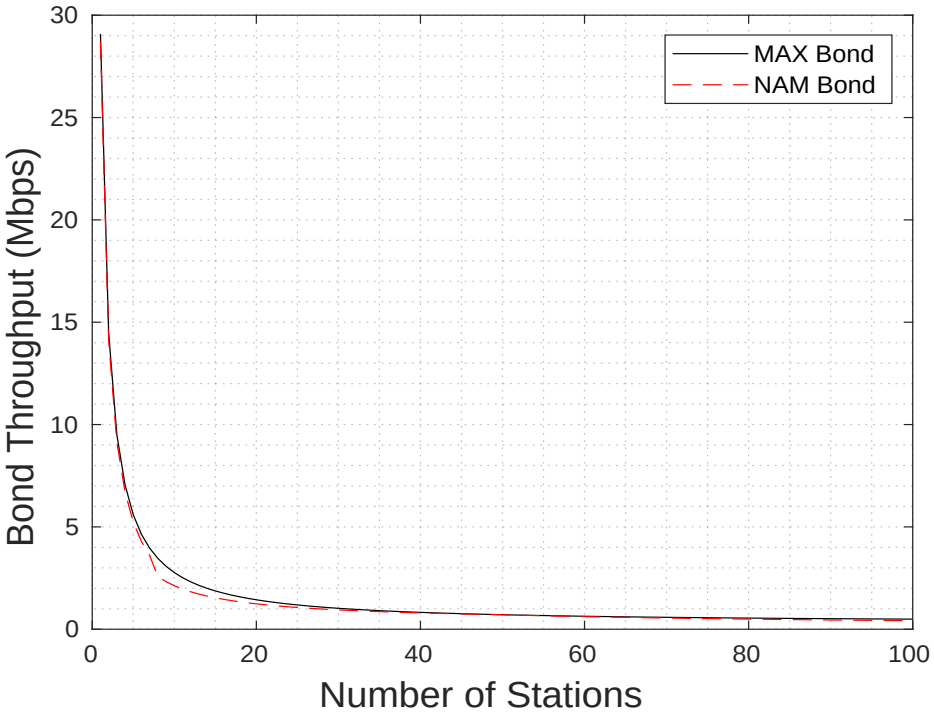

Figure 4.21: Test 1C Nam Algorithm Bond Throughput Results

The link bandwidth utilisation for the Nam algorithm in Test 1C is shown in Figure 4.22. The overall mean link utilisation for all applied load combinations considered was 91.932%. As observed the utilisation was approximately 100% for applied loads between $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 7, n_2 = 50\}$ but as the load was incremented to $\{n_1 = 8, n_2 = 50\}$ the efficiency reduced significantly to a minimum of 74%. As the load was further increased the achieved utilisation increased again to a maximum of 100% at exactly $\{n_1 = 50, n_2 = 50\}$ before then decreasing gradually to 84% at $\{n_1 = 100, n_2 = 50\}$.

Here, for applied loads between $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 7, n_2 = 50\}$ only one link of the bond was active, but as the load incremented to $\{n_1 = 8, n_2 = 50\}$ the bond throughput gap fell below the threshold of 0.9 and the Nam algorithm responded by re-activating the second 802.11 link, which had the effect of suddenly decreasing the bandwidth utilisation of the bond. The utilisation then gradually increased to a maximum 100% at $\{n_1 = 50, n_2 = 50\}$ as the performance of the two individual links became comparable, and then reduced as the load further increased to $\{n_1 = 100, n_2 = 50\}$ and the performance disparity between links increased.

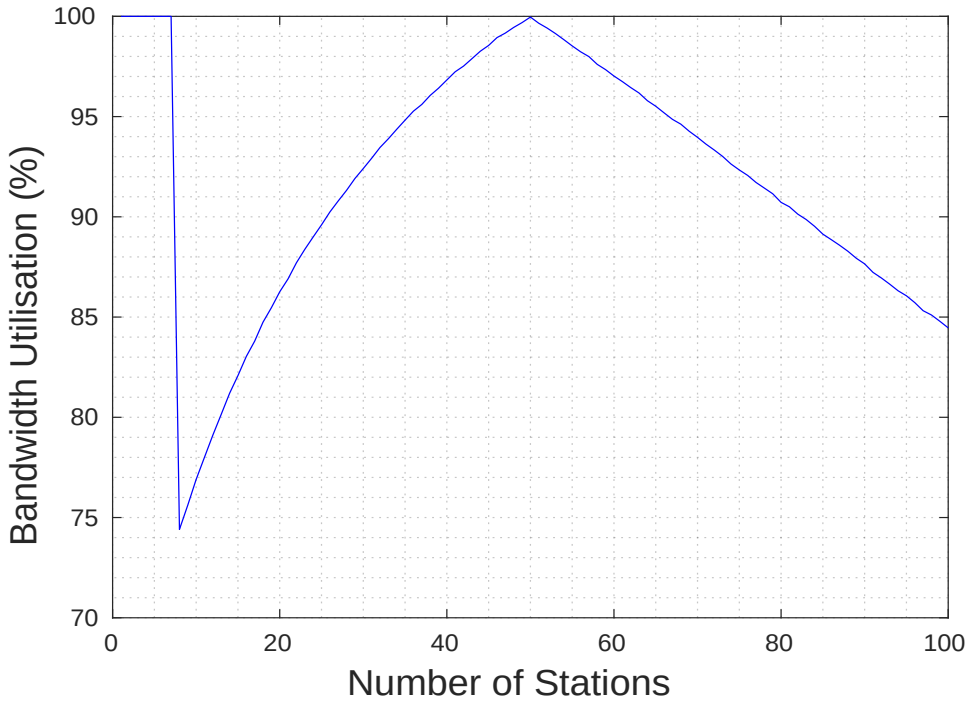

Figure 4.22: Test 1C Nam Algorithm Link Bandwidth Utilisation Results

The Nam algorithm bond delay results for Test 1C are shown in Figure 4.23 below. The overall mean bond delay for all applied load combinations considered was 62142.1458 μs. For applied loads between $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 7, n_2 = 50\}$ the bond delay followed the Link 1 delay which increased non-linearly from a minimum of 0.023 μs to approximately 3969 μs. As the load was incremented to $\{n_1 = 8, n_2 = 50\}$ the bond delay increased to 45600 μs and remained constant until $\{n_1 = 50, n_2 = 50\}$ after which it increased gradually to a maximum of 116700 μs at $\{n_1 = 100, n_2 = 50\}$. The overall mean link delay difference for all applied load combinations considered was 26525.4449 μs.

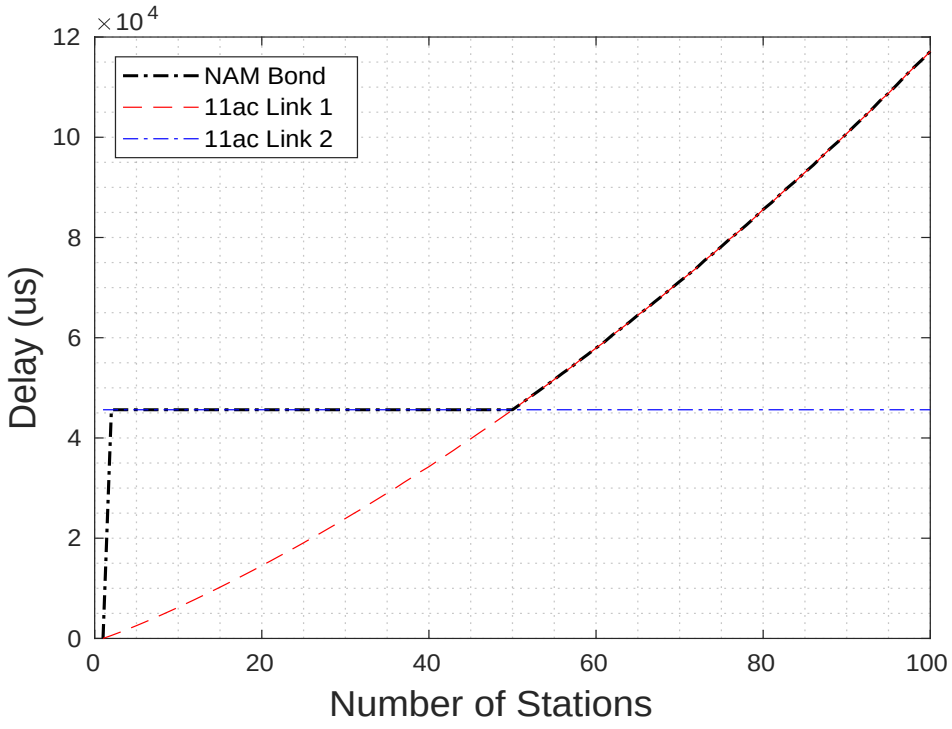

Figure 4.23: Test 1C Nam Algorithm Bond Delay Results

4.4.1.6 Nam Test 1C: Discussion

The main finding from Test 1C was that the dynamic interface selection algorithm by Nam increased the mean throughput by 18.77% and the link bandwidth utilisation by 2.53% when compared to the traditional static link selection approach. The static round robin method achieved a mean throughput of 1.2422 Mbps while the mean under the Nam algorithm was approximately 1.4754 Mbps and just 7.5% lower than the maximum possible mean throughput of 1.595 Mbps. The mean link bandwidth utilisation was 91.942 % compared to the round robin utilisation of 89.776%. The Nam throughput is plotted alongside the round robin throughput in Figure 4.24 below. The difference in behaviour between the two methods is evident between applied loads of $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 7, n_2 = 50\}$ where the Nam algorithm outperformed the static link selection in terms of the achieved bond throughput. However as the applied load was raised to $\{n_1 = 8, n_2 = 50\}$ and higher still towards $\{n_1 = 100, n_2 = 50\}$ the Nam algorithm performance was reduced to that of the round robin bond, decreasing exponentially to an overall test low of 0.411 Mbps.

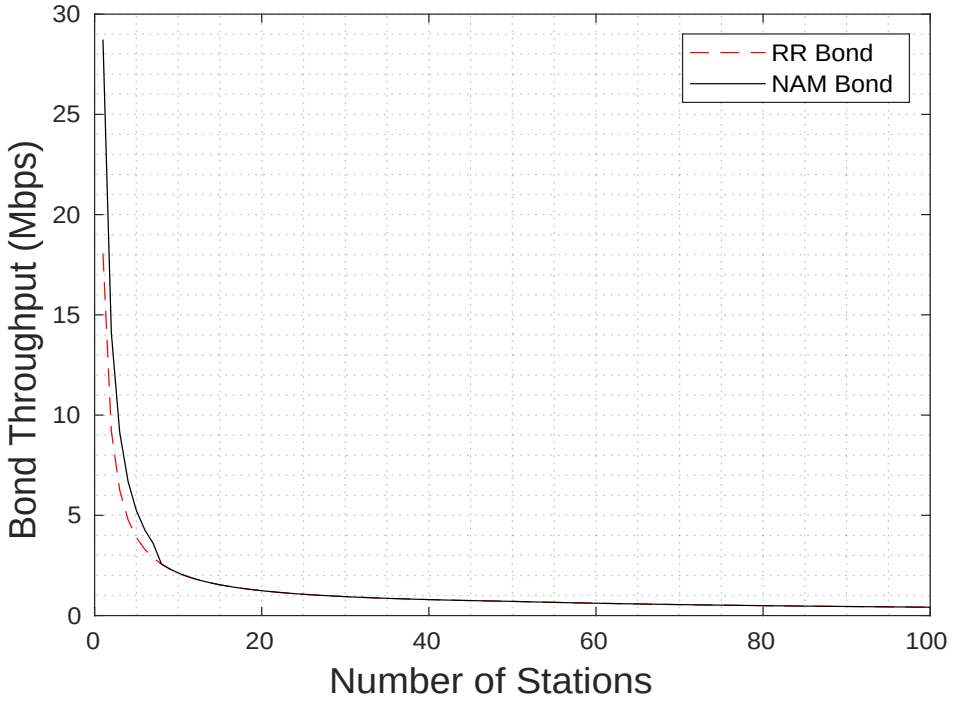

Figure 4.24: Test 1C Nam vs RR Bond Throughput Comparison

At the initial load of $\{n_1 = 1, n_2 = 50\}$ only the first 11ac link of the bond was in use as the measured throughput gap was in excess of the specified threshold. However as the applied load was incremented to $\{n_1 = 8, n_2 = 50\}$ the performance disparity between the two bond links decreased which caused the throughput gap to decrease below 0.9. The Nam algorithm then responded by reattaching the second 11ac link to the bond. The bond throughput gap continued to decrease to zero at an applied load of exactly $\{n_1 = 50, n_2 = 50\}$ where the performance of the two links was comparable, and then increased to just over 0.6 at the heaviest test load of $\{n_1 = 100, n_2 = 50\}$ as the performance disparity grew again. Note that the throughput gap output did not exceed the 0.9 threshold with the exception of applied loads where $\{n_1 < 8, n_2 = 50\}$.

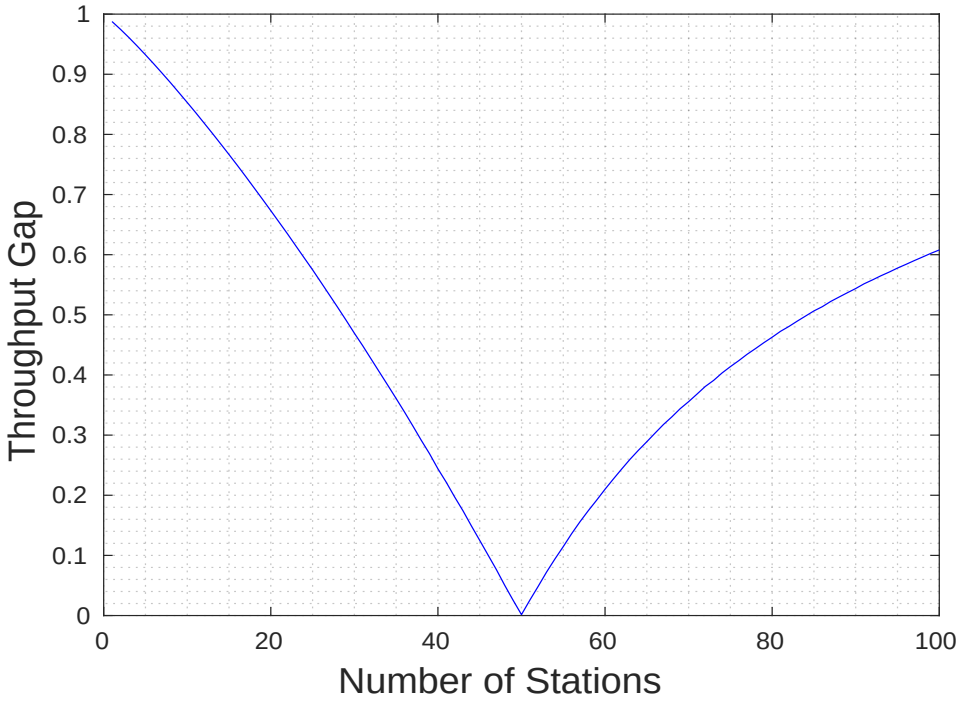

Figure 4.25: Test 1C Nam Algorithm Bond Throughput Gap Results

In Test 1C we saw the Nam algorithm respond to the measured bond throughput gap reducing below the 0.9 threshold and also saw the reaction when the fastest bond link changes from one slave to the other. From the bond delay for the test in Figure 4.23 of the previous results section we saw the overall behaviour change as the applied load reached $\{n_1 = 50,\ n_2 = 50\}$. However this change was not prompted by a crossing of the throughput gap threshold but instead by the relative throughput performance of the individual 11ac slave links. As shown in Figure 4.47 below for applied loads of $\{n_1 < 50,\ n_2 = 50\}$ the slower slave device was Link 2 with the constant applied load of n = 50, but for loads $\{n_1 > 50,\ n_2 = 50\}$ this was Link 1 with the variably applied load. This is of consequence for the Nam algorithm which states that when the throughput gap exceeds 0.9 then only the fastest single link of the bond should be used, and this link changed during the test as the applied load exceeded a value of $\{n_1 = 50,\ n_2 = 50\}$.

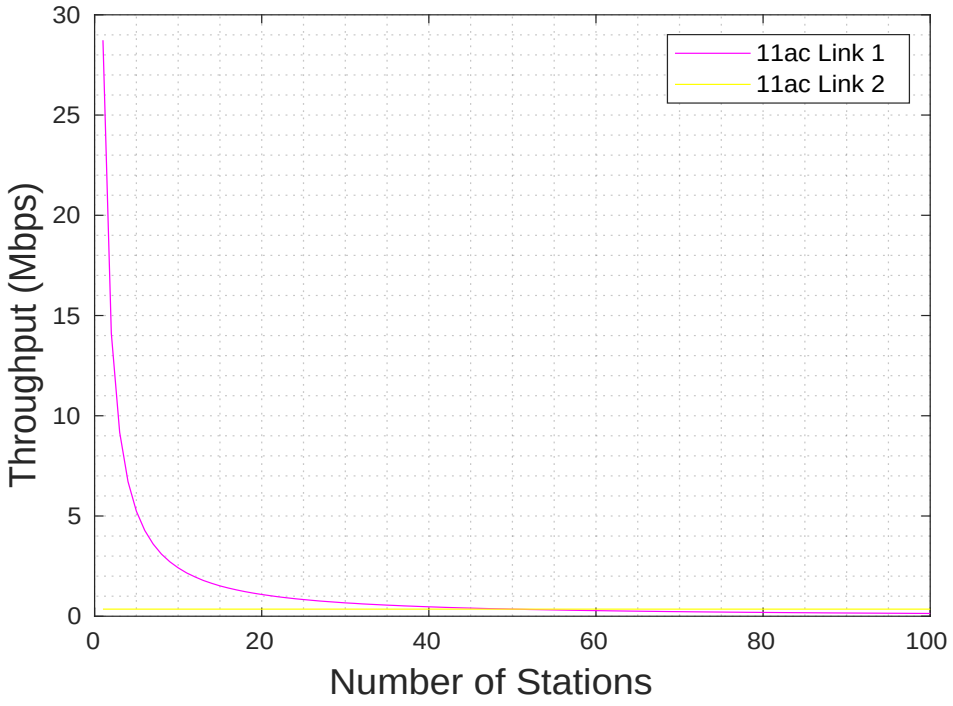

Figure 4.26: Test 1C Nam Algorithm Link Throughput Comparison

4.4.1.7 Nam Test 1D: Results

The throughput results for the dynamic link selection algorithm by Nam in Test 1D are shown in Figure 4.27 below. The achieved bond throughput is plotted as the dashed red line and the maximum theoretical throughput as the solid black line. The mean bond throughput for all applied load combinations considered test was 1.2725 Mbps. Both metrics displayed a non-linear decreasing curve as the applied bond load increased. At the initial load of $\{n_1 = 1, n_2 = 100\}$ the achieved bond throughput was 28.7 Mbps and the maximum throughput 28.84 Mbps and as the load increased towards $\{n_1 = 16, n_2 = 100\}$ both outputs decreased gradually to 1.404 Mbps and 1.543 Mbps respectively. As the load incremented to $\{n_1 = 17, n_2 = 1\}$ the maximum throughput continued its previous downward trajectory to 1.452 Mbps while the the achieved throughput decreased more significantly to 0.9552 Mbps. Both metrics continued to decrease as the applied load was further raised while the difference between them gradually reduced. At the heaviest considered load of $\{n_1 = 100, n_2 = 100\}$ both the achieved throughput and the maximum throughput were approximately 0.278 Mbps. The mean throughput difference between the two 11ac slave links for all applied load combinations considered in the test was 0.21799 Mbps.

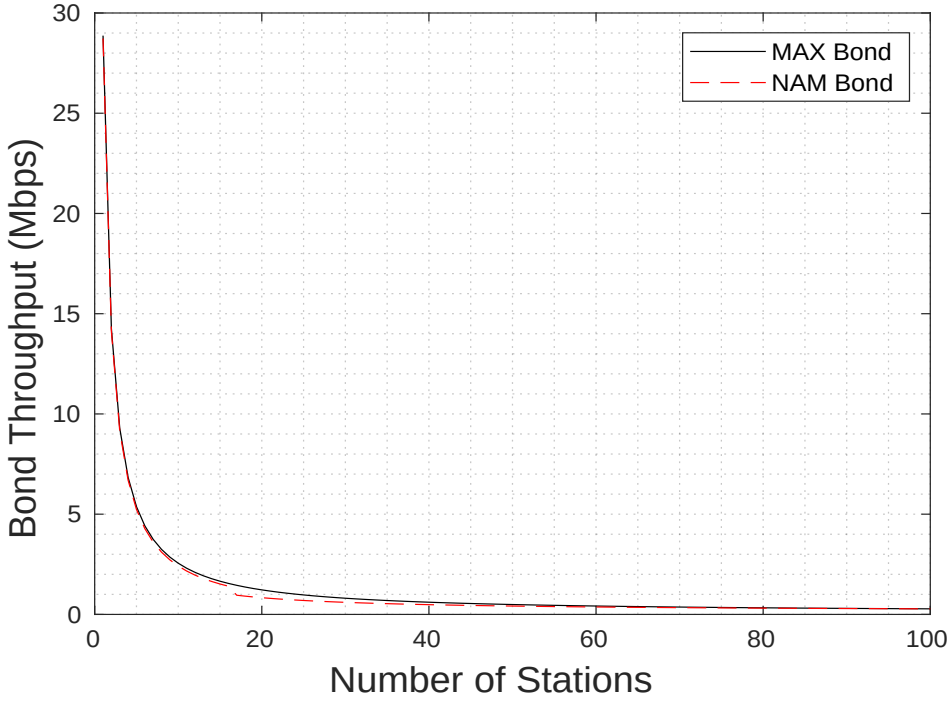

Figure 4.27: Test 1D Nam Algorithm Bond Throughput Results

The link bandwidth utilisation results for the Nam algorithm and the unbalanced load scenario in Test 1D are shown in Figure 4.28. The mean utilisation for all applied bond load combinations considered was 88.43%. As observed the utilisation was 100% for loads between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 16, n_2 = 100\}$. Then as the load incremented to $\{n_1 = 17, n_2 = 100\}$ the output decreased suddenly to a minimum of 66% before increasing non-linearly to 99.47% as the load was further raised towards $\{n_1 = 100, n_2 = 100\}$.

For applied loads between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 16, n_2 = 100\}$ the performance disparity between the two bond links created a bond throughput gap that was greater than the 0.9 threshold, and the Nam algorithm reacted by reverting to the use of a single link resulting in a bandwidth utilisation of 100%. Then as the load incremented to $\{n_1 = 17, n_2 = 100\}$ the performance disparity reduced which caused the throughput gap to drop below 0.9, and the Nam algorithm responded by re-attaching the second link which suddenly reduced the bandwidth utilisation. Then as the load was further raised to $\{n_1 = 100, n_2 = 100\}$ the performance disparity continued to decrease and the resulting bandwidth utilisation gradually increased.

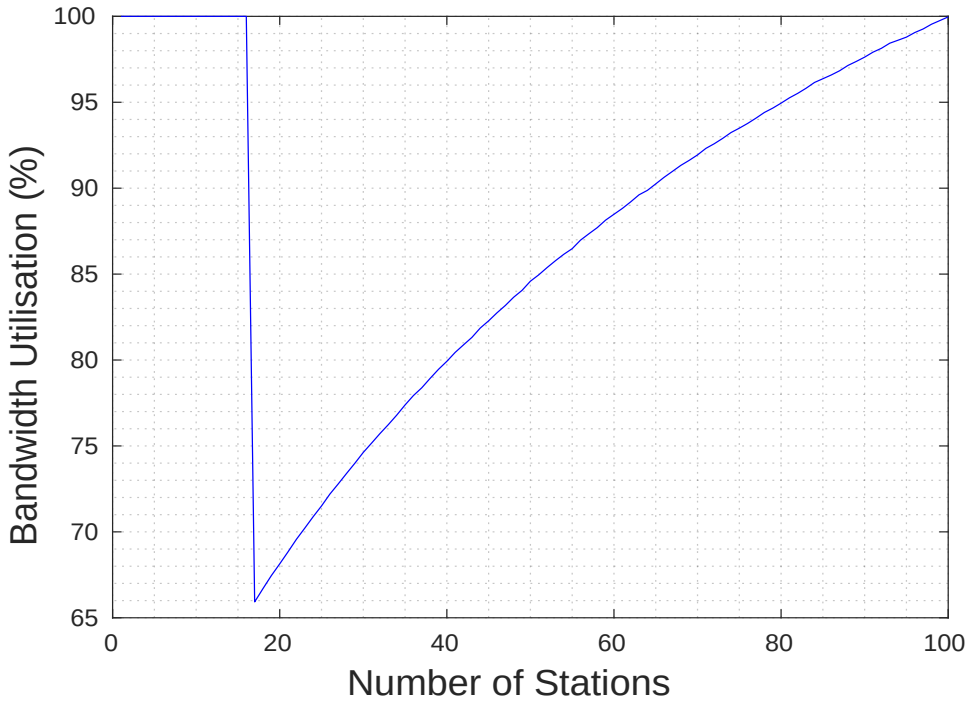

Figure 4.28: Test 1D Nam Algorithm Link Bandwidth Utilisation Results

The Nam algorithm bond delay results for Test 1D are given in Figure 4.29. The overall mean bond delay for all applied loads used was 115962.75 μs. As seen below for applied loads between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 16, n_2 = 100\}$ the bond delay followed the Link 1 delay which increased non-linearly from 0.0286 μs to 10930 μs. As the applied load was incremented to $\{n_1 = 17, n_2 = 100\}$ the delay began to follow the Link 2 delay and increased to approximately 115962.75 μs and then remained constant as the load was further increased towards the heaviest considered level of $\{n_1 = 100, n_2 = 100\}$. The overall mean delay difference between the twin 11ac links for all applied loads considered in the test was 49014.35 μs.

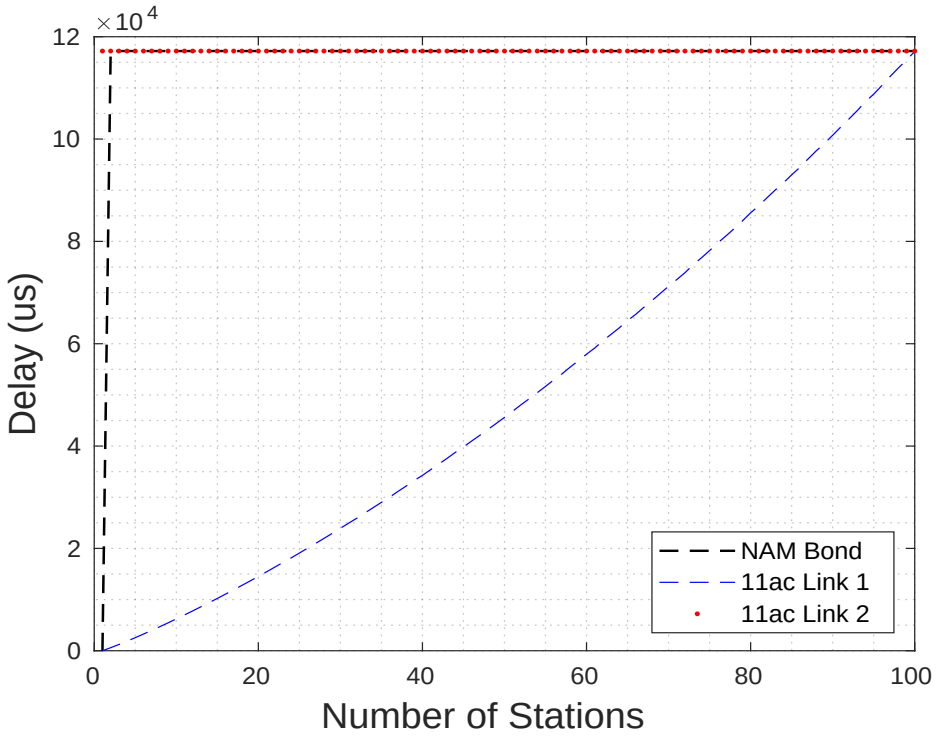

Figure 4.29: Test 1D Nam Algorithm Bond Delay Results

4.4.1.8 Nam Test 1D: Discussion

The main finding from Test 1D was that the dynamic Nam algorithm outperformed the static round robin approach by 48.3% in terms of the mean bond throughput and increased the mean link bandwidth utilisation by 8.44%. The mean bond throughput under the Nam algorithm was 1.2725 Mbps with a mean bond throughput efficiency of 88.394%, while the achieved mean values for round robin were 0.85805 Mbps and 81.615% respectively. The bond throughput for both methods is plotted alongside the maximum theoretical throughput in Figure 4.30 below. The difference in behaviour between the static and dynamic link configuration approaches is evident between applied loads of $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 16, n_2 = 100\}$. During this range as the load was increased both outputs decreased exponentially and the round robin throughput was significantly less than the Nam algorithm throughput. However as the load incremented to $\{n_1 = 17, n_2 = 100\}$ the Nam algorithm throughput reduced to approximately the same value as round robin and remained constant as the load was then further increased towards the heaviest prescribed level of $\{n_1 = 100, n_2 = 100\}$.

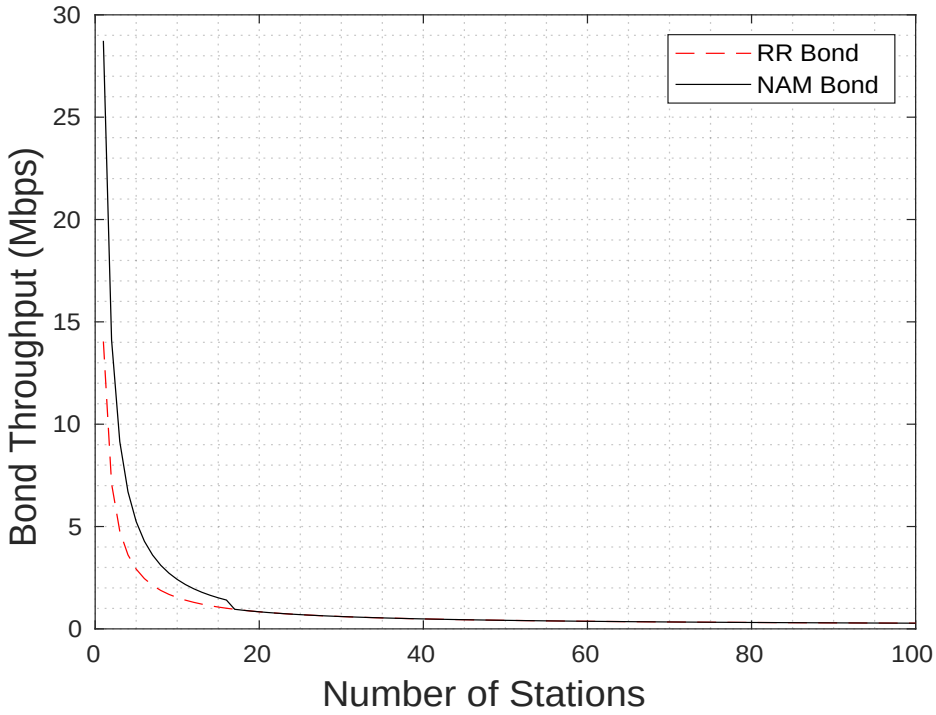

Figure 4.30: Test 1D Nam vs RR Bond Throughput Comparison

In Test 1D we saw the Nam algorithm respond to the bond throughput gap decreasing below the specified threshold by re-attaching the slower 11ac slave link. As seen in Figure 4.31 between applied loads of $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 16, n_2 = 100\}$ only the first 11ac link of the bond was in use. Then as the load was increased on the first 11ac link the performance difference decreased until a load of $\{n_1 = 17, n_2 = 1\}$ was reached at which point the measured bond throughput gap reduced below the given threshold of 0.9. The dynamic link selection algorithm reacted by re-attaching the second 11ac link to the bond and in doing increased the overall mean bond throughput as well as the overall link bandwidth utilisation. As seem from the graph the throughput gap did not rise above the specified threshold and so both 802.11 links were used for the remainder of the test.

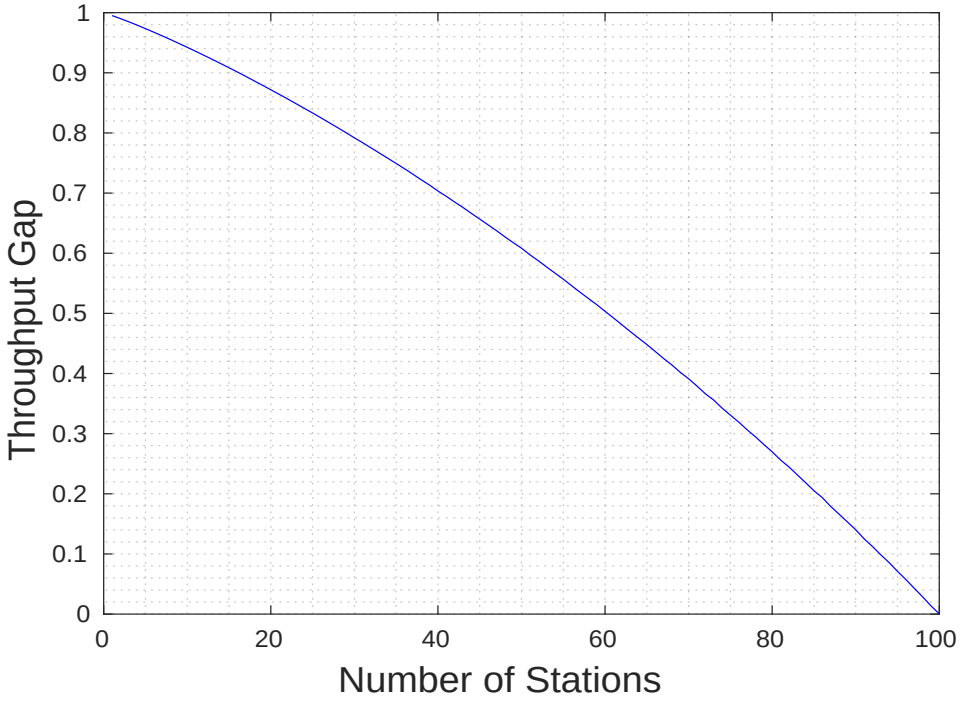

Figure 4.31: Test 1D Nam Algorithm Bond Throughput Gap Results

4.4.2 Heterogeneous 11ac-11ah Bond

Next we present the results for the heterogeneous 11ac-11ah bond in Tests 2A to 2D.

4.4.2.1 Nam Test 2A: Results

The bond throughput for the Nam algorithm and balanced load in Test 2A is plotted in Figure 4.32 as the solid blue line alongside the maximum theoretical bond throughput as the solid black line. The overall mean Nam bond throughput for all applied loads used was 1.2999 Mbps and the maximum theoretical mean 11ac-11ah bond throughput was 1.8897 Mbps. As seen the Nam output decreased non-linearly as the applied bond load was raised and there was a visible difference between the achieved bond throughput and the theoretical maximum. From a maximum of 26.34 Mbps at the initial load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ the Nam bond throughput decreased non-linearly to a minimum of 0.165 Mbps at the heaviest applied load of $\{n_{11ac} = 100, n_{11ah} = 100\}$. Over the same range the maximum theoretical bond throughput decreased non-linearly from a maximum of 41.89 Mbps to a minimum of 0.22 Mbps. The overall mean link throughput difference for all applied loads considered was 0.59 Mbps.

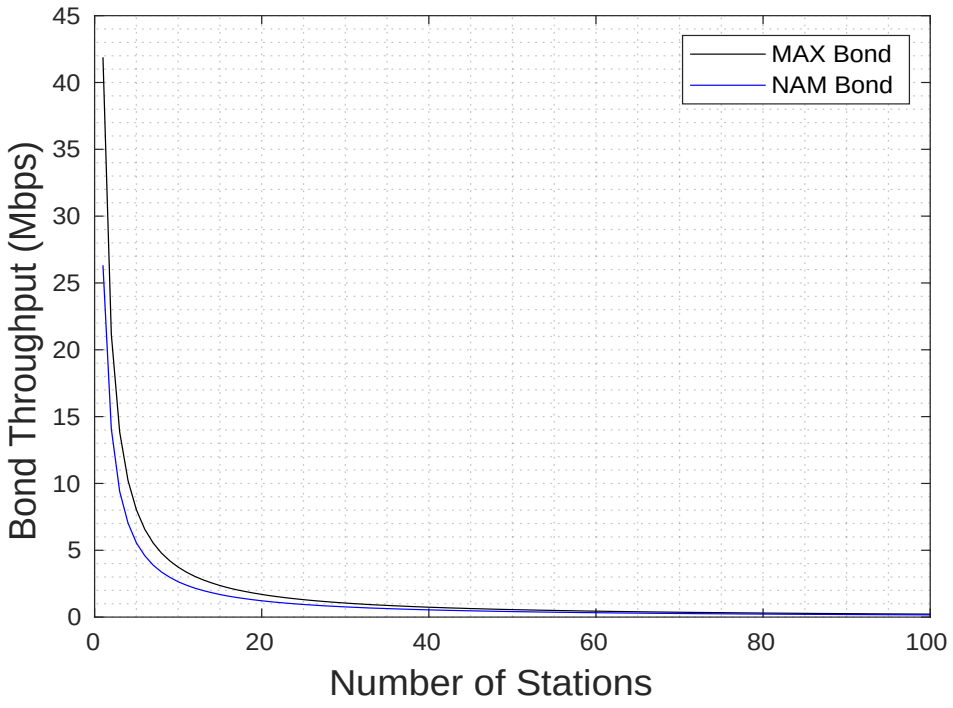

Figure 4.32: Test 2A Nam Algorithm Bond Throughput Results

The 11ac-11ah link bandwidth utilisation results for the dynamic Nam link selection algorithm in Test 2A are shown in Figure 4.33. The overall mean utilisation for all applied load combinations considered in the test was 72.9%. As seen the efficiency increased non-linearly from a minimum of 62.8% at $\{n_{11ac} = 1, n_{11ah} = 1\}$ to a maximum of 74.5% at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$.

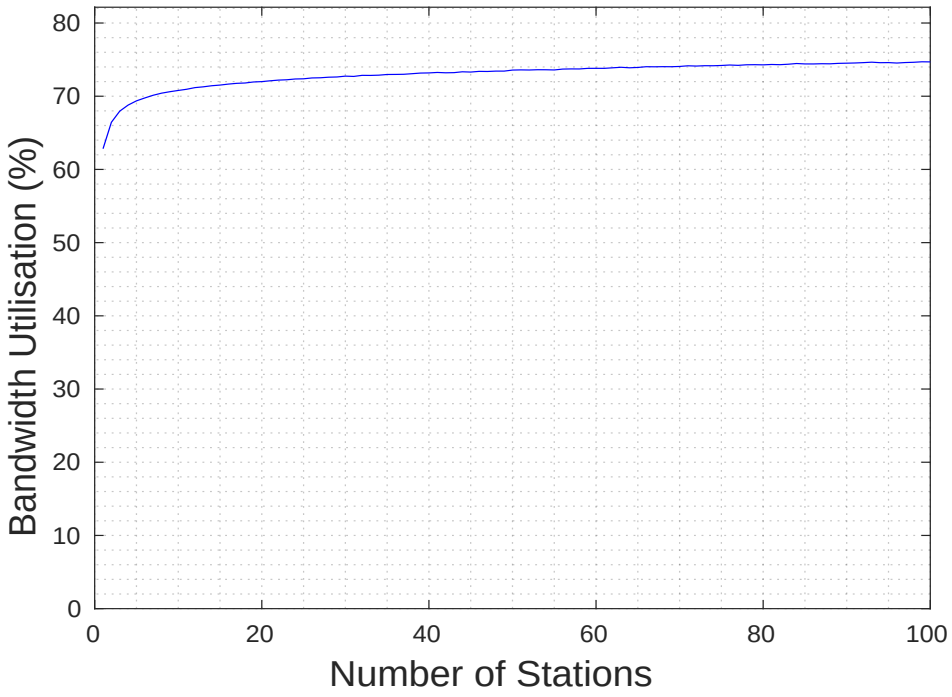

Figure 4.33: Test 2A Nam Algorithm Link Bandwidth Utilisation Results

The bond delay results under the Nam algorithm for the balanced load scenario in Test 2A are presented in Figure 4.34. The overall mean bond delay for all applied load combinations considered was 85313.46 μs. As seen below the bond delay followed the slower 11ah delay for all applied load combinations. From a minimum of 0.0136 μs at $\{n_{11ac} = 1, n_{11ah} = 1\}$ the bond delay increased non-linearly to a maximum value of 196500 μs at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$. The overall mean access delay difference between the 11ac and 11ah slave links for all applied load combinations considered was 35112.9 μs.

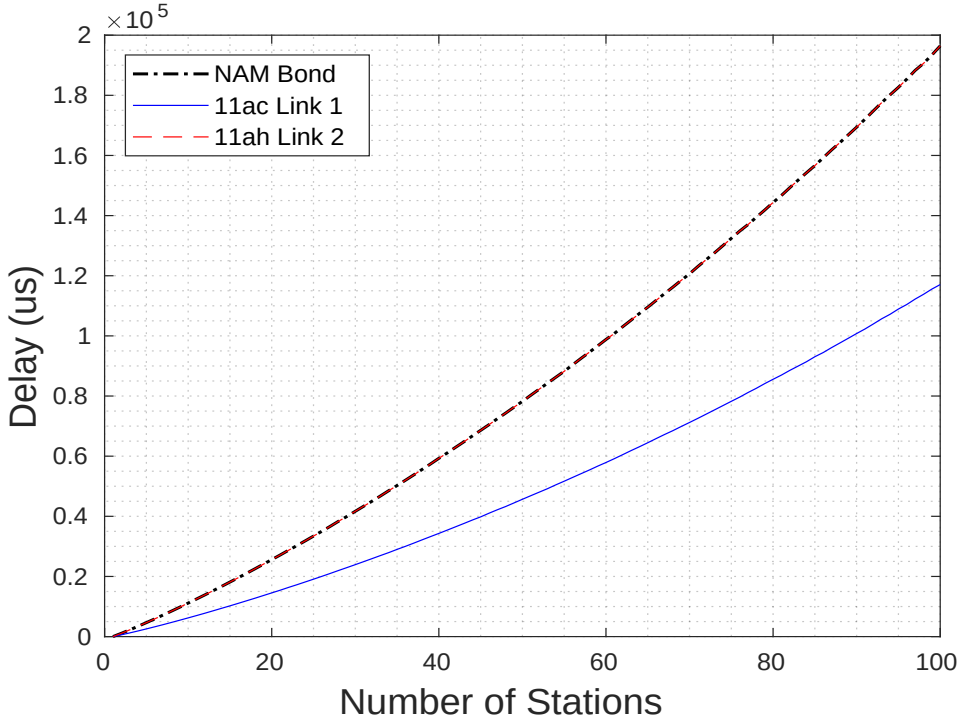

Figure 4.34: Test 2A Nam Algorithm Bond Delay Results

4.4.2.2 Nam Test 2A: Discussion

The main observation from the Nam evaluation in Test 2A was that the dynamic Nam algorithm performance was equivalent to static round robin for all applied load combinations used. This was similar to Test 1A above and is seen for the current experiment in Figure 4.35 below where the Nam throughput is plotted as the dotted blue line and the round robin throughput as the solid red line. The achieved bond throughput under both dynamic and static approaches was approximately equal for all applied loads from $\{n_{11ac} = 1, n_{11ah} = 1\}$ to $\{n_{11ac} = 100, n_{11ah} = 100\}$. Furthermore both methods achieved an overall mean bond throughput of 1.2999 Mbps and a mean link utilisation of 72.98%.

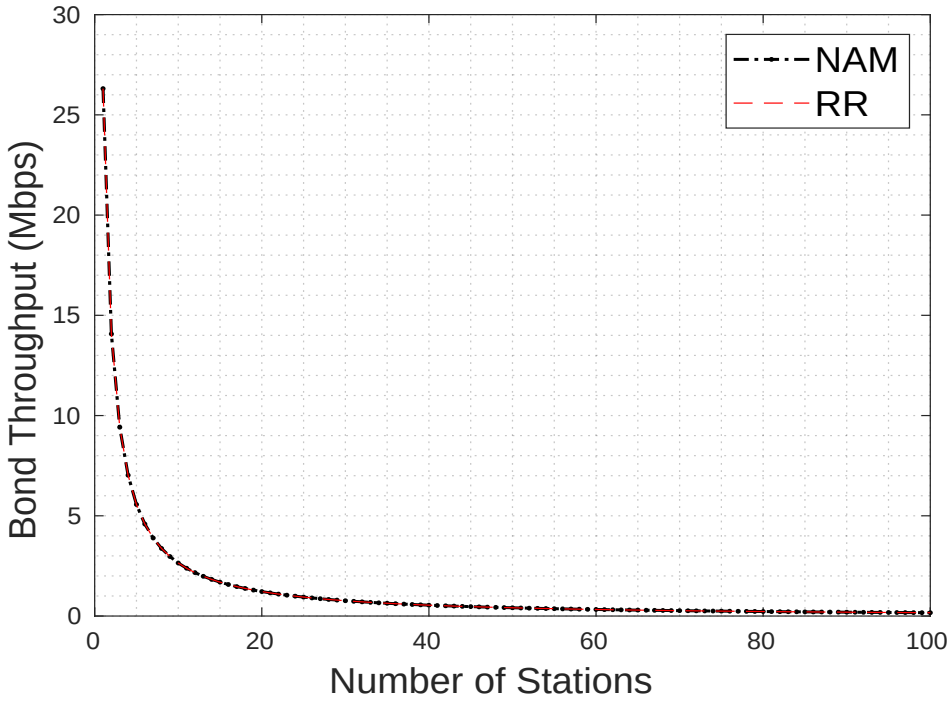

Figure 4.35: Test 2A Nam vs RR Algorithm Bond Throughput Performance Comparison

The behaviour observed was as expected and due to the fact that the bond throughput gap threshold of 0.9 for removing a problematic link was not exceeded at any stage of the simulation. Therefore both 11ac and 11ah links were in use at all times. This also occurred for Nam using the homogeneous 11ac-11ac bond in Test 1A. However the difference for the current test is the heterogeneous 11ac-11ah bond used and the non-identical DCF parameters which in results a significant bond throughput gap at all applied loads. This is seen for the current test in Figure 4.36 below where the bond throughput gap was measured at 0.54 at $\{n_{11ac} = 1, n_{11ah} = 1\}$ and then decreased non-linearly to 0.4 at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$. Crucially the bond throughput gap did not rise above the specified threshold of 0.9 at any point during the simulation and therefore both 11ac and 11ah links were in use at all times and the Nam performance was equivalent to that of the traditional static round robin approach.

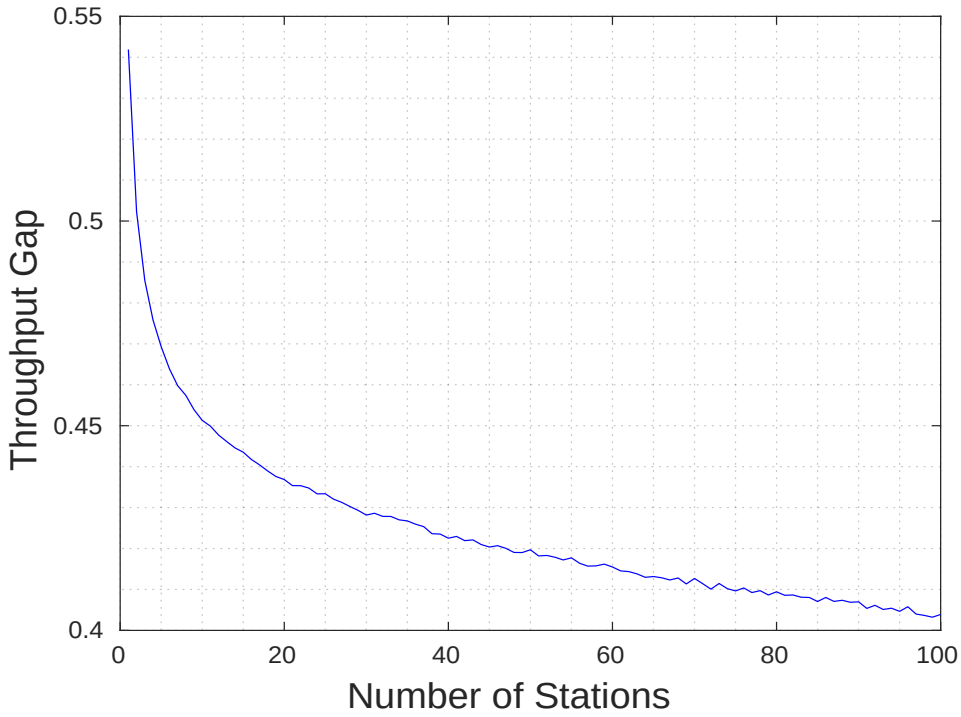

Figure 4.36: Test 2A Nam Algorithm Bond Throughput Gap

4.4.2.3 Nam Test 2B: Results

The bond throughput under the Nam algorithm for the unbalanced scenario in Test 2B is plotted in Figure 4.37 as the solid blue line alongside the maximum theoretical bond throughput as the solid black line. The mean bond throughput for all applied loads considered was 13.6157 Mbps compared to a theoretical maximum mean bond throughput of 29.9717 Mbps. This is also reflected below where the achieved Nam throughput is visibly lower than the theoretical maximum bond throughput at all applied loads. The output of both metrics decreased non-linearly as the load was raised. At $\{n_{11ac} = 1, n_{11ah} = 1\}$ the achieved bond throughput was 26.31 Mbps while the maximum theoretical bond throughput was 41.89 Mbps. As the load was incremented to $\{n_{11ac} = 17, n_{11ah} = 1\}$ the maximum theoretical throughput continued in its previous downward trajectory while the achieved bond throughput reduced more significantly to 13.16 Mbps and remained constant as the applied load was further raised towards $\{n_{11ac} = 100, n_{11ah} = 1\}$. Meanwhile the maximum bond throughput reduced gradually to 13.3 Mbps. The mean link throughput difference for all applied load combinations used was 1.5315 Mbps.

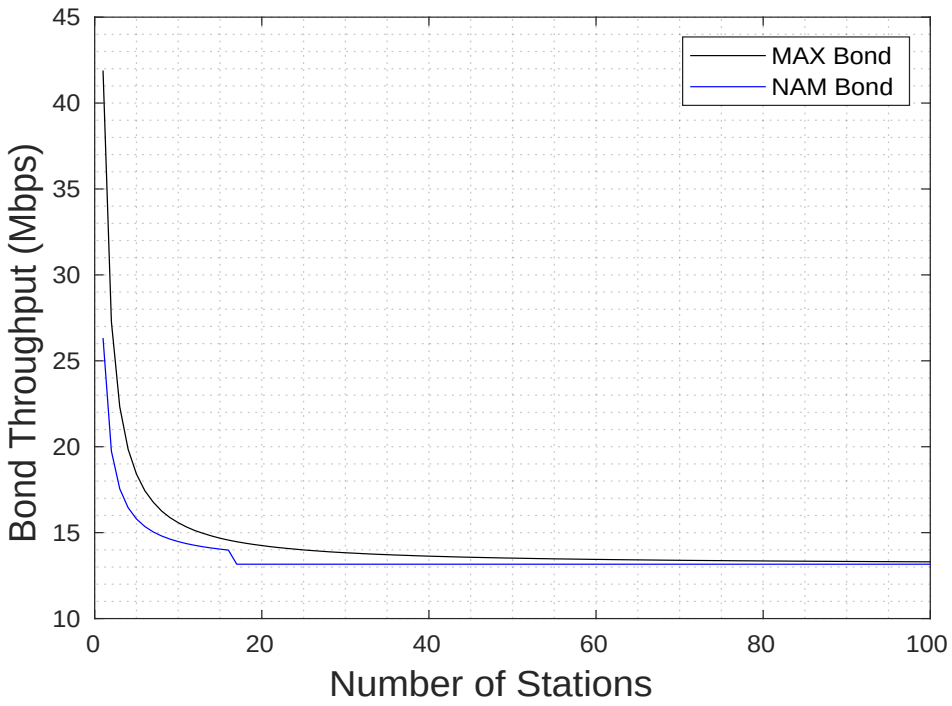

Figure 4.37: Test 2B Nam Algorithm Bond Throughput

The link bandwidth utilisation results for the Nam algorithm in Test 2B are given in Figure 4.38. The overall mean 11ac-11ah utilisation for all applied load combinations used was 98.059%. During the simulation the achieved utilisation increased non-linearly from a minimum of 62.81% at $\{n_{11ac} = 1, n_{11ah} = 1\}$ to 95.96% at $\{n_{11ac} = 16, n_{11ah} = 1\}$. Then as the load was incremented to $\{n_{11ac} = 17, n_{11ah} = 1\}$ the utilisation increased suddenly to 100% and remained constant as the load was further raised towards $\{n_{11ac} = 100, n_{11ah} = 1\}$.

This occurred because at applied loads between $\{n_{11ac} = 1, n_{11ah} = 1\}$ and $\{n_{11ac} = 16, n_{11ah} = 1\}$ the performance disparity between the bond links was low such that the bond throughput gap was less than 0.9. Then, as the load was incremented to $\{n_{11ac} = 17, n_{11ah} = 1\}$ the performance disparity increased between links increased enough for the throughput gap to exceed the 0.9 threshold. The Nam algorithm reacted by removing the problematic link, leaving only a single 802.11 link and a 100% bandwidth utilisation for the remainder of the test.

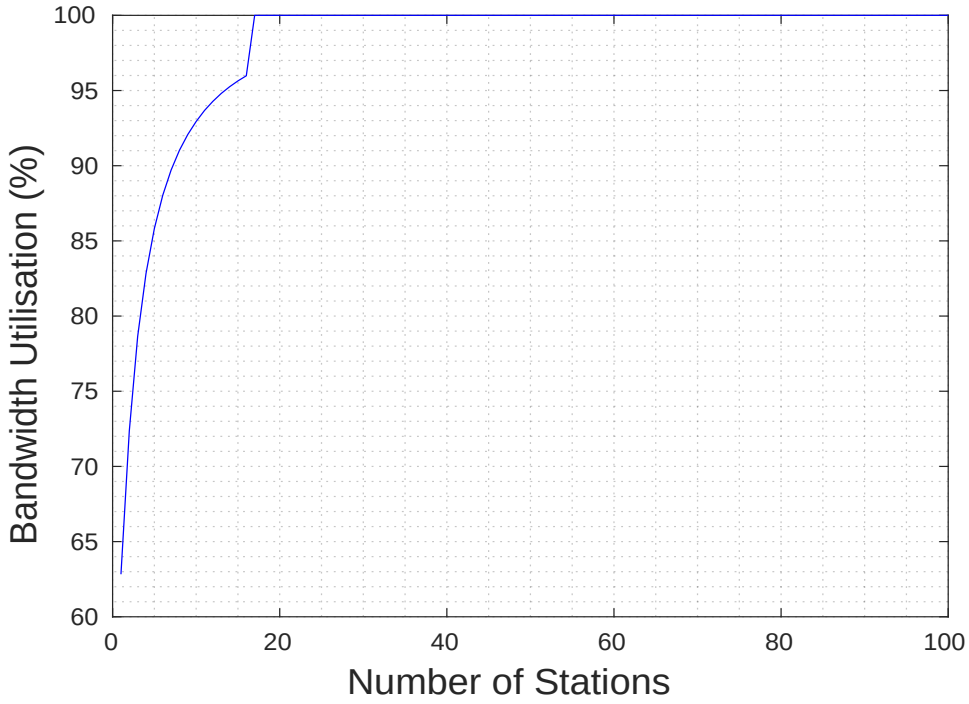

Figure 4.38: Test 2B Nam Algorithm Link Bandwidth Utilisation

The 11ac-11ah bond delay results for the Nam algorithm in Test 2B are plotted in Figure 4.39 below. The overall mean bond delay for all applied load combinations considered was 50197.75 μs. The bond delay followed the 11ah link delay initially and increased non-linearly from a minimum of 0.01984 μs at $\{n_{11ac} = 1, n_{11ah} = 1\}$ to a maximum value of 11050 μs at $\{n_{11ac} = 16, n_{11ah} = 1\}$. Then as the load was incremented to $\{n_{11ac} = 17, n_{11ah} = 1\}$ the delay output reduced to roughly its value and remained constant following the delay of the statically loaded 11ac link as the load as further raised towards $\{n_{11ac} = 100, n_{11ah} = 1\}$. The overall mean delay difference between the 11ac and 11ah bond links for all applied load combinations used was 838.1 μs.

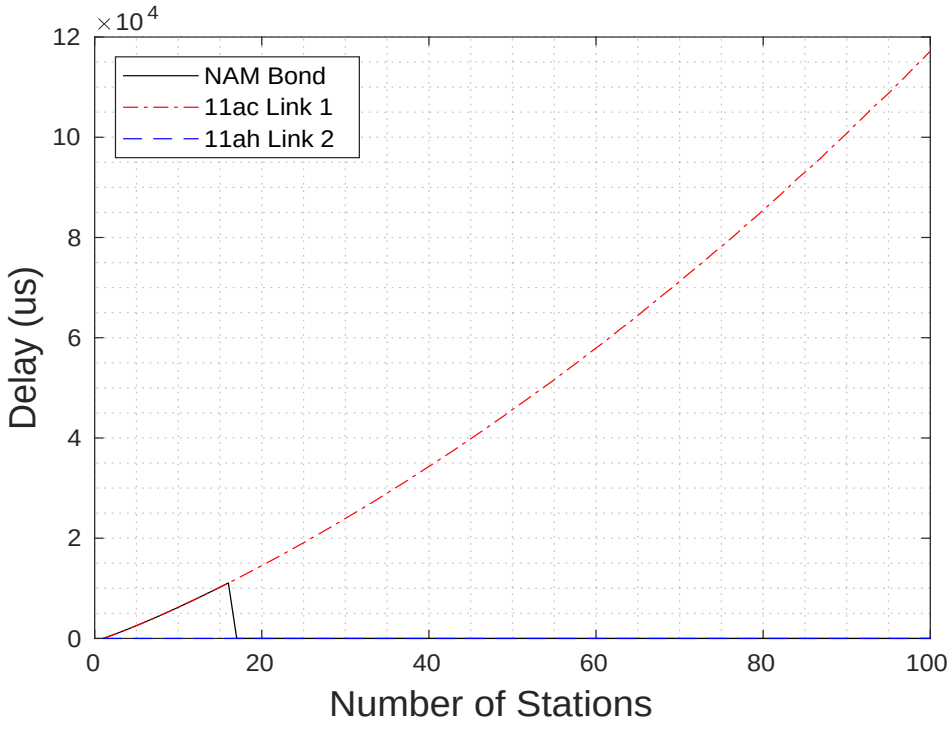

Figure 4.39: Test 2B Nam Algorithm Bond Delay

4.4.2.4 Nam Test 2B: Discussion

In Test 2B the dynamic Nam algorithm under-performed the static round robin technique by 1.74% in terms of the mean bond throughput while outperforming the mean link utilisation marginally by 0.72%. The overall mean throughput of the Nam algorithm was 13.6157 Mbps with a mean throughput efficiency of 98.059% while under static round robin these values were 13.8537 Mbps and 97.363% respectively. The Nam bond throughput for the test is plotted alongside the round robin throughput and maximum theoretical bond throughput in Figure 4.40 below. The difference in behaviour between the dynamic and static link selection methods is evident between applied loads of $\{n_{11ac} = 18, n_{11ah} = 1\}$ and $\{n_{11ac} = 100, n_{11ah} = 1\}$. As the load was raised from $\{n_{11ac} = 1, n_{11ah} = 1\}$ to $\{n_{11ac} = 17, n_{11ah} = 1\}$ the Nam algorithm performance was identical to round robin, but when incremented to $\{n_{11ac} = 18, n_{11ah} = 1\}$ the Nam throughput dropped significantly to 13.16 Mbps and then remained constant as the load was further increased to $\{n_{11ac} = 100, n_{11ah} = 1\}$. By contrast the round robin throughput continued its slower downward trend.

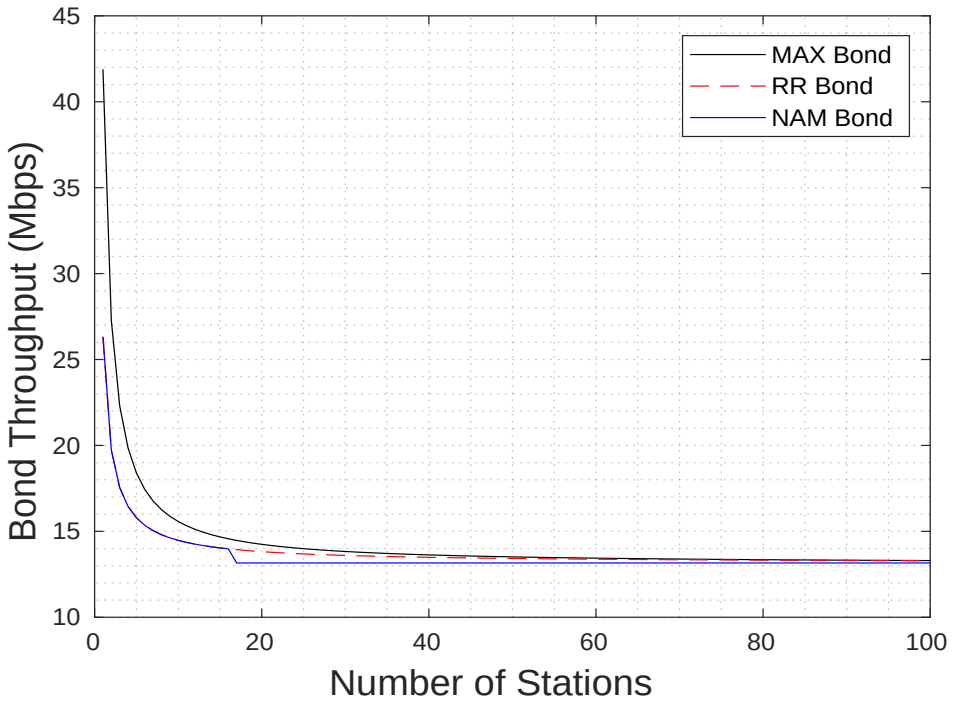

Figure 4.40: Test 2B Nam vs RR Algorithm Bond Throughput Performance Comparison

Similar to Test 1B and as expected the Nam algorithm reacted to the bond throughput gap exceeding the prescribed threshold by removing the slower of the 11ac and 11ah slave links. At $\{n_{11ac} = 1, n_{11ah} = 1\}$ both slave links were attached to the bond but as the load on the 11ac link was raised the bond became increasingly unbalanced resulting in the bond throughput gap exceeding 0.9 at $\{n_{11ac} = 17, n_{11ah} = 1\}$ and triggering the automatic removal of the slower bond link. In doing so the Nam algorithm actually under-performed the static round robin mean bond throughput but marginally out-performed the mean bandwidth utilisation. Again the switch to a single link was the reason the link throughput difference and delay difference suddenly reduce to zero at a load of $\{n_{11ac} = 17, n_{11ah} = 1\}$ and was also why the bond throughput efficiency increased to 100% at the same load.

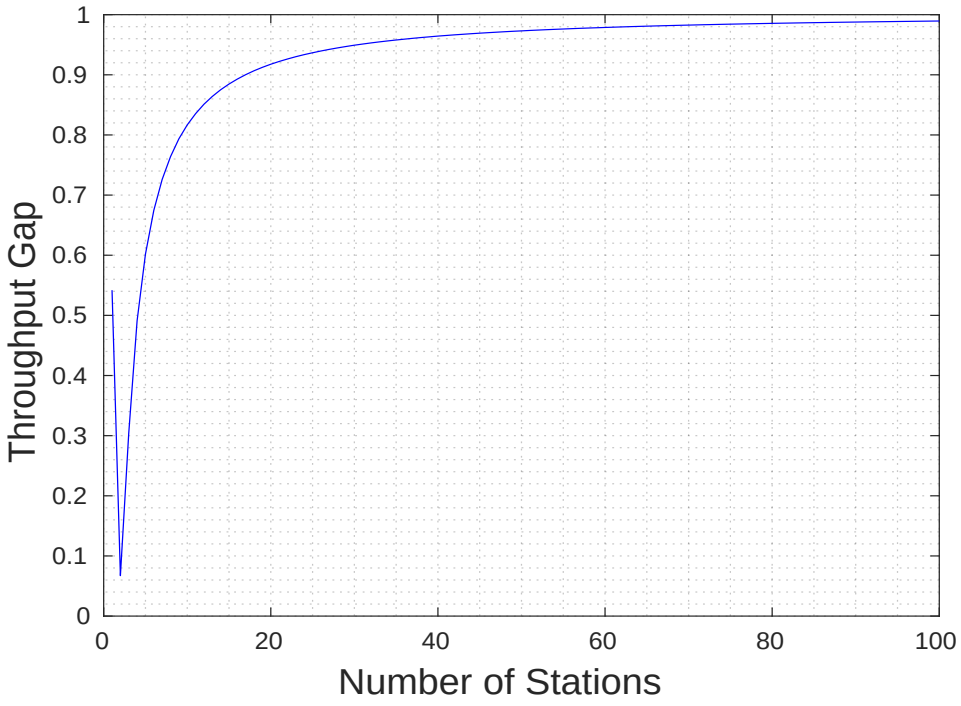

Figure 4.41: Test 2B Nam Algorithm Bond Throughput Gap Results

4.4.2.5 Nam Test 2C: Results

The bond throughput results for the Nam algorithm in Test 2C are shown in Figure 4.42. The achieved Nam throughput is plotted as the solid blue line and the maximum theoretical bond throughput as the solid black line. The overall mean throughput for the heterogeneous 11ac-11ah bond for all applied load combinations considered was 1.2297 Mbps. A non-linear decreasing output was observed as the applied bond load was raised with a visible difference between the achieved throughput and the theoretical maximum. At the initial load of $\{n_{11ac} = 1, n_{11ah} = 50\}$ the achieved throughput was measured at 28.73 Mbps and the maximum throughput at 28.93 Mbps. As the load increased to $\{n_{11ac} = 11, n_{11ah} = 50\}$ the achieved throughput decreased non-linearly to 2.163 Mbps and the maximum throughput to 2.369 Mbps while the difference between the two metrics appeared constant. However as the load was incremented to $\{n_{11ac} = 12, n_{11ah} = 50\}$ the maximum throughput reduced to 2.17 Mbps while the achieved throughput reduced more significantly to 1.064 Mbps. As the load was further raised towards $\{n_{11ac} = 100, n_{11ah} = 50\}$ both metrics continued their general downward trend and the difference between them appeared to reduce. At the maximum test load the achieved bond throughput was measured at 0.3088 Mbps and the maximum theoretical bond throughput at 0.3451 Mbps. The mean throughput difference for all applied load combinations considered was 0.25294 Mbps.

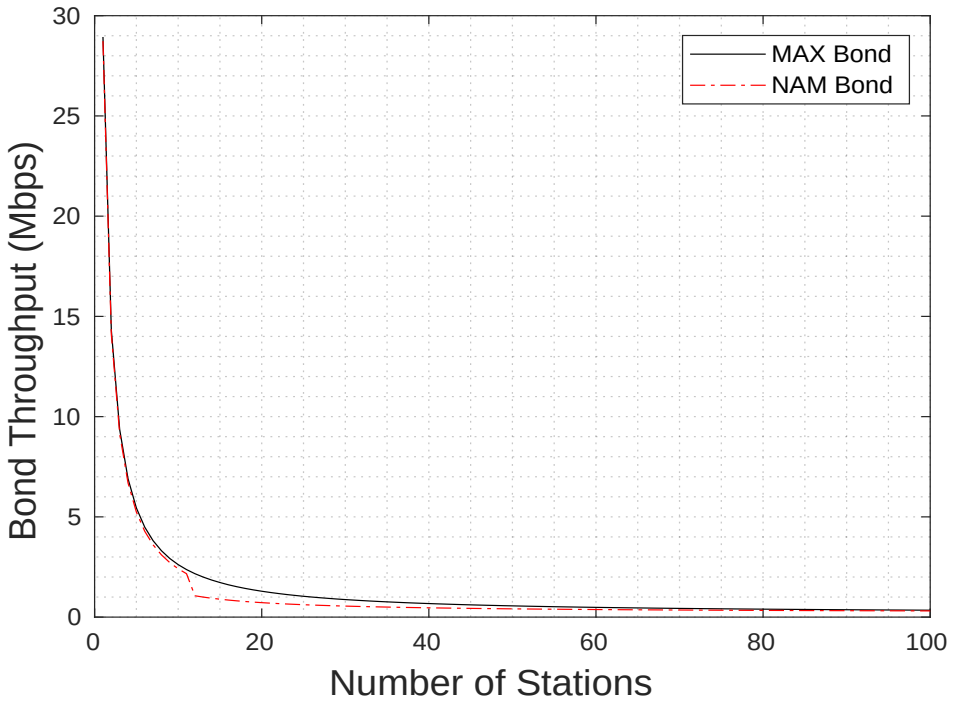

Figure 4.42: Test 2C Nam Algorithm Bond Throughput Results

The link bandwidth utilisation for the Nam algorithm in Test 2C is shown in Figure 4.43. The overall mean bond utilisation for all loads considered was 76.76%. As seen from the graph for applied loads between $\{n_{11ac} = 1, n_{11ah} = 50\}$ and $\{n_{11ac} = 11, n_{11ah} = 50\}$ the achieved utilisation was 100%. Then as the load was incremented to $\{n_{11ac} = 12, n_{11ah} = 50\}$ the simulation output decreased suddenly to a minimum of 49.01%. As the load was further raised the efficiency increased non-linearly to a final value of 89.47% at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 50\}$.

Here, for applied loads between $\{n_{11ac} = 1, n_{11ah} = 50\}$ and $\{n_{11ac} = 11, n_{11ah} = 50\}$ only a single link was attached to the bond due to the difference in applied load and the resulting performance disparity across the bond. Because only a single link was active, the resulting bandwidth utilisation was equal to 100%. Then as the load incremented to $\{n_{11ac} = 12, n_{11ah} = 50\}$, the Nam algorithm re-attached the inactive link as the disparity reduced and the bond throughput gap dropped below the 0.9 threshold. This had the effect of suddenly reducing the overall bandwidth utilisation. As the load was further increased, the performance disparity between the bond links gradually reduced which increased the bandwidth utilisation.

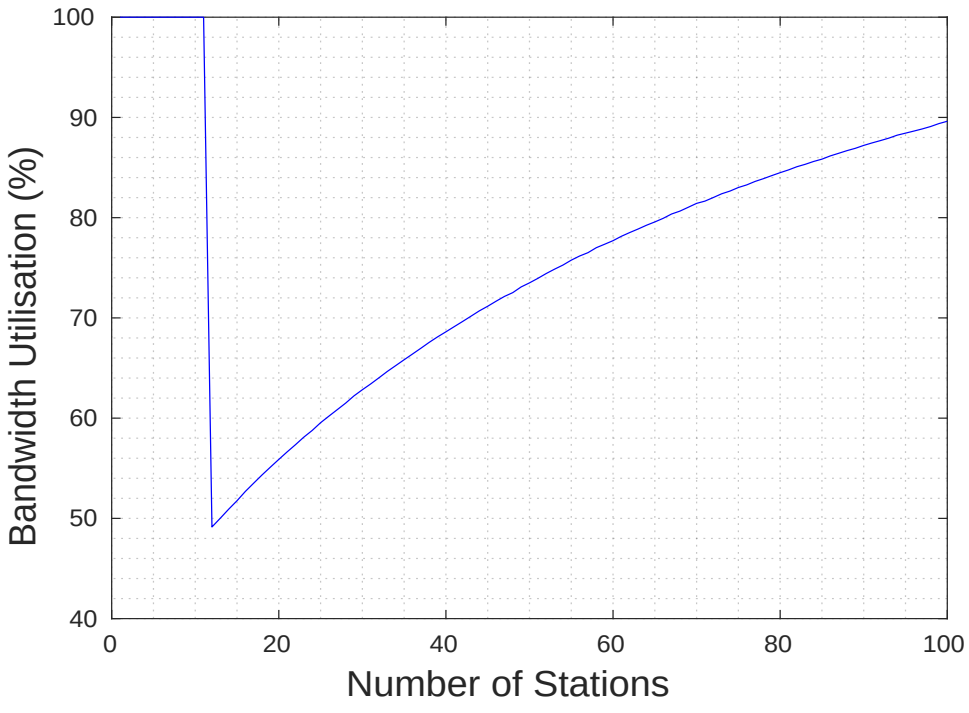

Figure 4.43: Test 2C Nam Algorithm Link Bandwidth Utilisation Results

The bond delay results under the Nam algorithm in Test 2C are presented in Figure 4.44. The overall mean bond delay for all applied load combinations used was 57953.634 μs. For the initial applied load of $\{n_{11ac} = 1, n_{11ah} = 50\}$ the bond delay was equal to the 11ac link delay at 0.002348 μs. Then as the load was incremented to $\{n_{11ac} = 2, n_{11ah} = 50\}$ the bond delay began to follow the 11ah link and increased suddenly to 77900 μs. The output remained constant as the applied load was further raised until $\{n_{11ac} = 75, n_{11ah} = 50\}$ where the bond delay began to follow the 11ac link and increased non-linearly to 117200 μs at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 50\}$. The mean delay difference for all applied load combinations used was 12682.3774 μs.

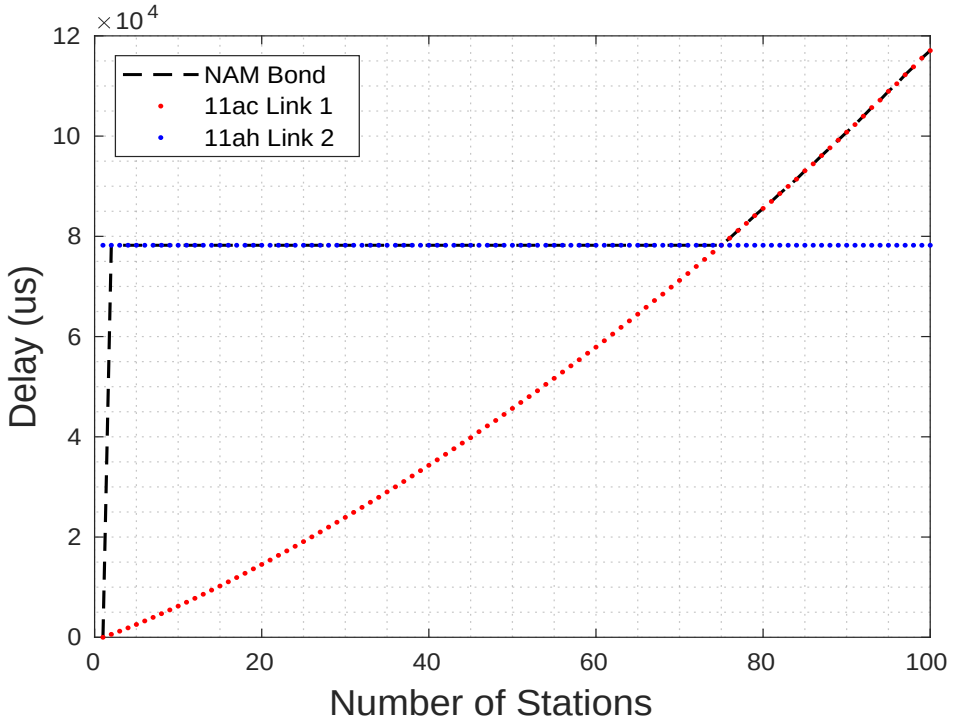

Figure 4.44: Test 2C Nam Algorithm Bond Delay Results

4.4.2.6 Nam Test 2C: Discussion

In Test 2C the Nam selection algorithm outperformed the static round robin method by 38.757 % in terms of achieved mean throughput and by 9.91% in terms of the bandwidth utilisation. The achieved mean round robin bond throughput was 0.7506 Mbps while the Nam algorithm mean was 1.2297 Mbps. The mean link utilisation for round robin was 70.479% while for Nam this was 76.76%. The Nam throughput is plotted alongside the round robin throughput and the maximum theoretical throughput in Figure 4.45. The difference in behaviour between the two approaches is seen between applied loads of $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 11, n_2 = 50\}$. As the load was increased during this range the Nam algorithm throughput was close to the maximum theoretical throughput while the achieved round robin throughput was significantly lower. Then as the load was incremented $\{n_1 = 12, n_2 = 50\}$ the Nam algorithm throughput dropped suddenly to approximately 1.06 Mbps, the same value as the the round robin method, where it remained as the applied load was further increased towards $\{n_1 = 100, n_2 = 50\}$.

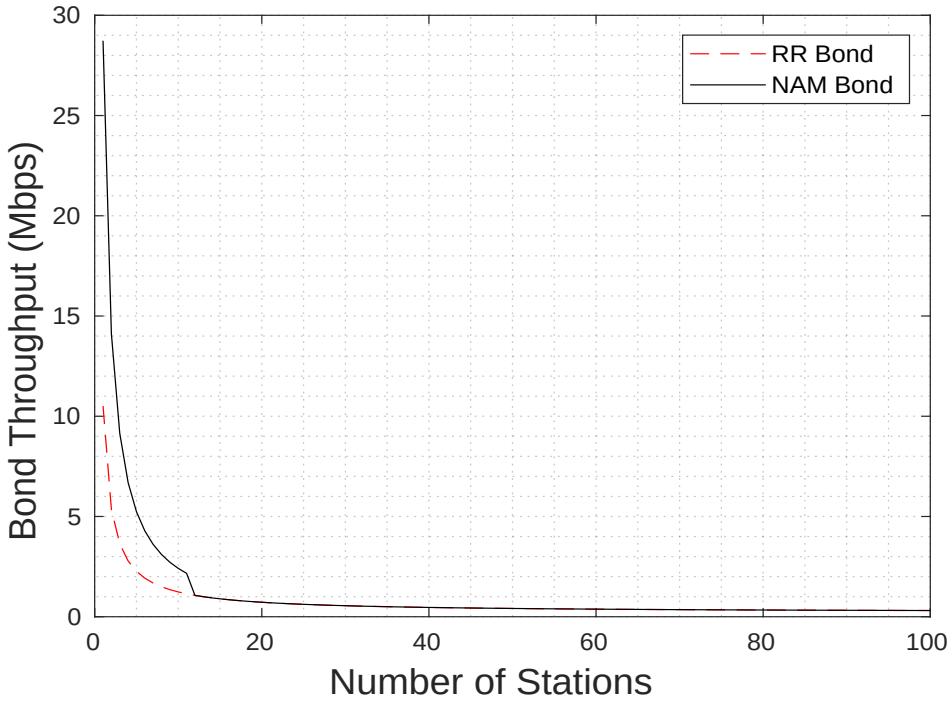

Figure 4.45: Test 2C Nam vs RR Algorithm Bond Throughput Performance Comparison

At the initial load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ only the 11ac bond link was in use due to the measured bond throughput gap being in excess of the prescribed threshold. The throughput gap continued to reduce as the individual throughput performance of the two bond links became more comparable, and as the load reached $\{n_1 = 12, n_2 = 1\}$ the gap decreased below the 0.9 threshold and the algorithm responded by attaching the 11ah link to the bond. The throughput gap continued to decrease to zero at an applied load of exactly $\{n_{11ac} = 76, n_{11ah} = 50\}$ and then increased non-linearly to just over 0.3 at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 50\}$.

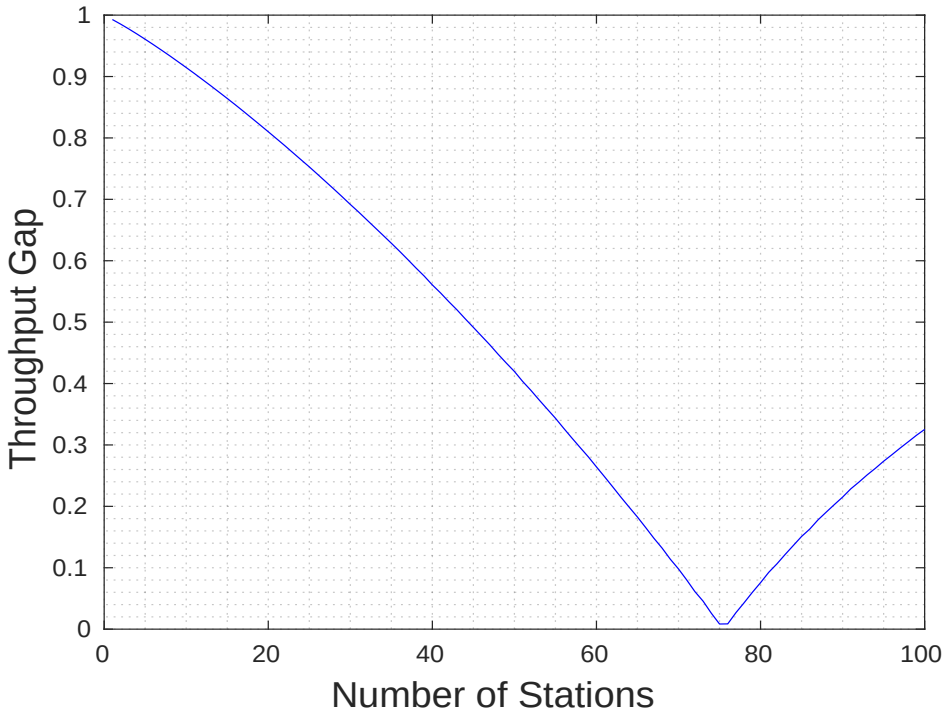

Figure 4.46: Test 2C Nam Algorithm Bond Throughput Gap Results

As per the analysis of Test 1C above, in Test 2C we saw the Nam algorithm respond to the bond throughput gap reducing below the 0.9 threshold and how it reacts when the fastest bond link changes from one slave to the other. From the bond delay shown in Figure 4.44 in the previous results we saw the behaviour change as the applied load reached $\{n_1 = 75, n_2 = 50\}$ but this was not prompted the throughput gap threshold being exceeded but instead by a change in the relative throughput performance of the individual slave links. As seen in Figure 4.47 below for applied loads of $\{n_{11ac} < 75, n_{11ah} = 50\}$ the slower slave device was the 11ah link with an applied load of n = 50, but for loads $\{n_{11ah} > 75, n_{11ah} = 50\}$ the slower link was the 11ac link with the variably applied load. This is of consequence for the test results because the Nam algorithm states that when the throughput gap exceeds 0.9 then only the *fastest* single link of the bond should be used which changed as the applied load exceeded $\{n_{11ac} = 75, n_{11ah} = 50\}$.

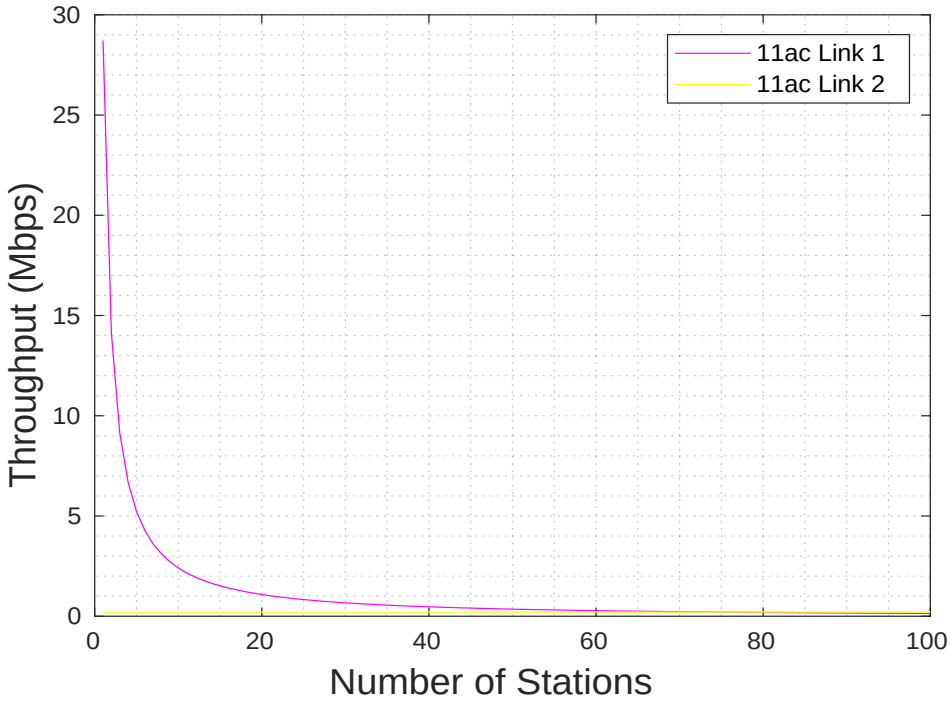

Figure 4.47: Test 1C Nam Algorithm Link Throughput Comparison

4.4.2.7 Nam Test 2D: Results

The bond throughput results for the Nam algorithm under the balanced load scenario in Test 2D are presented in Figure 4.48 below. The achieved Nam bond throughput is plotted as the dashed red line and the maximum theoretical bond throughput as the dotted black line. The overall mean achieved throughput for the heterogeneous 11ac-11ah bond for all applied loads used was 1.175 Mbps. Both throughput metrics decreased non-linearly as the applied load was raised. At the initial load of $\{n_{11ac} = 1, n_{11ah} = 100\}$ the achieved bond throughput and the maximum throughput were both approximately 28.74 Mbps. As the load increased to $\{n_{11ac} = 25, n_{11ah} = 100\}$ the achieved bond throughput reduced to 0.831 Mbps and the maximum throughput to 0.914 Mbps as the difference between the two metrics became more noticeable. Notably at $\{n_{11ac} = 26, n_{11ah} = 100\}$ the two metrics further diverged with the maximum throughput continuing its previous trajectory to 0.878 Mbps while the achieved bond throughput decreasing more significantly to 0.4 Mbps. Both metrics then continued to decrease non-linearly as the applied load was further raised. Between $\{n_{11ac} = 26, n_{11ah} = 100\}$ and the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$ the achieved bond throughput reduced non-linearly to 0.165 Mbps while the maximum theoretical bond throughput decreased to 0.219 Mbps. The overall mean link throughput difference during the test was 0.17827 Mbps.

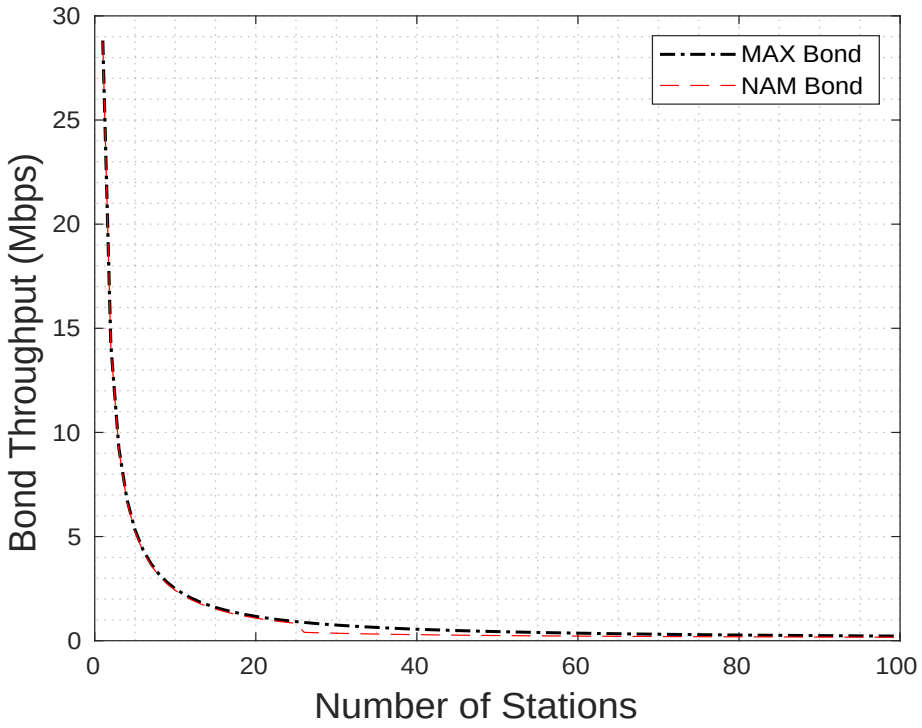

Figure 4.48: Test 2D Nam Algorithm Bond Throughput Results

The link bandwidth utilisation results for the Nam algorithm in Test 2D are shown in Figure 4.49. The overall mean link utilisation for all applied load combinations considered was 71.154%. As seen for applied loads between $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 25, n_{11ah} = 100\}$ the efficiency was approximately 100%. Then as the load incremented to $\{n_{11ac} = 26, n_{11ah} = 100\}$ the output decreased to a minimum of 0.4562 before increasing non-linearly as the load was further raised to a final value of 74.5% at the heaviest prescribed load of $\{n_{11ac} = 100, n_{11ah} = 100\}$.

For applied loads between $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 25, n_{11ah} = 100\}$ the performance disparity between links was large and the bond throughput gap was greater than 0.9 Here the Nam algorithm dictates that only a single link should be active in the bond, resulting to a bandwidth utilisation of 100%. Then as the load incremented to $\{n_{11ac} = 26, n_{11ah} = 100\}$ the disparity between the two links reduced and the bond throughput gap dropped below the 0.9 threshold, to which the algorithm responded by re-attaching the inactive bond link, also causing a sudden drop in the bandwidth utilisation. As the load further increased the performance disparity further reduced, which had the effect of gradually increasing the bandwidth utilisation.

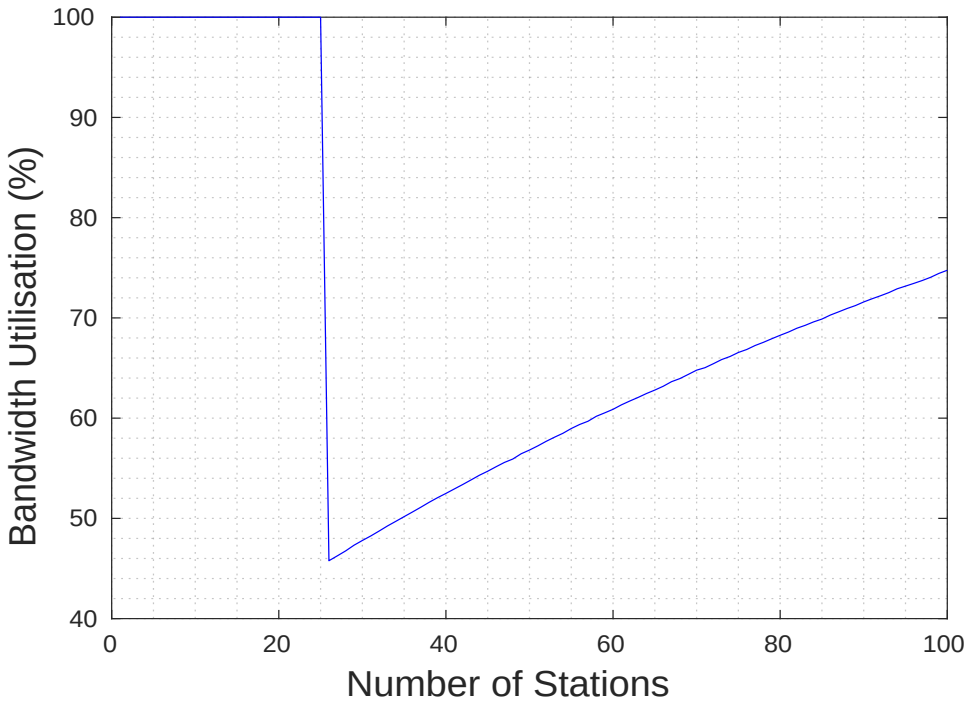

Figure 4.49: Test 2D Nam Algorithm Link Bandwidth Utilisation Results

The bond delay results for the Nam algorithm in Test 2D are presented in Figure 4.50. The overall mean bond delay for all applied load combinations considered was 193429.64 μs. For applied loads between $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 25, n_{11ah} = 100\}$ the bond delay followed the 11ac slave delay and increased non-linearly from 0.00284 μs to approximately 19100 μs. However as the load incremented to $\{n_{11ac} = 26, n_{11ah} = 100\}$ the bond delay began to follow the 11ah link and increased to 196000 μs and remained constant as the load was further raised towards $\{n_{11ac} = 100, n_{11ah} = 100\}$. The mean link delay difference for all applied load combinations considered was 98549.15 μs.

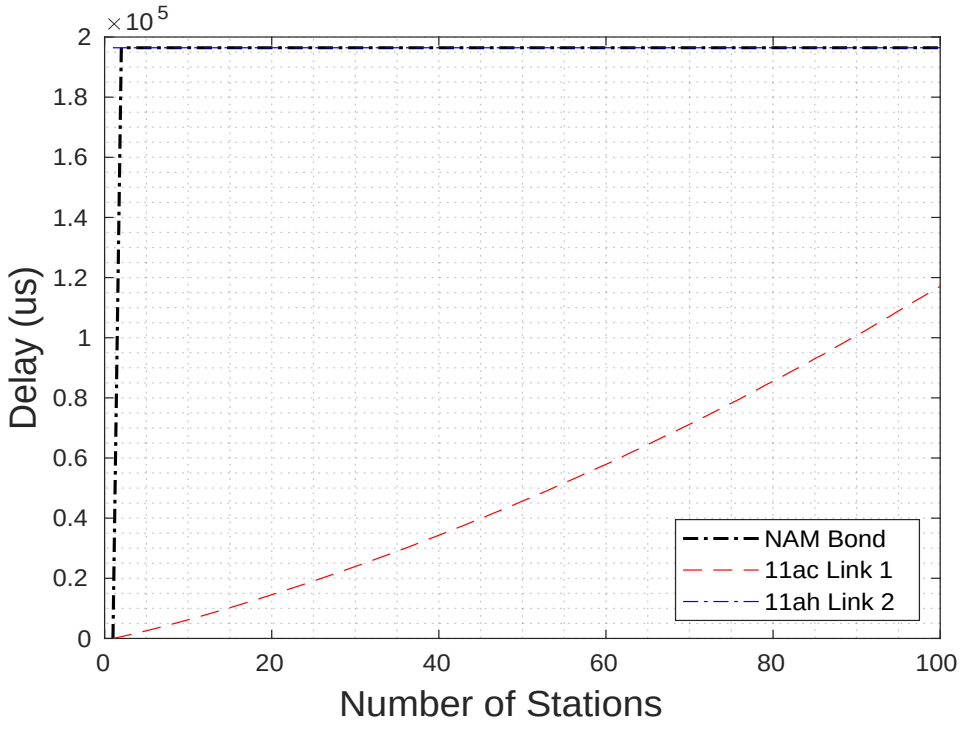

Figure 4.50: Test 2D Nam Algorithm Bond Delay Results

4.4.2.8 Nam Test 2D: Discussion

In Test 2D the main finding was that the dynamic link selection algorithm by Nam increased the overall mean bond throughput by 48% and increased link utilisation by 27.83% when compared to the traditional static round robin approach. The mean bond throughput under the Nam algorithm was 1.1755 Mbps with a mean bond throughput efficiency of 71.154% while the means achieved under round robin were 0.5135 Mbps and 74.197% respectively. The throughput of both approach is compared alongside the maximum theoretical bond throughput in Figure 4.51 where the difference in behaviour is evident at applied loads from $\{n_{11ac} = 1, n_{11ah} = 100\}$ to $\{n_{11ac} = 25, n_{11ah} = 100\}$. During this range the output of both methods decreased exponentially. The achieved Nam algorithm throughput was close to the maximum theoretical bond throughput while the round robin throughput was significantly less. As the load was incremented to $\{n_{11ac} = 26, n_{11ah} = 100\}$ the Nam algorithm throughput suddenly reduced to the same value as round robin and remained approximately constant as the load was further raised towards the heaviest prescribed level of $\{n_{11ac} = 100, n_{11ah} = 100\}$.

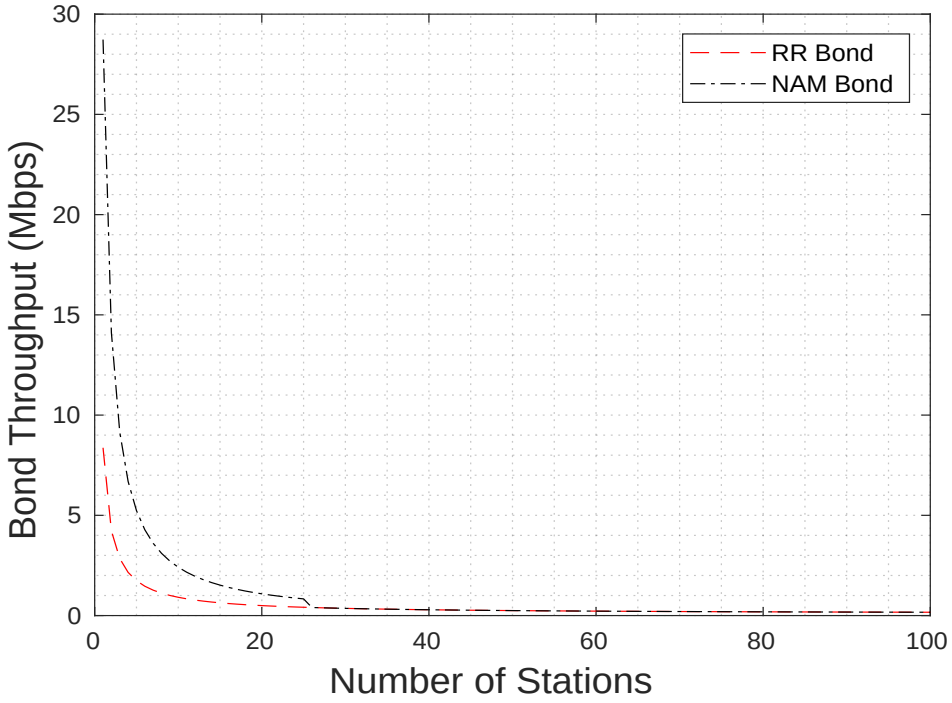

Figure 4.51: Test 2D Nam vs RR Bond Throughput Comparison

In Test 2D we saw the Nam algorithm respond to the bond throughput gap decreasing below the specified threshold at an applied load of $\{n_{11ac} = 26, n_{11ah} = 1\}$. For low loads between $\{n_{11ac} = 1, n_{11ah} = 1\}$ and $\{n_{11ac} = 25, n_{11ah} = 1\}$ only the first 11ac link of the bond was in use. During this range as the load was raised the performance disparity between the two slaves was reduced until eventually when incremented to $\{n_1 = 26, n_2 = 1\}$ the measured bond throughput gap reduced below the 0.9 threshold. The Nam algorithm reacted by re-attaching the second 11ac link to the bond thereby increasing the overall mean bond throughput but decreasing the mean bond throughput efficiency. As seen from the plot in Figure 4.52 below the bond throughput gap continued to decrease to approximately 0.4 at an applied load of $\{n_{11ac} = 100, n_{11ah} = 50\}$ and crucially did not rise again above the prescribed threshold.

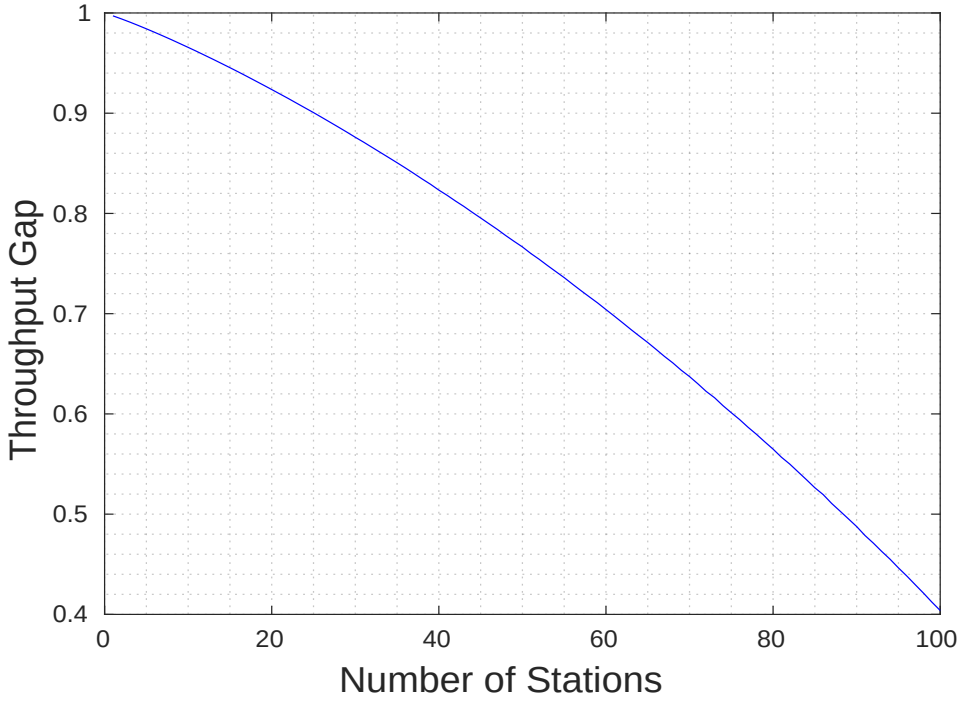

Figure 4.52: Test 2D Nam Algorithm Bond Throughput Gap Results

4.5 Nam Results Evaluation

This section provides a detailed evaluation of the Nam algorithm performance results from Tests 1A-1D and 2A-2D for both the homogeneous 11ac-11ac and heterogeneous 11ac-11ah bonds.

Across all eight individual tests the throughput-gap-based dynamic link selection outperformed the static bond configuration by 26.81% in terms of the mean bond throughput and by 7.28% in terms of mean link bandwidth utilisation. The overall mean Nam bond throughput was 6.48 Mbps with a mean bandwidth utilisation of 87.32% compared to 5.09 Mbps and 81.4% for static round robin. The results are discussed in further detail for each bond type in the following sections.

4.5.1 Homogeneous 11ac-11ac Bond

For the homogeneous 11ac-11ac bond there was a significant improvement using the dynamic approach by Nam when compared to the traditional static bond configuration.

Taking an average of Tests 1A-1D the Nam algorithm outperformed static round robin by 12.19% in terms of the mean link bandwidth utilisation and by 42.14% in terms of the achieved bond throughput. The overall mean bond throughput was 8.649 Mbps with a link bandwidth utilisation of 94.932% compared to a mean throughput of 6.085 Mbps and utilisation of 84.62% for static round robin.

The mean simulation results for each test are summarised in Table 4.2 below, along with the 99.9% confidence interval that allows us to say that for 99.9% of simulations run, the true mean is likely to fall within the specified range. Note that direct comparison of the bond throughput can be misleading due to the different applied loads and individual link throughput in each test. The link bandwidth utilisation therefore provides the best measure of the bond efficiency and the size of the impact of the inherent slow-down effect due to the in-order delivery requirement. For example the best bandwidth utilisation was seen in Test 1A with a mean of nearly 100% while the worst was in Test 1D with 88.39% where a severe throughput slow-down was observed.

	Test 1A	**Test 1B**	**Test 1C**	**Test 1D**
Bond T.put (Mbps)	2.478 ±0.1037	29.33 ±0.407	1.4754 ±0.019	1.273 ±0.017
T.put Diff. (Mbps)	0.0018 ±4 * 10^{-6}	1.5502 ±0.06	0.3284 ±0.0053	0.218 ±0.008
B.width Util. (%)	99.898 ±0.008	99.507 ±0.004	91.932 ±0.006	88.394 ±0.009
Bond Delay (us)	50257 ±597	181.663 ±29.5	62142 ±678	115962 ±1115
Delay Diff. (us)	106 ±33	181.66 ±29.4	26525 ±378	49014 ±546

Table 4.2: Summary of Dynamic Nam Algorithm Results for Homogeneous 11ac-11ac Bond

In general the dynamic link selection algorithm by Nam was found to significantly improve the overall bond throughput and link bandwidth utilisation when compared to the traditional static round robin configuration. However as shown in the table above there was still a significant

throughput degradation in Tests 1B to 1D due to the performance disparity between the individual 802.11 bond links and the associated slow-down effect.

Furthermore it worth commenting that despite being originally designed for use at the Transport Layer over end-to-end interface bonds using MPTCP, the throughput-gap-based decision-making approach by the authors Nam et al. provided a reasonable overall performance at the Link Layer for the point-to-point 802.11 interface bonds assumed in the simulation testing.

Comparing the dynamic Nam algorithm performance against the results of the static round robin bond configuration in Chapter 3, the main findings from Tests 1A-1D are as follows:

1. Test 1A: In the benchmark balanced load scenario the dynamic Nam algorithm performance was equivalent to the static round robin approach for all applied loads considered with a mean link utilisation of 99.898% and a mean bond throughput of 2.478 Mbps. As seen in in Chapter 3 for the homogeneous 11ac-11ac and 65Mbps-65Mbps interface bond the balanced loads used created the ideal conditions for the 802.11 bond throughput and so no significant slow-down effect was observed. Therefore the dynamic Nam algorithm threshold for removing problematic bond links was never reached and the resulting performance was equivalent to the static bond configuration throughout the test.

2. Test 1B: In the first unbalanced load test the dynamic Nam algorithm outperformed the static round robin approach by 53.02% in terms of the overall bandwidth utilisation and by 23.83% in terms of the mean achieved bond throughput. Here the throughput gap threshold of 0.9 was exceeded for applied loads of $\{n_1 >= 8, n_2 = 1\}$ where the dynamic link selection algorithm reverted to the use of the single fastest link and by doing so significantly increased the overall performance compared to the static bond configuration.

3. Test 1C: In the second unbalanced applied load scenario the Nam algorithm outperformed static round robin by 2.53% in terms of bandwidth utilisation and also increased the mean achieved bond throughput by 18.77%. Notably the bond throughput gap threshold was exceeded for $\{n_1 <= 7, n_2 = 50\}$ where the Nam algorithm switch to the single fastest link thereby significantly increasing the overall performance compared to the static round robin bond.

4. Test 1D: In the final unbalanced load test the dynamic link selection algorithm outperformed the static round robin method by 8.44% in terms of the overall bandwidth utilisation and by 48.3% in terms of the mean bond throughput. During the test the algorithm throughput gap threshold of 0.9 was exceeded for applied loads of $\{n_1 <= 7, n_2 = 50\}$ for which the algorithm reverted to the use of the single faster link and was again able to significantly increase the overall bond throughput and link bandwidth utilisation

For all unbalanced scenarios in Tests 1B-1D the dynamic link selection algorithm by Nam was able to mitigate the inherent slow-down effect for heterogeneous link performance due to the in-order delivery requirement and thereby significantly increase the overall mean bond throughput and mean link bandwidth utilisation when compared to the traditional static round robin bond configuration.

Note that a non-negligible bandwidth wastage still occurred for Tests 1B-1D which was due primarily to the inefficiency and inflexibility of the static round robin bond used. A certain amount of waste is tolerated and only when the bond throughput gap increases above the 0.9 threshold does the algorithm act by removing the problematic and slower slave link.

Also note that the dynamic Nam algorithm outperformed the static round robin approach in terms of the bond throughput *and* the bandwidth utilisation in *all* four tests 1A-2D for the homogeneous 11ac-11ac bond, which was in contrast to the heterogeneous 11ac-11ah bond in tests 2A-2D in the following section.

While the algorithm significantly increased the overall bond throughput and bandwidth utilisation when compared to the static bond, the performance was not robust for *all* applied bond in the evaluation. For example in Test 1B for applied loads between $\{n_1 = 7,\ n_2 = 50\}$ and $\{n_1 = 9,\ n_2 = 50\}$ the Nam algorithm failed to guard against the counter productive scenario where the aggregate bond throughput is reduced below that of the fastest 802.11 link. The condition can be seen in the Figure 4.53 below which shows the Nam bond throughput plotted alongside that of the fastest single 802.11 link.

The reason for this lies in the main decision-making metric, the bond throughput gap, which is a normalised value representing the difference in throughput between the two bond 802.11 links in proportion to the link with the largest throughput. While this provides a good general performance in the simulations, a lack of knowledge with respect to the predicted bond throughput means that the algorithm cannot provide protection against the aforementioned counter productive scenario, limiting its efficiency at certain applied loads. The problem was also seen in Tests 1C and 1D as shown in Table 4.3 below.

Applied Load Scenario	**Problematic Applied Loads**	**Bond Throughput Gap**
Test 1B	$\{n_1 = 7,\ n_2 = 50\}$ to $\{n_1 = 9,\ n_2 = 50\}$	0.8745 to 0.9055
Test 1C	$\{n_1 = 7,\ n_2 = 50\}$ to $\{n_1 = 15,\ n_2 = 50\}$	0.8858 to 0.7666
Test 1D	$\{n_1 = 16,\ n_2 = 38\}$ to $\{n_1 = 9,\ n_2 = 50\}$	0.894 to 0.7225

Table 4.3: Problematic Homogeneous 11ac-11ah Bond Loads for Nam Algorithm

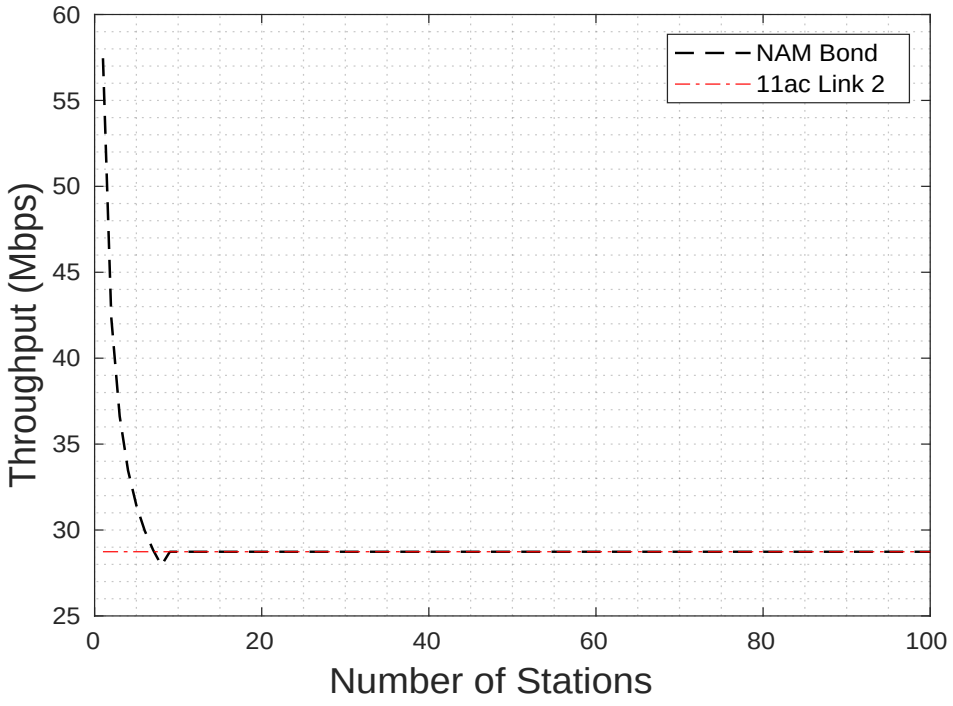

Figure 4.53: Test 1B Nam Algorithm Bond Throughput vs Fastest Single Link Throughput

4.5.2 Heterogeneous 11ac-11ah Bond

In the Nam evaluation for the heterogeneous 11ac-11ah interface bond there was a smaller but still substantial improvement from the dynamic link selection approach compared to the static round robin configuration. Across all four tests conducted the throughput-gap-based dynamic link selection algorithm increased the overall mean bond throughput by 5.61% and the mean link bandwidth utilisation by 1.23%. The mean bond throughput achieved by Nam was 4.33 Mbps with a mean link bandwidth utilisation of 79.718%. By comparison the mean bond throughput for round robin was 4.1 Mbps with a mean bandwidth utilisation of 78.75%.

As seen in Chapter 3 the difference in DCF parameters at the MAC layer between the 11ac and 11ah devices creates a significant disparity in the resulting link throughput performance, which leads to a non-negligible bond throughput slow-down even for the same raw PHY data rate of 65 Mbps and identical balanced applied load.

While overall there still was a substantial benefit the dynamic link selection algorithm by Nam did not outperform static round robin in terms of both bond throughput and link utilisation in all four tests. A minor trade-off was seen in Test 2B where the Nam algorithm sacrificed 0.71% of the achieved bond throughput in order to increased the link bandwidth utilisation by 1.75%, with a more significant trade-off in Test 2D. However in general the Nam algorithm performance was reasonable and was able to increase the achieved bond throughput compared to the static round robin configuration, although worse compared to the homogeneous dual-11ac case.

	Test 2A	**Test 2B**	**Test 2C**	**Test 2D**
Bond T.put (Mbps)	1.2999 ±0.017	13.6157 ±0.8	1.2297 ±0.03	1.1755 ±0.0144
T.put Diff. (Mbps)	0.59 ±0.063	1.5315 ±0.062	0.253 ±0.004	0.17827 ±0.009
B.width Util. (%)	72.9 ±0.0078	98.059 ±0.009	76.76 ±0.008	71.154 ±0.0079
Bond Delay (us)	85313 ±1601	501978 ±1567	57953 ±554	193430 ±1652
Delay Diff. (us)	35113 ±598	838.1033 ±11	12682 ±249	98549 ±989

Table 4.4: Summary of Dynamic Nam Algorithm Results for Heterogeneous 11ac-11ah Bond

When comparing the results for the dynamic Nam algorithm against the results for the static round robin bond in Chapter 3 under similar applied load scenarios, the main findings from Tests 2A-2D for the heterogeneous 11ac-11ah as summarised as follows:

1. Test 2A: In the balanced load test the dynamic Nam algorithm matched the static round robin approach with a mean utilisation of 72.9% and a mean bond throughput of 1.299 Mbps. As seen from the round robin results in Chapter 3 in Test 2A there was a significant disparity between the individual 11ac and 11ah links even for the balanced applied loads and 65Mbps-65Mbps PHY rates used. This created a substantial slow-down effect due to the in-order delivery requirement but the bond throughput gap threshold of 0.9 was not exceeded, and therefore dynamic Nam algorithm performance was equivalent to the static bond configuration.

2. Test 2B: In the first unbalanced load test the dynamic Nam algorithm under-performed the static round robin approach by 0.71% in terms of the overall bandwidth utilisation but outperformed by 1.75% in terms of the mean achieved bond throughput. In this example the difference in applied load between the two links unbalanced the bond and created a substantial slow-down effect. This was such that the bond throughput gap was exceeded for applied loads where $\{n_{11ac} >= 17, n_{11ah} = 1\}$ and the algorithm reverted to the use of a single 802.11 link in order to increase the mean bond throughput and link bandwidth utilisation.

3. Test 2C: In the second unbalanced load test the dynamic Nam algorithm outperformed the static round robin bond by 66.04% in terms of the mean achieved throughput and by 8.88% in terms of the link bandwidth utilisation. Notably the Nam algorithm reverted to the use of a single link when the bond throughput gap exceeded the specified threshold for applied loads where $\{n_{11ac} <= 11, n_{11ah} = 50\}$ and by doing so substantially increased the bond throughput performance compared to the static bond configuration.

4. Test 2D: In the final unbalanced load test for heterogeneous 11ac-11ah interface bond the dynamic Nam algorithm under-performed the static round robin method by 4.28% in terms of the overall bandwidth utilisation but outperformed it in terms of the link bandwidth utilisation by 129.92%. In the test the Nam algorithm switched to the single fastest link between applied loads where $\{n_{11ac} <= 25, n_{11ah} = 1\}$ which again increased the link bandwidth utilisation significantly while reducing the achieved bond throughput by a small amount.

As with the homogeneous 11ac-11ac bond tests in all tests at multiple stages of the simulation the achieved Nam bond throughput was reduced below that of the single fastest 802.11 link due to the inherent slow-down effect, therefore rendering the use of the technique counter productive to the aim of increasing the overall link throughput. Notably this occurred far more often for the heterogeneous 11ac-11ah compared to the dual-11ac bond, which is shown in Table 4.5 below, with Tests 2C and 2D being particularly impacted by the condition.

Applied Load Scenario	**Problematic Applied Loads**	**Bond Throughput Gap**
Test 2A	$\{n_{11ac} = 1, n_{11ah} = 1\}$	0.5421
Test 2B	$\{n_{11ac} = 1, n_{11ah} = 1\}$	0.5421
Test 2C	$\{n_{11ac} = 11, n_{11ah} = 50\}$ to $\{n_{11ac} = 41, n_{11ah} = 50\}$	0.8948 to 0.5472
Test 2D	$\{n_{11ac} = 25, n_{11ah} = 100\}$ to $\{n_{11ac} = 82, n_{11ah} = 100\}$	0.8957 to 0.5498

Table 4.5: Problematic Heterogeneous 11ac-11ah Bond Loads for Nam Algorithm

4.6 Improving Bandwidth Utilisation

This section presents a novel predicted-throughput-based dynamic link selection algorithm for 802.11 interface bonding, designed specifically to avoid the counter productive scenario where the bond throughput is reduced below the throughput of the fastest single link, as observed in the previous static bond configuration experiments and also for the dynamic link selection method by Nam.

The main decision-making algorithm is implemented on top of the 802.11 interface bonding framework given in Chapter 3 and is evaluated using the same set of applied load scenarios used for the contemporary approach by Nam and the previous static round robin bond configuration.

The MatLab implementation of the dynamic link selection algorithm is described using pseudocode in Section 4.6.1.1, and the simulation input parameters and channel load scenarios used are described in Section 4.6.1.2 and Section **??** respectively. The simulation results are presented in Section 4.7 as a function of the applied load, and a evaluation is given in Section 4.8 with a detailed comparison between the performance of the MT algorithm and the contemporary throughput-gap-based approach.

4.6.1 Experimental Setup

This section describes the methodology and experiment setup used for the evaluation of the novel, predicted-throughput-based algorithm.

4.6.1.1 MT Algorithm MatLab Implementation

This section describes the implementation and operation of a link selection algorithm designed to maximise the overall mean achieved throughput by removing problematic bond link whenever the predicted bond throughput is less than the throughput of the single fastest 802.11 slave.

We begin by asking a straightforward question: assuming that a mobile application user has a need or desire to use link aggregation over multiple 802.11 Wireless LAN interfaces, then given a set of specific link states, i.e. the current achievable link latencies and data rates, will the use of round robin bonding be productive in terms of the overall throughput performance? The ideal interface selection algorithm should avoid inefficient or counter productive use of bonding by reverting to a single interface when necessary and use both interfaces only when bonding is viable, i.e. when there is a predicted benefit in terms of the aggregated data rate. To help answer this question we apply the predicted bond throughput from the round robin experiments in the previous chapter to provide the necessary thresholds for the dynamic interface selection decision-making framework using. Please note that the actual mechanism of bond creation is not considered in this analysis, only the decision-making logic.

Using the previously validated predictive model, an algorithm for intelligent and dynamic interface selection is constructed as a simple binary choice as to whether bonding should be used or not. In the proposed algorithm, if the predicted compound throughput of the bond is greater than that of the fastest single interface, then bonding is used, but if the predicted throughput is less than that of the fastest single 802.11 device, then only the fastest single link is used. With this approach the inefficient and wasteful use of bonding is avoided. i.e. the technique is only applied when there is a measurable performance benefit. This is in contrast the static and manual bond configuration approach used in most round robin implementations such as Linux Bonding and similar to the throughput-gap-based approach by Nam.

Algorithm 4: Maximum Throughput Link Selection Algorithm Pseudo-code

```
Data: S_1, S_2
Result: BondInterfaces[]
S_pred;
for all possible combinations of nodes do
    S_pred = calcS_pred(S_1, S_2);
    if (S_pred < S_1) or ((S_pred < S_2) then
        fast = max(S_1, S_2)
        slow = min(S_1, S_2)
        BondInterfaces[fast] = 1;
        BondInterfaces[slow] = 0;
    else
        BondInterfaces[0] = 1; BondInterfaces[1] = 1;
```

Pseudo-code for the proposed dynamic interface selection approach is shown above in Algorithm 4. The link selection algorithm is implemented in MatLab on top of the round robin evaluation framework presented in Chapter 3.

Stepping through each line of the pseudocode, the algorithm input parameters are S_1 and S_2 which are the respective throughputs of each individual link. The output, BondInterfaces[], is a vector representing the newly assigned state of each link of the bond. In the vector a 0 represents an inactive interface and a 1 represents an active interface, i.e. the interfaces indicated by 0 do not form part of the dynamically configured interface bond.

All possible node combinations are considered and a decision is made in each case as to which interfaces should be included in the bond. The throughput of each link are passed to the $calcS_{pred}$ subroutine, which uses them to estimate the predicted bond throughput S_{pred} using the statistics from the static round robin study. In the *if* conditional statement the value of S_{pred} is compared to the throughput of the individual links. If S_{pred} is less than the throughput of either link, then the max() and min() subroutines are used to find the fastest and slowest links. In the following lines the state of the faster link is set to 1 (active) and the state of the slower link is set to 0 (inactive). In the event that *if* conditional results in a 0 then both are set to 1 meaning that both 802.11 interfaces will be attached to the bond.

4.6.1.2 Simulation Input Parameters

Again to allow a direct comparison, the simulation input parameters used are identical to those used in the evaluation of the traditional static link selection approach in Section 3.4 of Chapter 3, and are also the same as those used for the Nam evaluation in Section 4.3.1.2 of the current chapter.

4.6.1.3 Test Scenarios for Homogeneous 11ac-11ac Bond

First we describe the tests for the homogeneous 11ac-11ac interface bond.

4.6.1.3.1 MT Test 1A: Balanced Load Scenario The first MT test for the dual-11ac bond uses the balanced applied load shown in Figure 4.54 below which is similar to RR Test 1A in Chapter 3. The number of nodes in each 11ac WLAN is incremented simultaneously from 1 to $n = 100$. At each stage as per the MT algorithm the predicted bond throughput is calculated using the individual link states and the statistics from the experiment in Chapter 3, and if less than that of the single fastest link then the slower of the two links is removed.

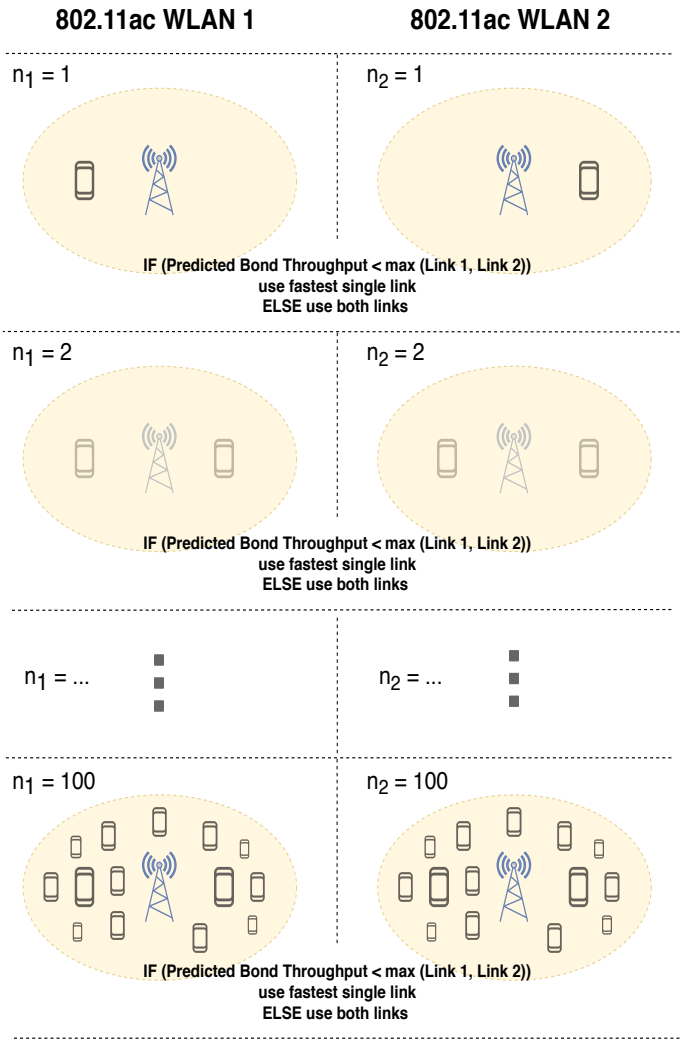

Figure 4.54: MT Test 1A 11ac-11ac Bond Balanced Load Scenario

4.6.1.3.2 MT Test 1B: Unbalanced Scenario (Low Load) The second test for the MT algorithm is similar to RR Test 1B where there is a light static load of n = 1 on the second 11ac link and the load on the first link is varied from n = 1 to n = 100. In MT Test 1A the predicted bond throughput is calculated for each combination of applied load and if less than the throughput of the single fastest link then the problematic slave is removed.

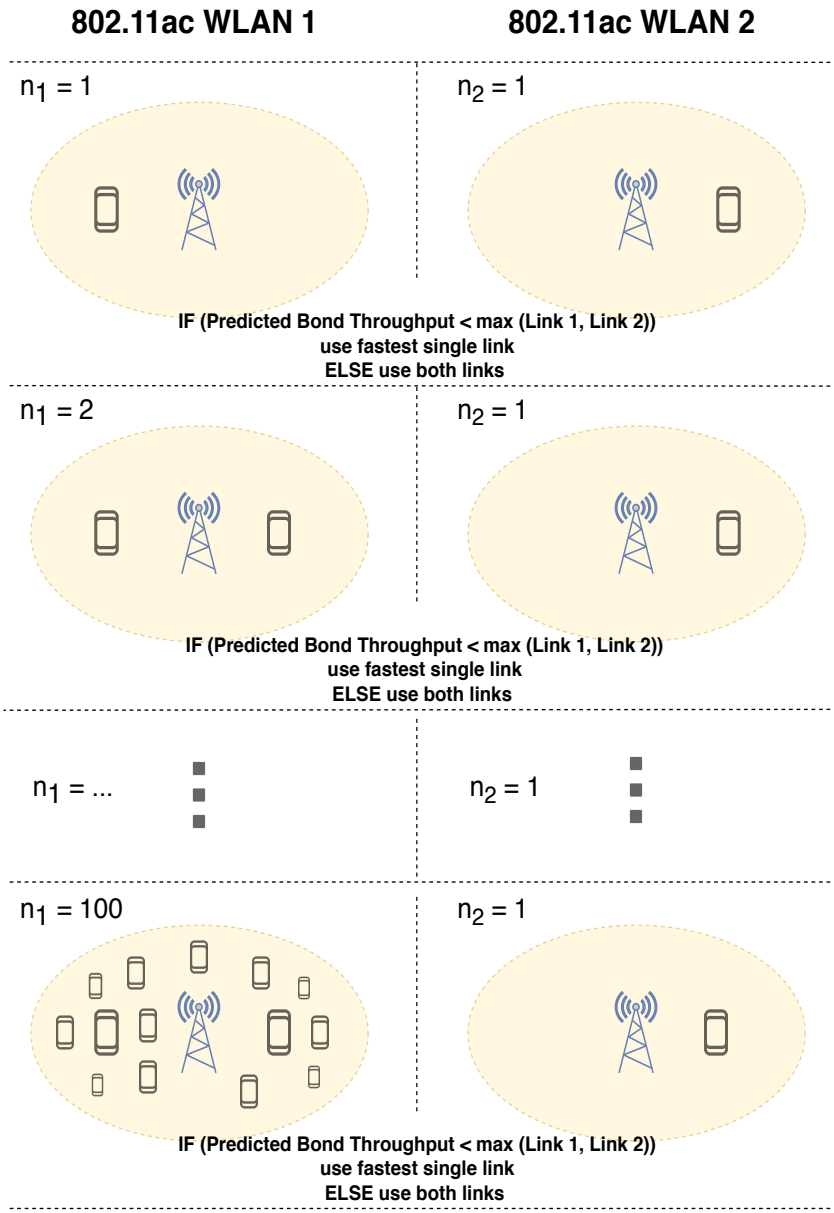

Figure 4.55: MT Test 1B 11ac-11ac Bond Unbalanced Load Scenario (Low Load)

4.6.1.3.3 MT Test 1C: Unbalanced Scenario (Moderate Load) The next MT test is the unbalanced load scenario for the homogeneous 11ac-11ac bond shown in Figure 4.56 below. The applied load on the second 11ac link is held constant at n = 50 while the load on the first 11ac link is incremented from $n_1 = 1$ to $n_1 = 100$. Again the predicted bond throughput is calculated at each iteration and if less than that of the single fastest link the slower slave is removed.

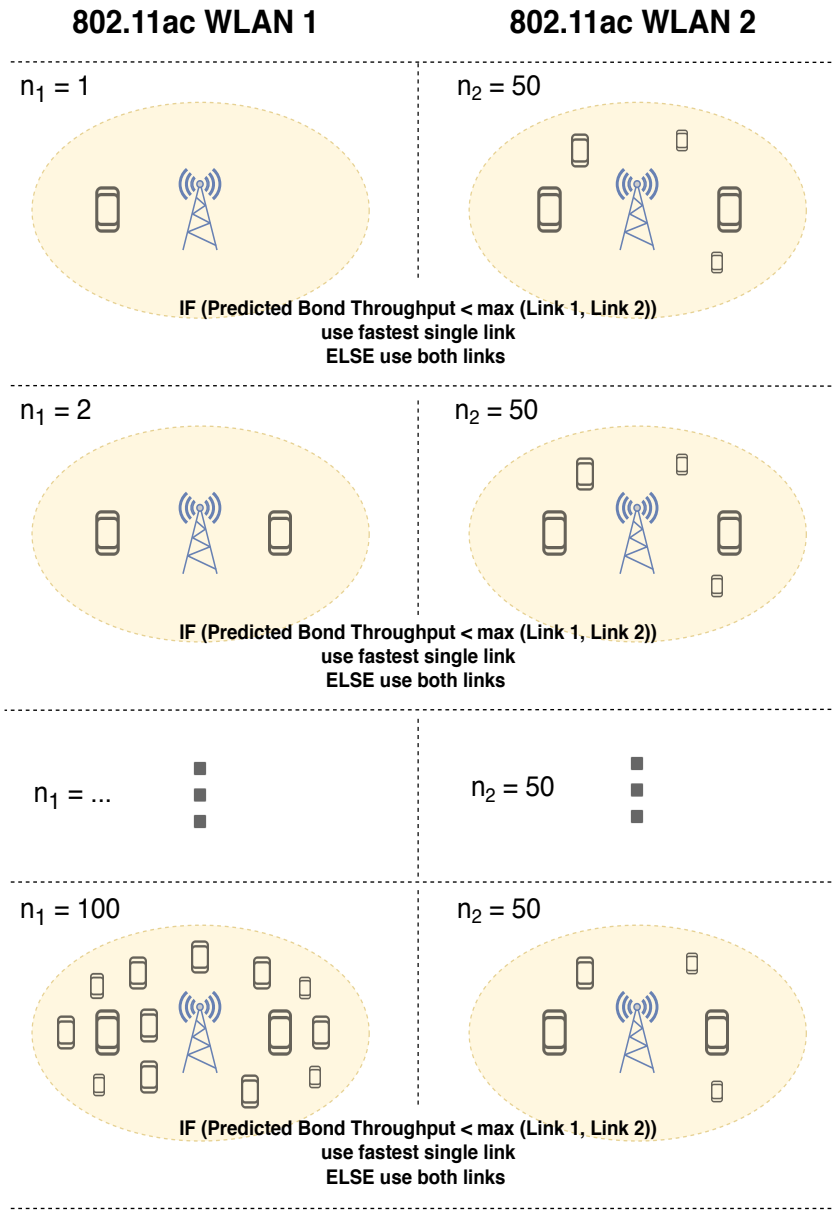

Figure 4.56: MT Test 1C 11ac-11ac Bond Unbalanced Load Scenario (Moderate Load)

4.6.1.3.4 MT Test 1D: Unbalanced Scenario (Heavy Load) The final MT test for the dual-11ac bond is the unbalanced scenario illustrated in Figure 4.57. The static load on the second 11ac link was set to the maximum value of n = 100 while the load on the second link was varied from $n_1 = 1$ to $n_1 = 100$. As per the previous tests at each iteration the predicted bond throughput is calculated using the individual link states and if less than the throughput of the single fastest link the problematic slave is removed.

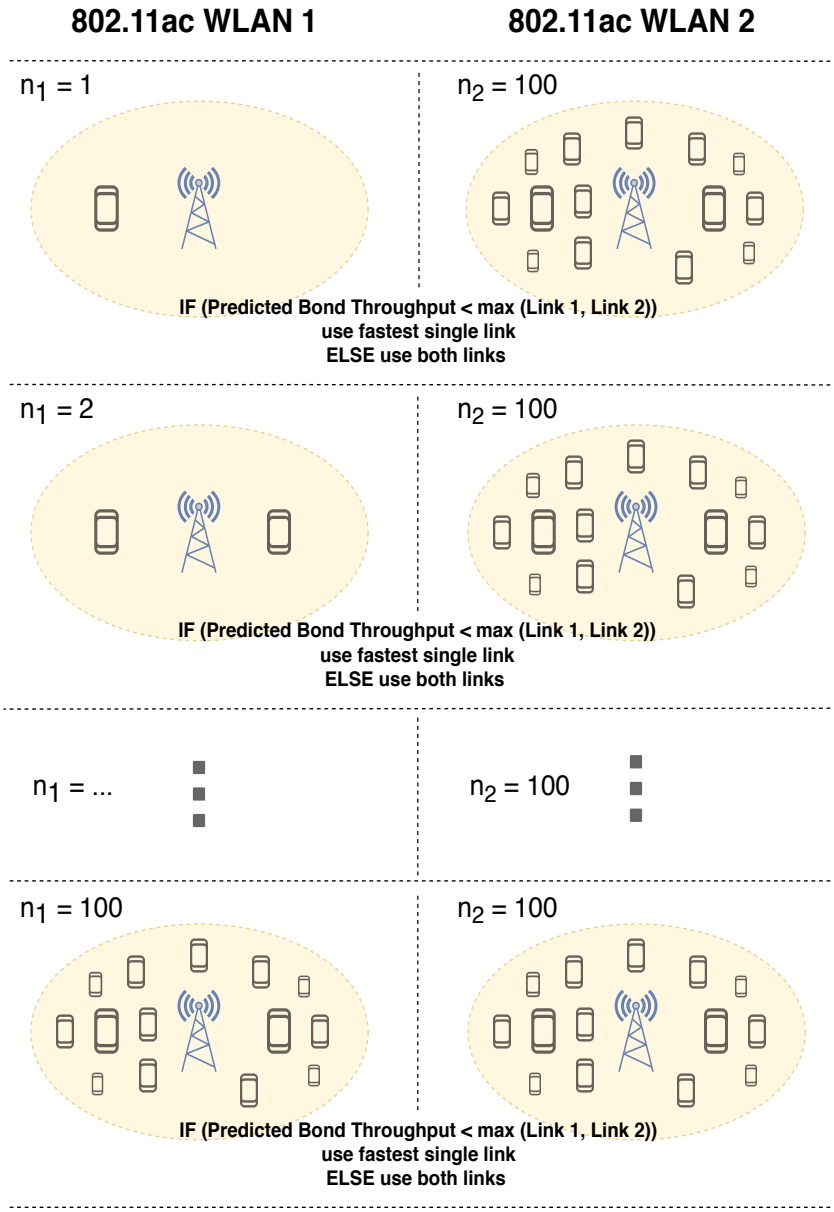

Figure 4.57: MT Test 1D 11ac-11ac Bond Unbalanced Load Scenario (Heavy Load)

4.6.1.4 Test Scenarios for Heterogeneous 11ac-11ah Bond

Next we present the tests used for the heterogeneous 11ac-11ah interface bond.

4.6.1.4.1 MT Test 2A: Balanced Load Scenario In the first MT test for the heterogeneous 11ac-11ah bond the balanced scenario illustrated in Figure 3.18 below is used. At each iteration the applied load on the 11ac and 11ah links is increased simultaneously from $n = 1$ to the maximum of $n = 100$. For each of the applied loads the predicted bond throughput is calculated and if less than that of the single fastest link then the slower of the two links is removed.

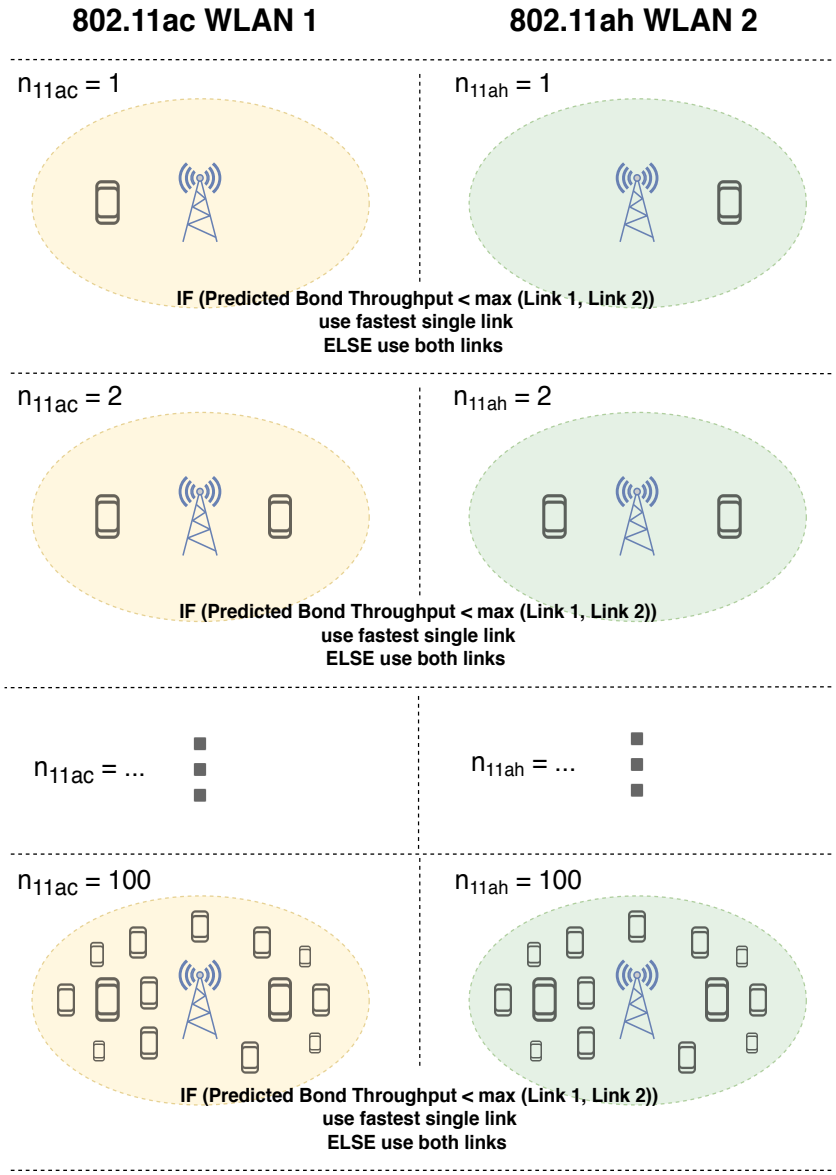

Figure 4.58: MT Test 2A 11ac-11ah Bond Balanced Load Scenario

4.6.1.4.2 MT Test 2B: Unbalanced Scenario (Moderate Load) In the first unbalanced MT test for the heterogeneous 11ac-11ah bond the applied load on the 11ah link is held constant at the low level of $n = 1$ nodes while the load on the 11ac link is varied from $n = 1$ to $n = 100$, as illustrated in Figure 4.59 below. Again the average bond throughput was calculated for each combination of applied link load across the bond. Again as per the algorithm at each iteration the predicted bond throughput is calculated and if less than that of the single fastest link then the problematic link is removed.

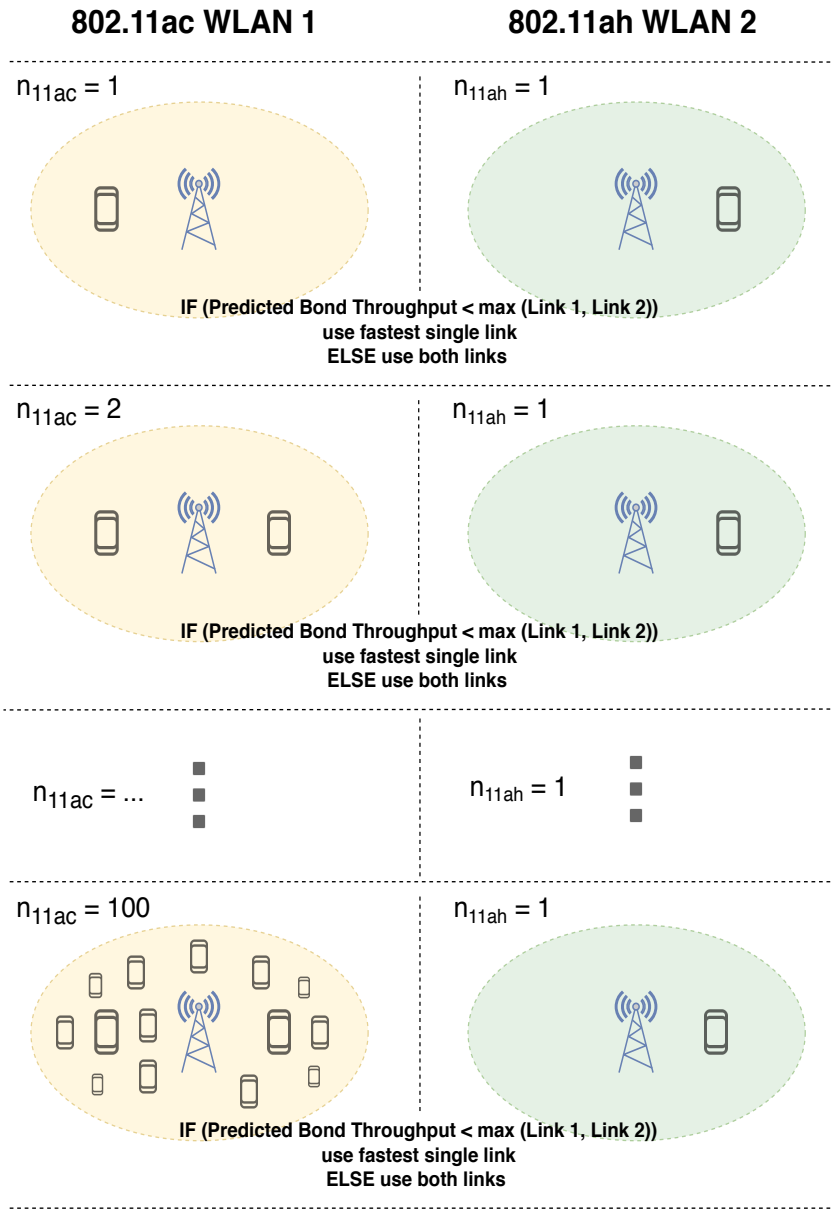

Figure 4.59: MT Test 2B 11ac-11ah Bond Unbalanced Load Scenario (Moderate Load)

4.6.1.4.3 MT Test 2C: Unbalanced Scenario (Moderate Load) The third MT test for the heterogeneous bond is the unbalanced scenario as illustrated in Figure 4.60 where the applied load on the 11ah link is constant at the moderate level of $n = 50$ nodes, while the load on the 11ac link is incremented from $n = 1$ to $n = 100$ nodes. Like the previous tests the predicted bond throughput is calculated and if found to be less than the throughput of the single fastest link then the slower link is removed from the bond.

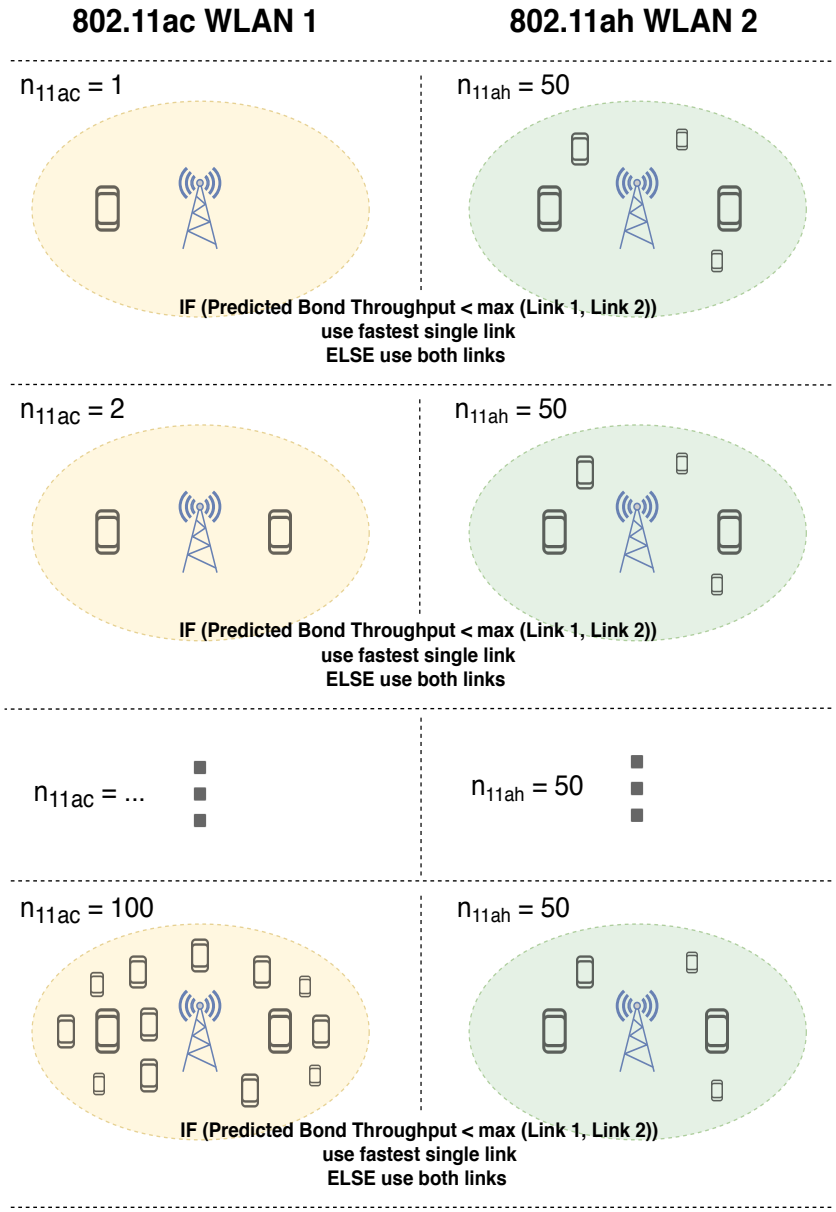

Figure 4.60: MT Test 2C 11ac-11ah Bond Unbalanced Load Scenario (Moderate Load)

4.6.1.4.4 MT Test 2D: Unbalanced Scenario (Heavy Load) The final MT test for the heterogeneous 11ac-11ah bond is the unbalanced scenario show in Figure 4.61 where the applied load on the 11ah link is held constant at $n = 100$ competing nodes while the load on the 11ac link is varied from $n = 1$ to $n = 100$. As per the algorithm, at each iteration the predicted bond throughput is calculated and if less than that of the single fastest link then the problematic slave is removed.

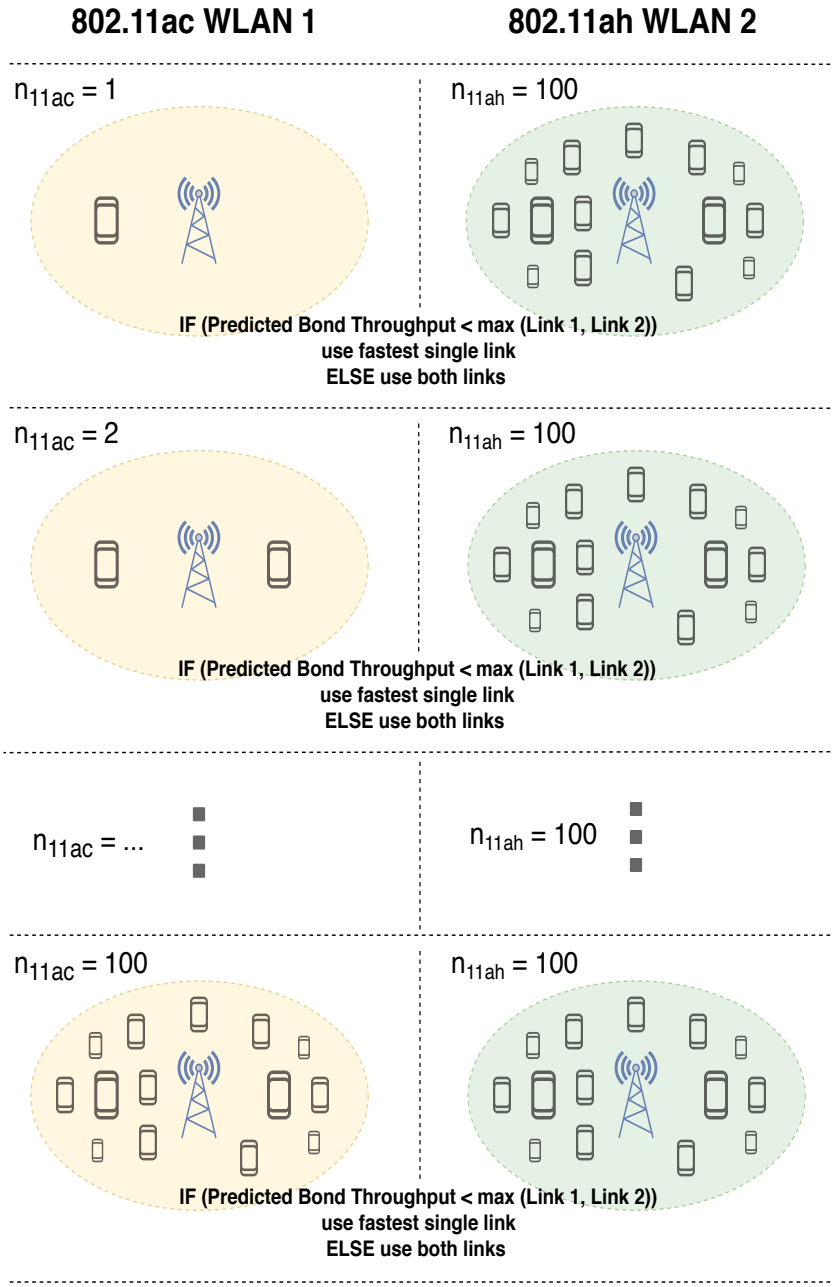

Figure 4.61: MT Test 2D 11ac-11ah Bond Unbalanced Load Scenario (Heavy Load)

4.7 MT Results Presentation

This section presents the results from the evaluation of the predicted-throughput-based dynamic link selection algorithm.

4.7.1 Homogeneous 11ac-11ac Bond

First we present the MT algorithm results for the homogeneous 11ac-11ac interface bond.

4.7.1.1 MT Test 1A: Results

The bond throughput results for the MT algorithm under the balanced load scenario in Test 1A are given in Figure 4.62 below. The overall mean bond throughput for the dynamic MT algorithm was 2.4783 Mbps while the mean theoretical maximum bond throughput was 2.4816 Mbps. The performance of the MT algorithm was approximately equal to the maximum theoretical bond throughput at all applied loads. As the applied bond load was raised both throughput metrics decreased non-linearly from a maximum value of 57.45 Mbps at $\{n_1 = 1, n_2 = 1\}$ to 0.276 Mbps at the heaviest test load of $\{n_1 = 100, n_2 = 100\}$. Compared to the absolute throughputs of the individual 11ac slave links the resulting throughput difference was negligible with an overall mean of 1.8053 kbps.

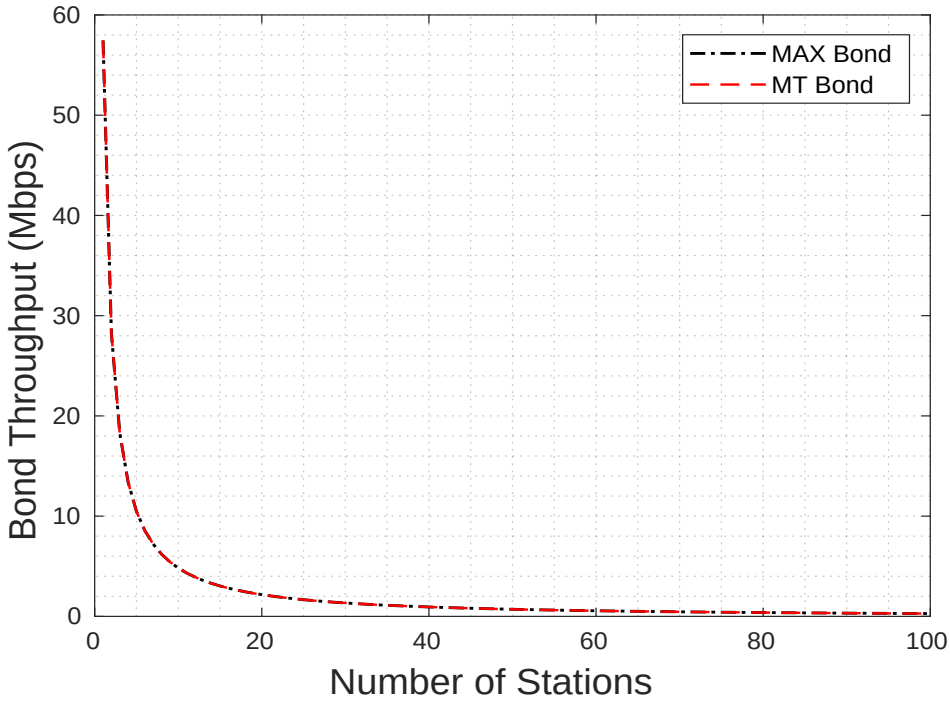

Figure 4.62: Test 1A MT Algorithm Bond Throughput Results

The link bandwidth utilisation results for the MT algorithm under the balanced load scenario in Test 1A are shown below in Figure 4.63 as a function of the applied load. As seen from the graph the achieved 11ac-11ac bond utilisation remained roughly constant at 100% or near with an overall mean of 99.899% and a minimum value of 99.7%.

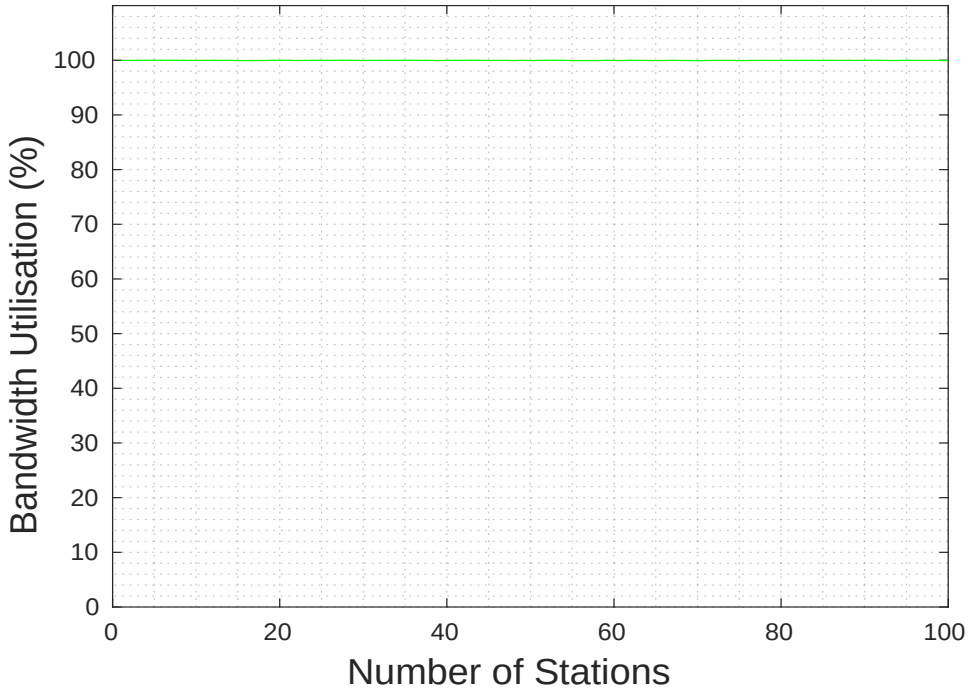

Figure 4.63: Test 1A MT Algorithm Link Bandwidth Utilisation Results

The bond delay results for the MT algorithm in Test 1A are shown in Figure 4.64 as a function of the applied load. The overall mean 11ac-11ac bond delay for the test was 50263.23 μs. The simulation output increased non-linearly as the applied bond load was raised and the bond delay was equal to both the 11ac Link 1 delay and the 11ac Link 2 delay. As seen below the resulting bond delay increased from a low of 0.00158 μs at the initial applied load of $\{n_1 = 1, n_2 = 1\}$ to 117000 μs at the heaviest load of $\{n_1 = 100, n_2 = 100\}$.

When compared to the absolute delay of each individual 802.11ac link the resulting delay difference was negligible at all applied loads with an overall mean of 115 μs and minimum and maximum values of 0.0009 μs and 725 μs respectively.

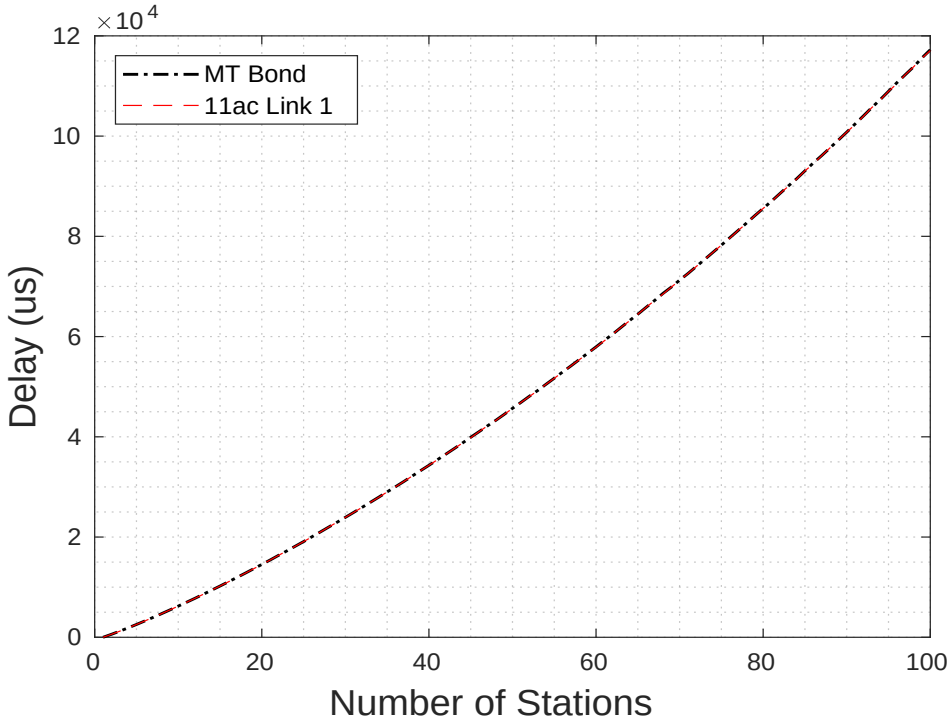

Figure 4.64: Test 1A MT Algorithm Bond Delay Results

4.7.1.2 MT Test 1A: Discussion

The main finding from the benchmark case in Test 1A was that the achieved bond throughput under the MT algorithm was close or equal to the throughput of the static round robin configuration for all applied loads from $\{n_1 = n_2 = 1\}$ to $\{n_1 = n_2 = 100\}$. The overall mean bond throughput of both approaches was 2.47 Mbps with a mean link utilisation of 99.99%. The equivalent performance can be seen from Figure 4.65 below where the MT throughput is plotted as the dotted and dashed black line and the round robin throughput as the solid red line.

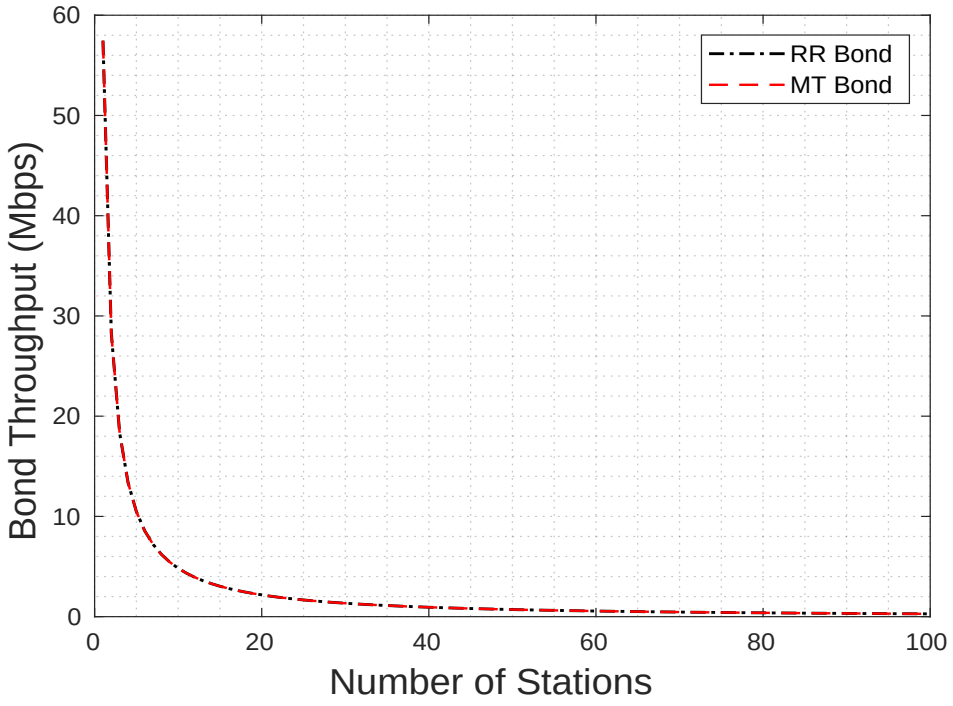

Figure 4.65: Test 1A Dynamic MT vs Static RR Bond Throughput Comparison

As discussed in the round robin evaluation in Chapter 3 the balanced applied load used in Test 1A provides the ideal conditions for the homogeneous 11ac-11ac bond as the throughput of both links is approximately equal for all applied loads. The MT algorithm is designed to remove the slower slave link whenever the predicted bond throughput is less than that of the fastest single available link, but as expected due to the balanced load and heterogeneous 11ac-11ac and 65Mbps-65-Mbps bond the condition was never met throughout Test 1A. This can be seen from Figure 4.66 where the predicted round robin robin throughput is plotted alongside the individual link throughputs. For this reason and as desired the dynamic MT link selection algorithm selected both 11ac slave links for use simultaneously for all applied loads considered giving an equivalent performance to the traditional static bond configuration.

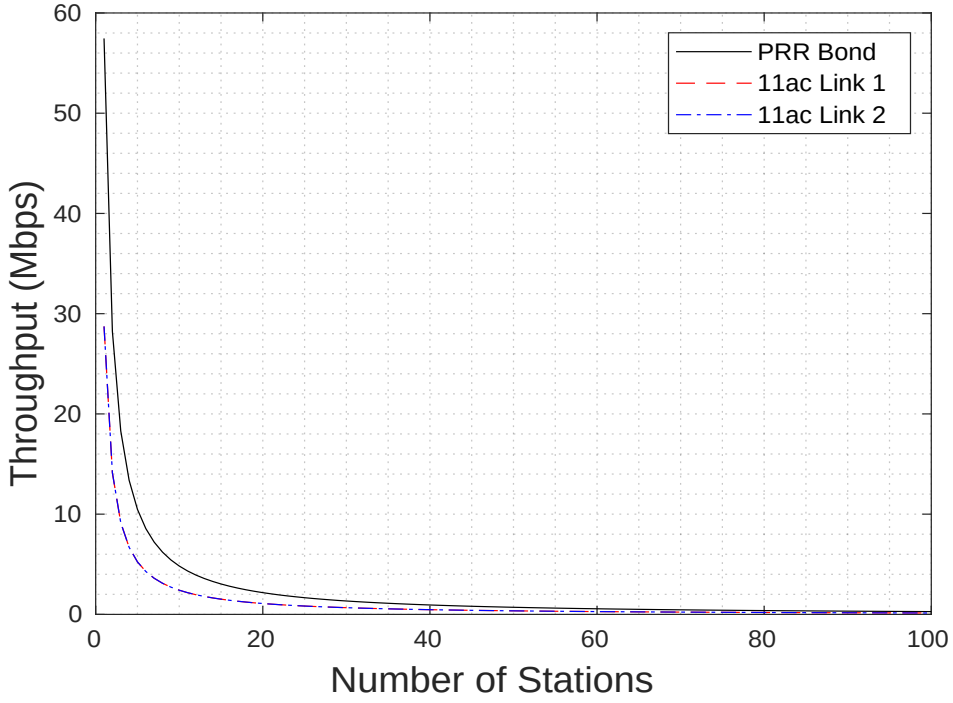

Figure 4.66: Test 1A Predicted Static Round Robin Throughput vs Individual Link Throughput

4.7.1.3 MT Test 1B: Results

The bond throughput results for the MT algorithm and the unbalanced load scenario in Test 1B are presented in Figure 4.67 below. The maximum theoretical bond throughput is given as the solid black line and the MT algorithm throughput as the solid green line. The overall mean achieved 11ac-11ac bond throughput for all applied load combinations used was 29.3371 Mbps. As seen from the graph between applied loads of $\{n_1 = 1, n_2 = 1\}$ and $\{n_1 = 7, n_2 = 1\}$ the throughput output decreased non-linearly from a maximum of 57.47 Mbps to 28.84 Mbps. Then as the applied load was incremented to $\{n_1 = 8, n_2 = 1\}$ the achieved throughput reduced to 28.73 Mbps and remained constant as the load was further increased towards $\{n_1 = 100, n_2 = 1\}$. The overall mean throughput difference between both 11ac bond links for all applied loads used was 1.2338 Mbps.

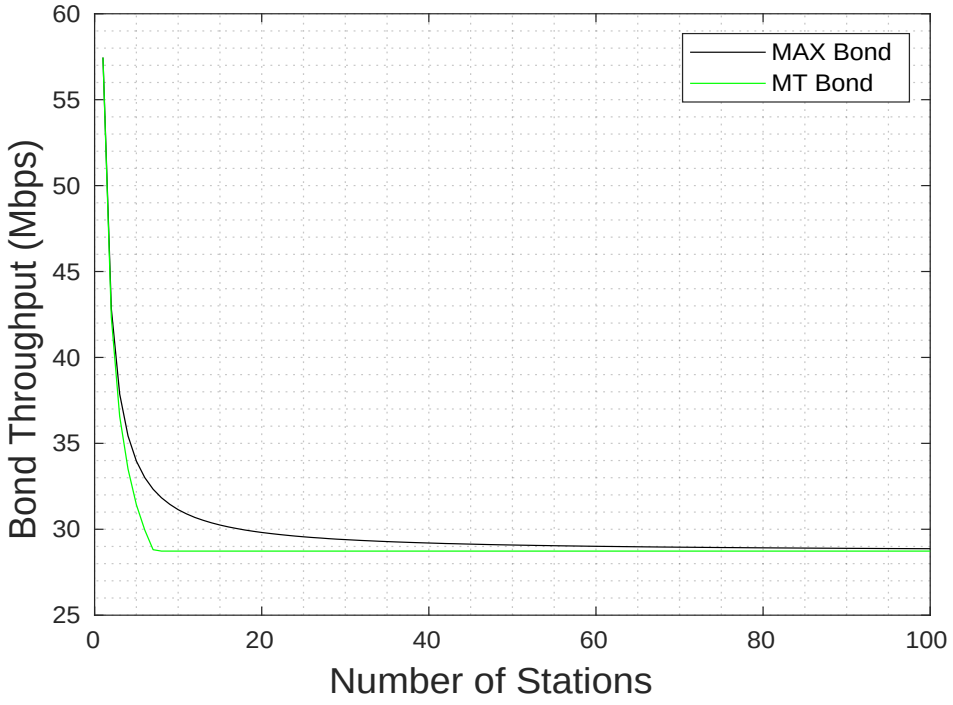

Figure 4.67: Test 1B MT Algorithm Bond Throughput Results

The link bandwidth utilisation results for the homogeneous 11ac-11ac bond and unbalanced load in Test 1B are presented in Figure 4.68. The overall mean bond utilisation for all applied loads used in the test was 99.625%. As seen from the graph the utilisation reduced non-linearly from a maximum of 100% at $\{n_1 = 1, n_2 = 1\}$ to a minimum value of 88.98% at $\{n_1 = 7, n_2 = 1\}$. Then as the load was incremented to $\{n_1 = 8, n_2 = 1\}$ the utilisation increased again to its maximum possible value and remained constant as the load was further raised towards $\{n_1 = 100, n_2 = 1\}$. This occurred because for applied loads between $\{n_1 = 1, n_2 = 1\}$ and $\{n_1 = 7, n_2 = 1\}$, the relatively minor performance disparity between links did not create a slow-down effect that was severe enough such that the overall bond throughput was reduced to below that of the fastest single link. As the load was incremented to $\{n_1 = 8, n_2 = 1\}$ this threshold was exceeded, and the MT algorithm responded by removing the offending link to leave a single link and 100% bandwidth utilisation for the rest of the test.

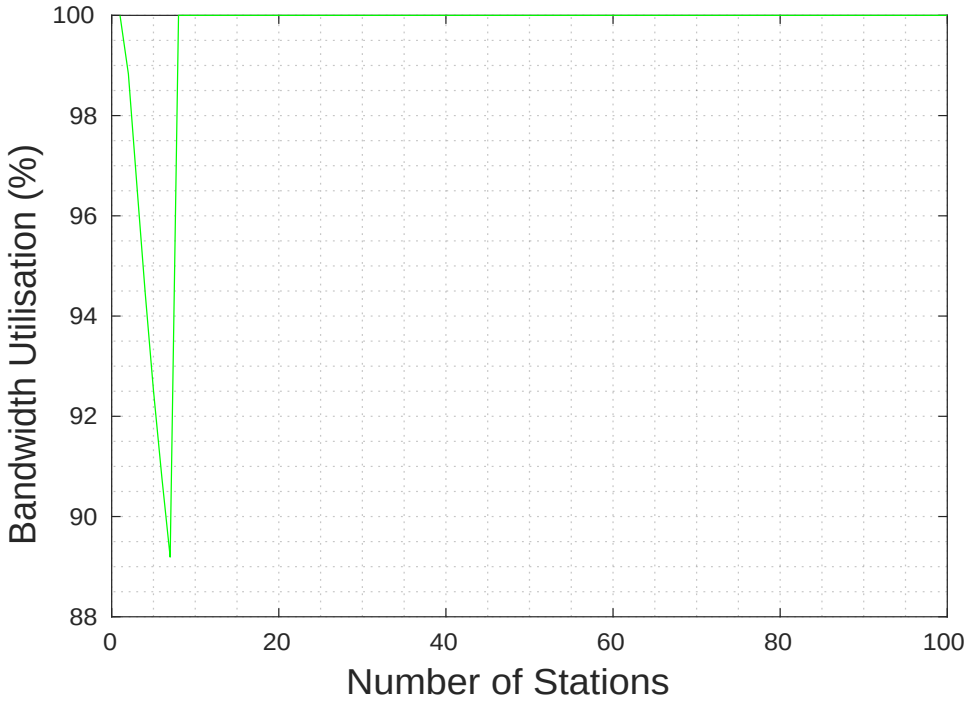

Figure 4.68: Test 1B MT Algorithm Link Bandwidth Utilisation Results

The bond delay results for the MT algorithm in Test 1B are presented in Figure 4.69 below. The overall mean bond delay for all applied load combinations considered was 134.593 μs. As seen from the graph between applied loads of $\{n_1 = 1, n_2 = 1\}$ and $\{n_1 = 7, n_2 = 1\}$ the bond delay followed the Link 1 delay and increased non-linearly from a minimum of 0.0194 μs to a maximum value of 3972 μs. Then as the load was incremented to $\{n_1 = 8, n_2 = 1\}$ the delay output suddenly reduced to 0.0023 μs, as the bond delay began to follow that of the single, statically loaded 802.11ac Link 2 as the load was further increased towards the maximum of $\{n_1 = 100, n_2 = 1\}$. The overall mean link delay difference for all applied load combinations considered in the test was 134.5907 μs.

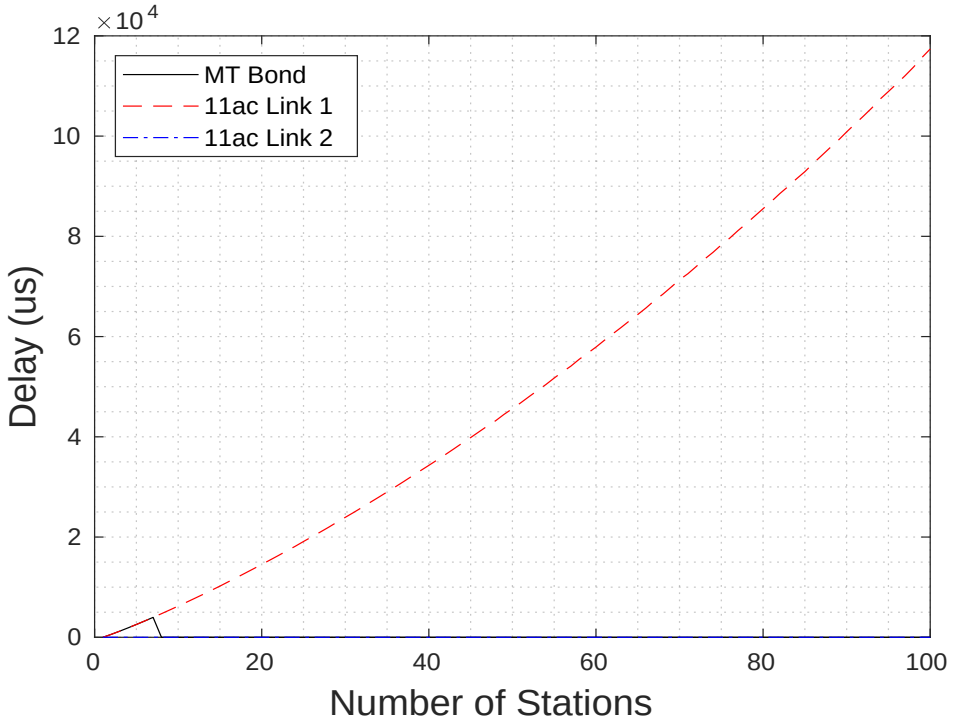

Figure 4.69: Test 1B MT Algorithm Bond Delay Results

4.7.1.4 MT Test 1B: Discussion

The main finding from Test 1B was that the dynamic MT link selection algorithm outperformed the static bond configuration by 48.46% in terms of the mean bond throughput and increased link bandwidth utilisation by 53.2%. The achieved mean bond throughput for the MT algorithm was 29.3372 Mbps with a mean utilisation of 99.625% compared to 19.7604 Mbps and 65.03% respectively for static round robin. The MT bond throughput is plotted in Figure 4.70 below as the dashed black line alongside the round robin bond throughput as the solid red line. As seen for applied loads between $\{n_1 = 1, n_2 = 1\}$ and $\{n_1 = 7, n_2 = 1\}$ the performance of the two methods was equivalent at each data point. However as the load incremented to $\{n_1 = 8, n_2 = 1\}$ and then further towards the heaviest considered load of $\{n_1 = 100, n_2 = 1\}$ the round robin throughput continued its downward trajectory while the MT algorithm throughput remained constant at 28.75 Mbps.

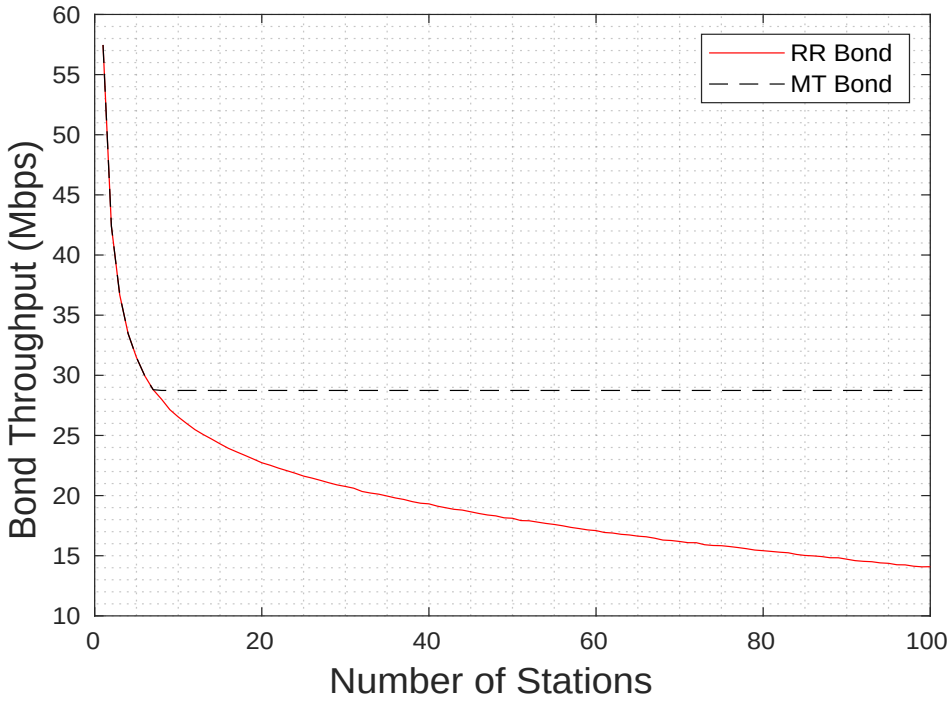

Figure 4.70: Test 1B MT vs RR Bond Throughput Performance Comparison

In Test 1B we saw the reaction of the MT dynamic algorithm when the predicted bond throughput reduced below the specified threshold. As seen in Figure 4.71 below for applied loads from $\{n_1 = 1, n_2 = 1\}$ to $\{n_1 = 7, n_2 = 1\}$ both links were in use and the predicted bond throughput reduced non-linearly from 57.42 Mbps to 30.04 Mbps. During the same interval the 11ac Link 2 throughput remained constant at approximately 28.75 Mbps while the 11ac Link 1 throughput reduced exponentially from 28.75 Mbps to 4.29 Mbps. As the applied load incremented to $\{n_1 = 8, n_2 = 1\}$ the predicted bond throughput reduced below the Link 2 throughput due to a severe slow-down effect and therefore the MT algorithm responded as desired by removing the problematic 11ac Link 1 from the bond. As the applied load was further raised towards $\{n_1 = 100, n_2 = 1\}$ the predicted bond throughput continued to decrease non-linearly and the slower 11ac Link 1 was therefore never re-attached. In taking this approach the dynamic MT link selection algorithm was able to avoid the counter productive use of bonding at $\{n_1 <= 8, n_2 = 1\}$ and increase the achieved overall mean bond throughput and link utilisation substantially when compared to the traditional static bond configuration.

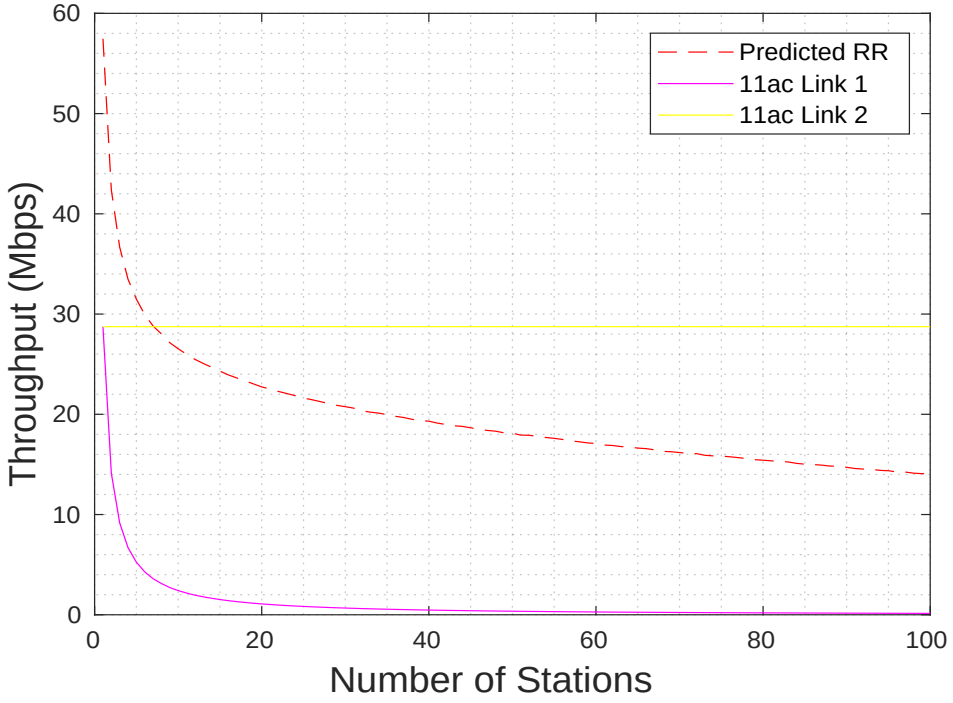

Figure 4.71: Test 1B MT vs Link Throughput Performance Comparison

4.7.1.5 MT Test 1C: Results

The homogeneous 11ac-11ac bond throughput under the MT algorithm the unbalanced load in Test 1C is shown in Figure 4.72. The overall mean achieved bond throughput for all applied load combinations considered in the test was 1.4914 Mbps. As seen from the graph a familiar non-linear decreasing curve was observed as the applied bond load was increased and there was a small difference seen between the achieved MT algorithm throughput and the maximum theoretical bond throughput. At the lightest load of $\{n_1 = 1, n_2 = 50\}$ the MT algorithm throughput was measured at approximately 28.74 Mbps with the maximum bond throughput slightly higher at 29.09 Mbps. As the applied load was increased towards the maximum of $\{n_1 = 100, n_2 = 50\}$ the achieved throughput decreased non-linearly to a minimum of 0.353 Mbps while the maximum theoretical bond throughput decreased to 0.4925 Mbps. The overall mean link throughput difference for all applied loads considered in the test was 0.19539 Mbps.

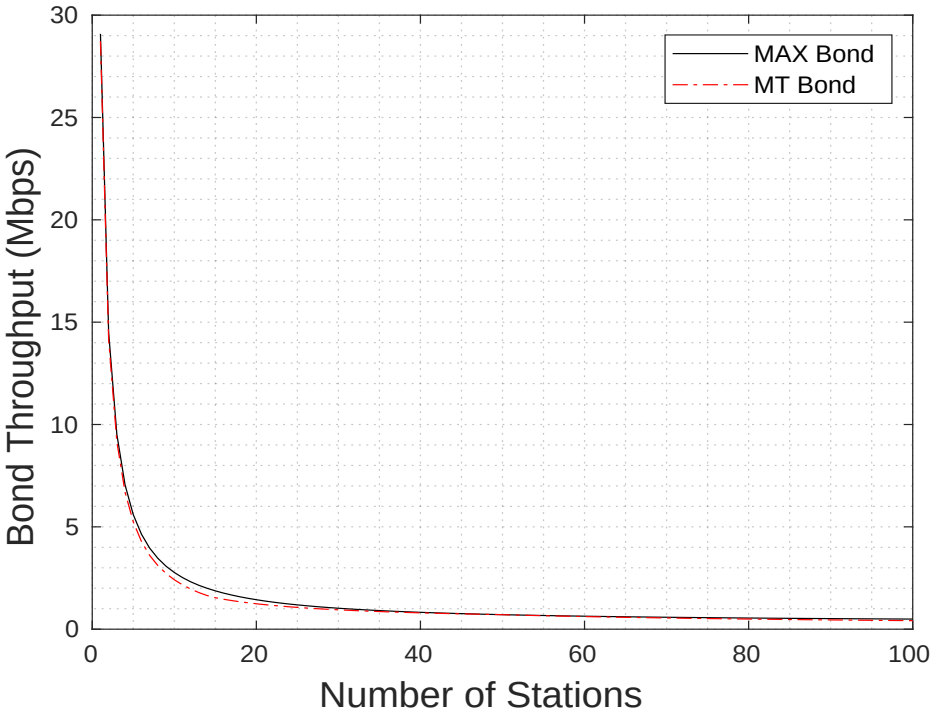

Figure 4.72: Test 1C MT Algorithm Bond Throughput Results

The MT algorithm link bandwidth utilisation results for Test 1C are shown in Figure 4.73 below. The overall mean bond utilisation for all applied load combinations considered was 93.479%. As seen from the graph the utilisation was equal to approximately 100% from $\{n_1 = 1, n_2 = 50\}$ to $\{n_1 = 14, n_2 = 50\}$. However as the load was incremented to $\{n_1 = 15, n_2 = 50\}$ the output decreased suddenly to 82.15% before then increasing gradually to 99.9% at exactly $\{n_1 = 50, n_2 = 50\}$. As the load was raised further the utilisation then decreased non-linearly to a final value of 84.13% at the heaviest prescribed load of $\{n_1 = 100, n_2 = 50\}$.

For applied loads between $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 16, n_2 = 50\}$ the MT algorithm throughput threshold was exceeded and therefore only a single link was active in the bond. Then as the load incremented to $\{n_1 = 17, n_2 = 50\}$, the bond throughput increased above that of the single fastest link, and therefore the inactive link was re-added to the bond causing a sudden decrease in the bandwidth utilisation. Then as the load was further increased the performance disparity between links gradually reduced until they become comparable at $\{n_1 = 50, n_2 = 50\}$. As the load was increased further still the performance disparity grew which gradually reduced the bond bandwidth utilisation.

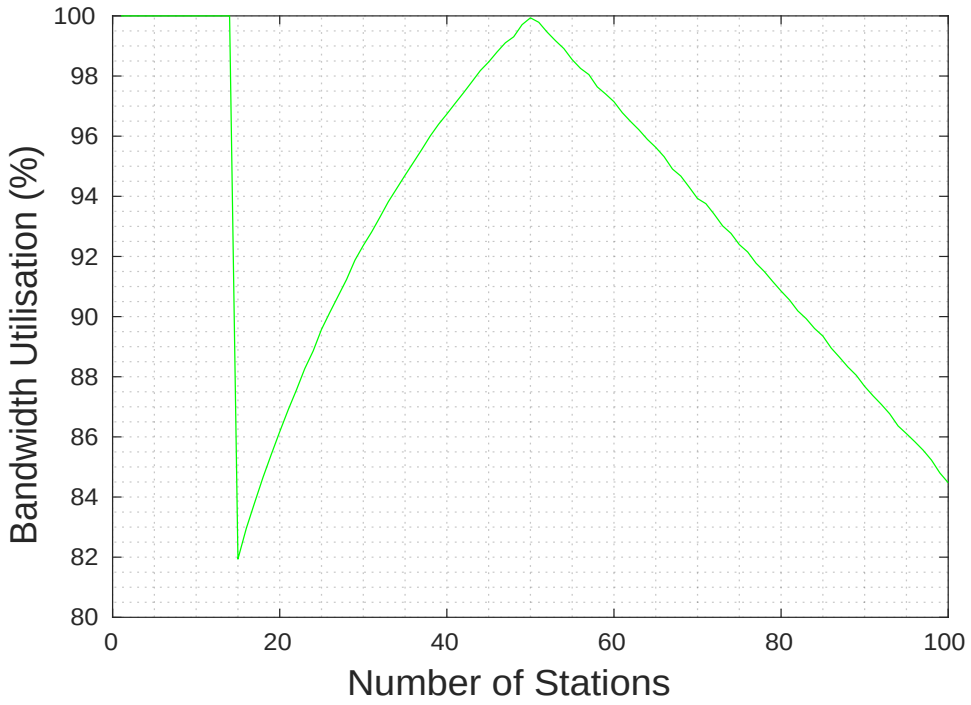

Figure 4.73: Test 1C MT Algorithm Link Bandwidth Utilisation Results

The bond delay results for the MT algorithm in Test 1C are shown in Figure 4.74 below. The overall mean dual-11ac bond delay for all applied load combinations considered was 56864.067 μs. For applied loads from $\{n_1 = 1, n_2 = 50\}$ to $\{n_1 = 16, n_2 = 50\}$ the bond delay was equal to the Link 1 delay which increased from a minimum of 0.0284 μs to a maximum of 44360 μs. However as the load was incremented to $\{n_1 = 17, n_2 = 50\}$ the bond delay started to follow the Link 2 delay and increased to approximately 45490 μs and remained constant as the load was further increased towards $\{n_1 = 50, n_2 = 50\}$. After this the bond delay began to follow the Link 1 delay again and increased non-linearly to 116400 at the heaviest considered load of $\{n_1 = 100, n_2 = 50\}$.

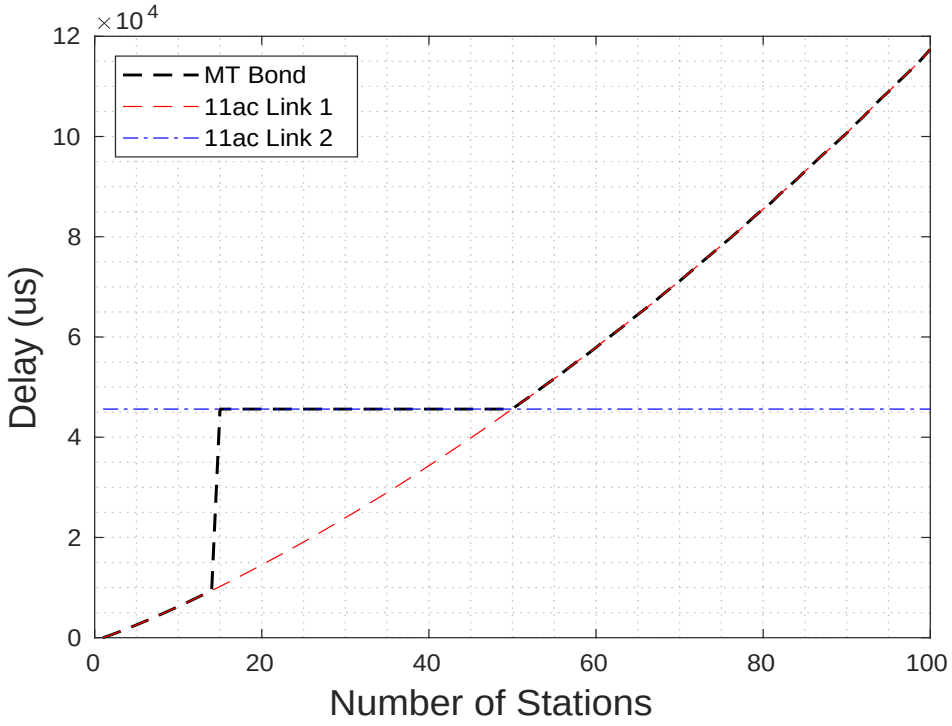

Figure 4.74: Test 1C MT Algorithm Bond Delay Results

4.7.1.6 MT Test 1C: Discussion

The main finding from Test 1C was that the dynamic MT algorithm outperformed the static bond configuration in terms of mean bond throughput by 20.06% and mean link utilisation by 5.04%. The achieved mean MT bond throughput was 1.4914 Mbps with a mean bandwidth utilisation of 94.188% compared to 1.2422 Mbps and 89.667% respectively for static round robin. The MT throughput is plotted in Figure 4.75 below as the dashed black line alongside the achieved round robin throughput as the dotted and dashed red line. As seen between $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 14, n_2 = 50\}$ the MT algorithm throughput decreased non-linearly from 29.09 Mbps to 1.644 Mbps while the round robin throughput decreased from 18.06 Mbps to 1.619 Mbps. Then as the load was incremented to $\{n_1 = 15, n_2 = 50\}$ the MT throughput reduced to 16.44 Mbps which was just below the 16.19 Mbps achieved by round robin. The MT bond throughput proceeded to decrease at a faster rate until $\{n_1 = 50, n_2 = 50\}$ where it was reduced to 0.3542 Mbps and then remained constant as the load was further increased towards $\{n_1 = 100, n_2 = 50\}$. For the same interval the round robin throughput continued its non-linear downward trend to a minimum value of 0.4159 Mbps.

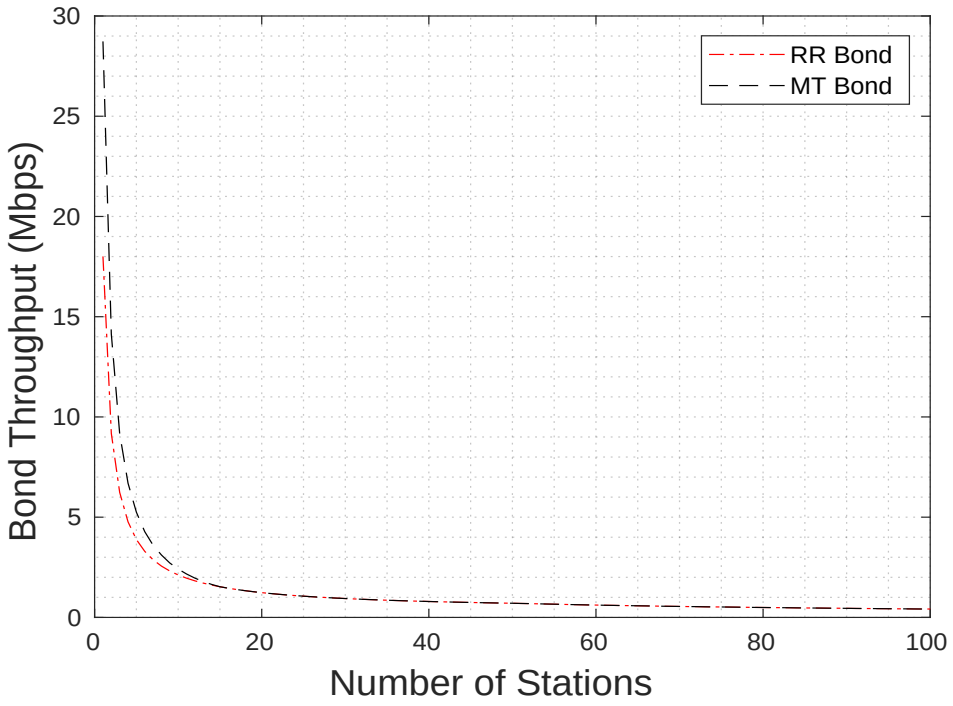

Figure 4.75: Test 1C MT vs RR Bond Throughput Performance Comparison

In Test 1C we saw the dynamic MT algorithm re-attach the slower 11ac link when the predicted bond throughput increased above the specified threshold. As seen from Figure 4.76 at the initial load of $\{n_1 = 1, n_2 = 50\}$ the predicted bond throughput was less than the throughput of the individual 11ac Link 1. However as the load was increased to $\{n_1 = 15, n_2 = 50\}$ the above condition no longer held and the algorithm responded by re-attaching the previously problematic slower 11ac Link 1 to the interface bond. As the load was further increased to $\{n_1 = 51, n_2 = 50\}$ 11ac Link 1 then became the slower of the two due to the increased channel contention such that the overall bond throughput was instead limited by 11ac Link 2 with the static applied load of n = 50. Because the predicted bond throughput did not decrease below the specified threshold again both links were used for the remainder of the test. By resorting to a single link the MT algorithm avoided the counter-productive use of bonding for applied loads where $\{n_1 <= 14, n_2 = 50\}$ and increased the overall mean throughput and mean link utilisation.

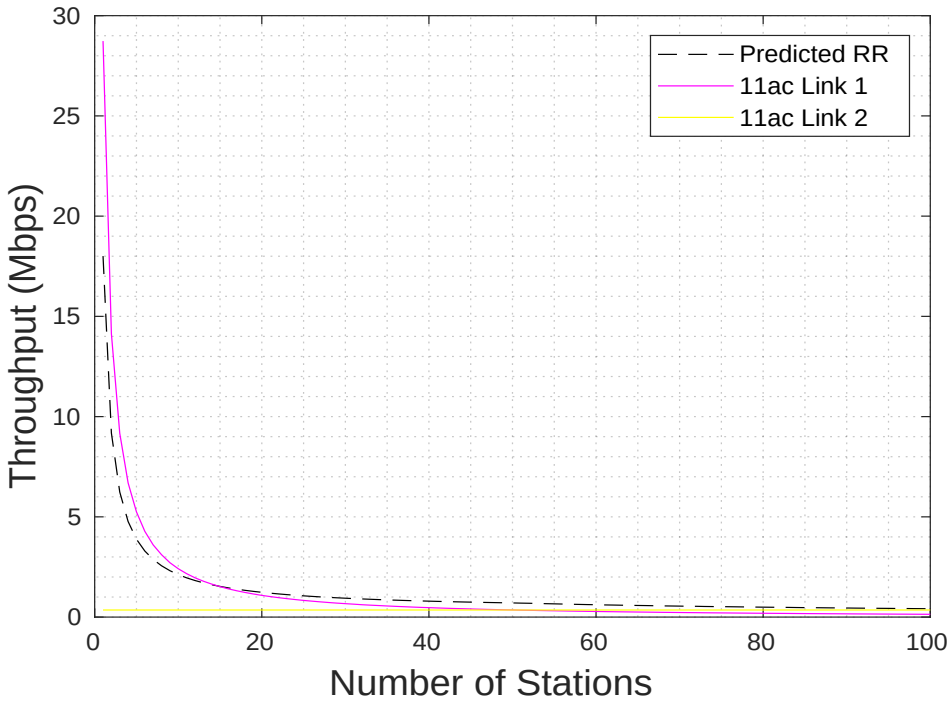

Figure 4.76: Test 1C MT vs Link Throughput Performance Comparison

4.7.1.7 MT Test 1D: Results

The bond throughput results under the MT algorithm for the unbalanced load in Test 1D are presented in Figure 4.77 below. The achieved MT throughput is plotted as the solid green line and the maximum theoretical bond throughput as the solid black line. The overall mean bond throughput for all applied load combinations considered was 1.3004 Mbps. A non-linear decreasing output was observed as the applied bond load was raised with a small difference between the achieved bond throughput and the theoretical maximum bond throughput. At the initial load of $\{n_1 = 1, n_2 = 100\}$ the achieved MT bond throughput was 28.74 Mbps and the maximum theoretical throughput slightly higher at 28.88 Mbps. As the applied load was raised both throughput metrics decreased exponentially as the difference between them became smaller. At the heaviest test load of $\{n_1 = 100, n_2 = 100\}$ both the achieved bond throughput and the theoretical maximum bond throughput were approximately equal at 0.278 Mbps. The overall mean link throughput difference for all applied loads considered was 0.076282 Mbps.

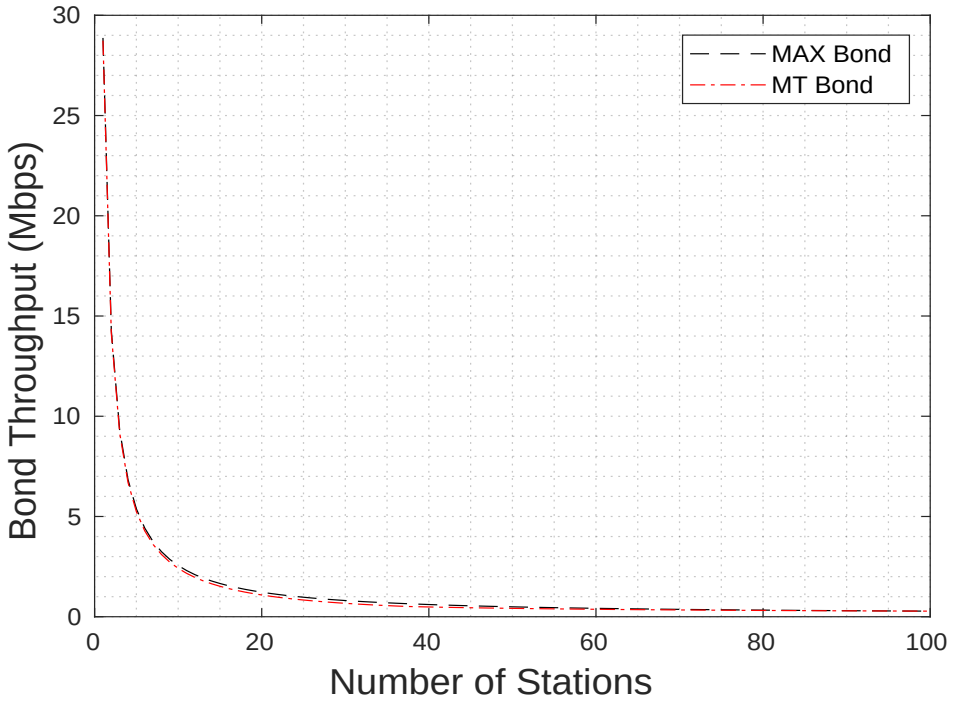

Figure 4.77: Test 1D MT Algorithm Bond Throughput Results

The MT algorithm link bandwidth utilisation results for Test 1D are shown in Figure 4.78 below. As seen from the graph the resulting utilisation for the homogeneous 11ac-11ac bond used was equal to approximately 100% for applied loads from $\{n_1 = 1, n_2 = 100\}$ to $\{n_1 = 38, n_2 = 100\}$. Then as the load was incremented to $\{n_1 = 39, n_2 = 100\}$ the output decreased suddenly to an overall minimum of 78.97% before then increasing non-linearly to a final value of 99.91% at the heaviest considered bond load of $\{n_1 = 100, n_2 = 100\}$. The overall mean bond utilisation for the test was 94.188%.

Between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 38, n_2 = 100\}$ only a single link was active in the bond because the MT algorithm threshold was exceeded, and this led to a bandwidth utilisation of 100%. Then as the load was incremented to $\{n_1 = 39, n_2 = 100\}$ the bond throughput exceeded that of the fastest single link and the MT algorithm responded by re-attaching the slower bond link, leading to a sudden reduction in the link bandwidth utilisation. As the load was further increased, the performance disparity between links reduced leading to a gradual increase in the bandwidth utilisation.

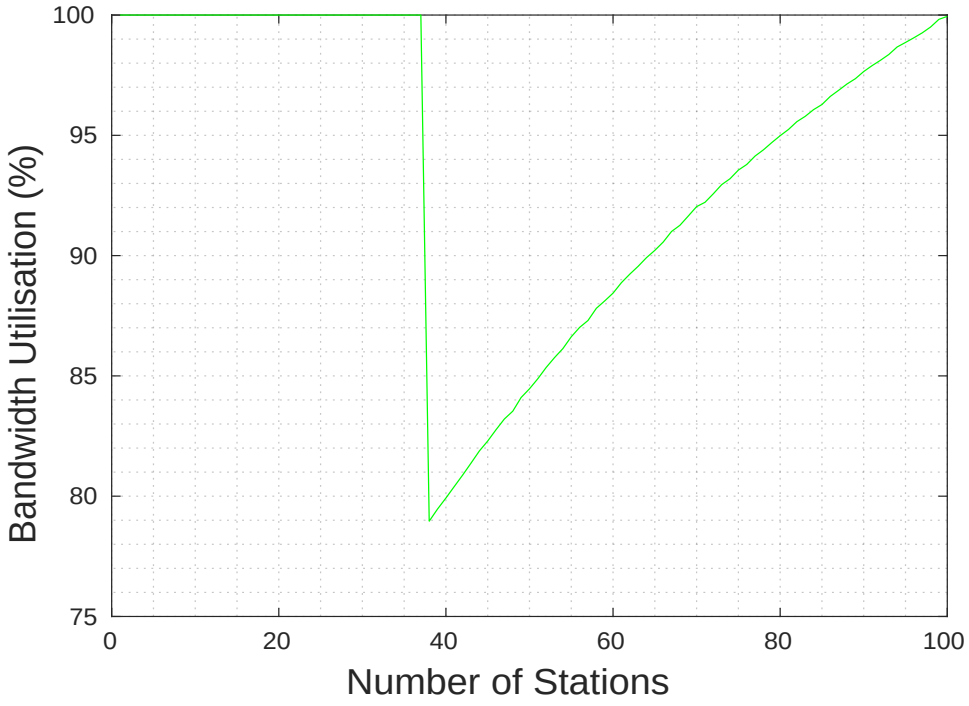

Figure 4.78: Test 1D MT Algorithm Link Bandwidth Utilisation Results

The dual-11ac bond delay results for the MT algorithm in Test 1D are presented in Figure 4.79 below. The overall mean bond delay for all applied load combinations used was 779074.58 μs. As seen for applied loads from $\{n_1 = 1, n_2 = 100\}$ to $\{n_1 = 38, n_2 = 100\}$ the bond delay followed the Link 1 delay which increased non-linearly from a low of approximately 0.00284 μs to 31050 μs. Then as the load was incremented to $\{n_1 = 38, n_2 = 100\}$ the bond delay began to follow the Link 2 delay which remained constant at approximately 116900 μs as the load was further increased towards $\{n_1 = 100, n_2 = 100\}$. The overall mean link throughput difference for all applied load combinations considered was 28857.6 μs.

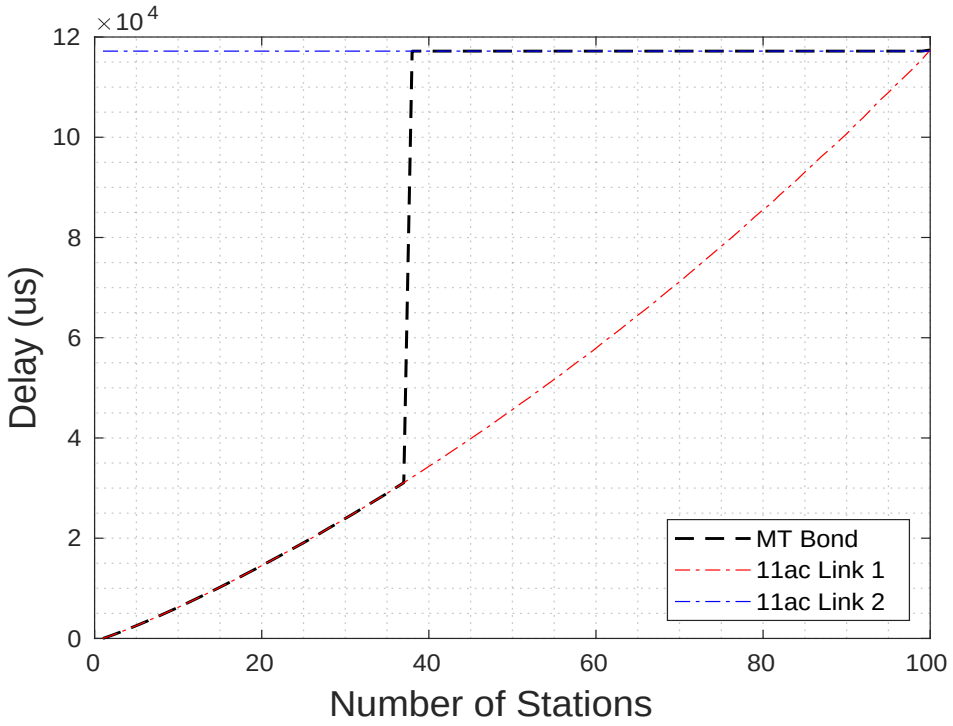

Figure 4.79: Test 1D MT Algorithm Bond Delay Results

4.7.1.8 MT Test 1D: Discussion

The main finding from Test 1D was that the dynamic MT algorithm outperformed the static configuration by 51.51% in terms of the mean bond throughput and by 15.36% in terms of link bandwidth utilisation. The achieved mean bond throughput for the MT algorithm was 1.3 Mbps with a mean utilisation of 94.1488% while the achieved means for static round robin were 0.85805 Mbps and 81.615% respectively. The performance of both methods is compared in Figure 4.80 below where the MT algorithm throughput is plotted as the dashed green line and the round robin throughput as the solid red line. As seen for applied loads between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 38, n_2 = 100\}$ the MT algorithm throughput decreased non-linearly from 28.73 Mbps to 0.536 Mbps and the round robin throughput decreased from 14.1 Mbps to 0.527 Mbps while the difference between the two outputs became smaller. As the load incremented to $\{n_1 = 39, n_2 = 100\}$ the MT throughput reduced to that of round robin and it continued to follow the same non-linear curve reducing to a minimum of 0.277 Mbps at the heaviest prescribed load of $\{n_1 = 100, n_2 = 100\}$.

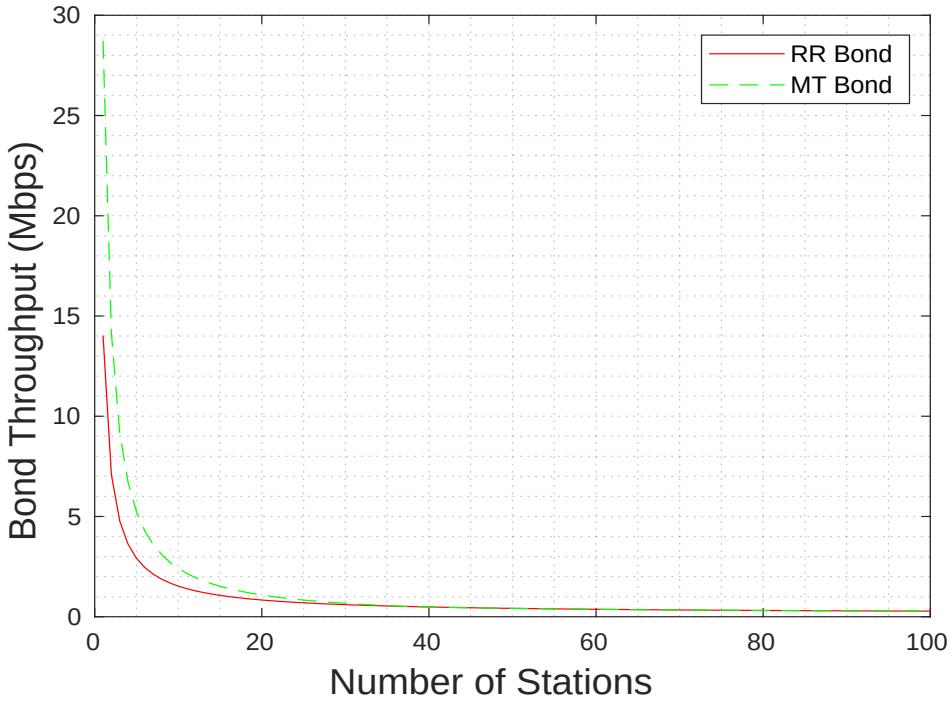

Figure 4.80: Test 1D MT vs RR Bond Throughput Performance Comparison

In Test 1D we saw the dynamic MT link selection algorithm re-attach the slower 11ac device when the predicted bond throughput increased above the throughput of the fastest available single link. As seen in Figure 4.81 for applied loads between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 38, n_2 = 100\}$ the above condition was not met and therefore only the faster 11ac link was in use. Then as the load was incremented to $\{n_1 = 39, n_2 = 100\}$ the predicted throughput reduced below that of the single fastest link (11ac Link 1) and the algorithm responded by re-attaching the slower link (11ac Link 2) to the bond. In doing to the MT algorithm avoided the counter productive use of bonding where $\{n_1 <= 38, n_2 = 100\}$ and increased the overall mean throughput and mean link bandwidth utilisation compared to the static round robin case.

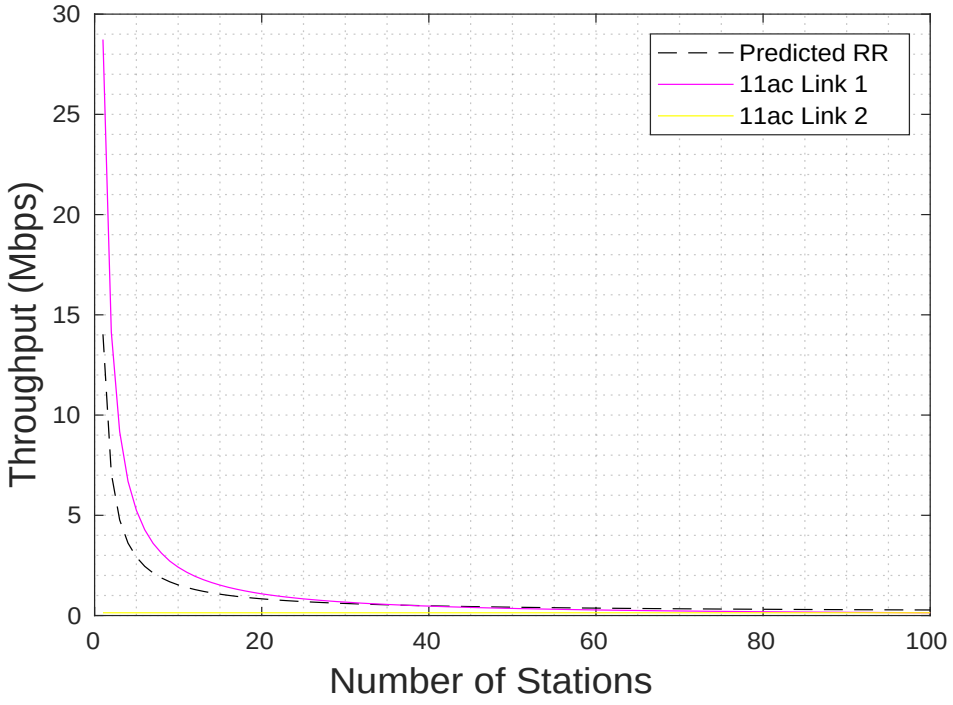

Figure 4.81: Test 1D MT vs Link Throughput Performance Comparison

4.7.2 Heterogeneous 11ac-11ah Bond

Next we present the results of the MT algorithm for the heterogeneous 11ac-11ah bond.

4.7.2.1 MT Test 2A: Results

The bond throughput results for the MT algorithm under the balanced load scenario in Test 2A are presented in Figure 4.82 below. The overall mean bond throughput for the MT algorithm was 1.324 Mbps while the mean theoretical maximum bond throughput was 1.8897 Mbps. A non-linear decreasing curve was observed for both metrics as the applied 11ac-11ah bond load was raised. From a maximum of 28.72 Mbps at the lightest test load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ the MT algorithm throughput decreased non-linearly to 0.1659 Mbps at the heaviest applied load of $\{n_{11ac} = 100, n_{11ah} = 100\}$. For the same applied loads the maximum theoretical bond throughput decreased non-linearly from a maximum of 41.88 Mbps to a minimum of 0.2219 Mbps. The mean throughput difference between the 11ac and 11ah bond links for all applied load combinations considered was 0.59008 Mbps and insignificant compared to the individual link throughputs.

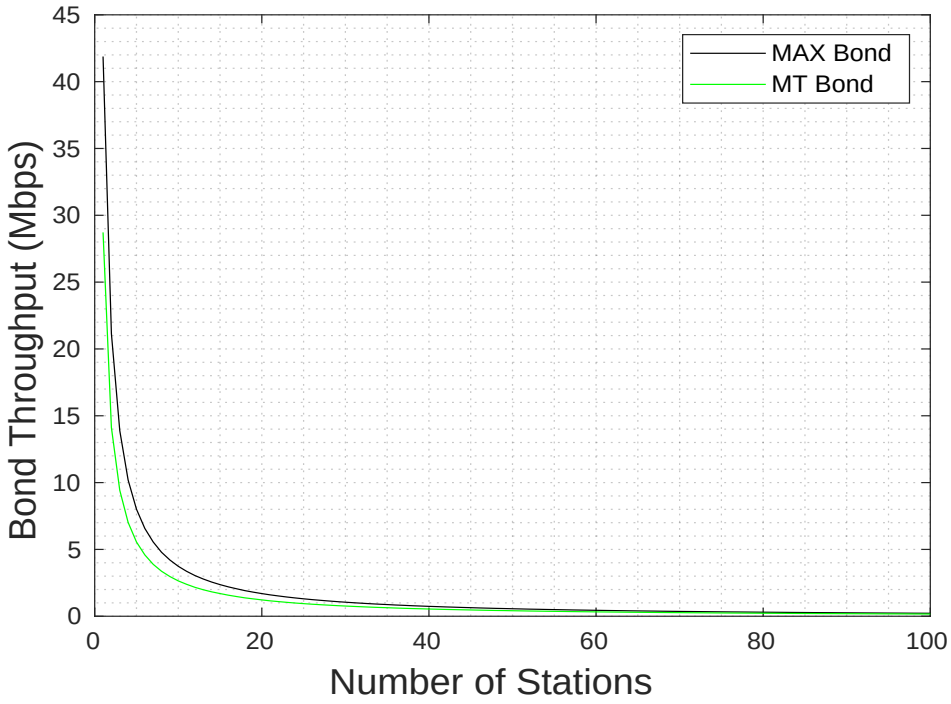

Figure 4.82: Test 2A MT Algorithm Bond Throughput Results

The link bandwidth utilisation results for the MT algorithm in Test 2A are shown in Figure 4.83 below. The overall mean 11ac-11ah bond utilisation for all applied load combinations considered was 73.705%. As seen for applied loads of $\{n_{11ac} = 1, n_{11ah} = 1\}$ and $\{n_{11ac} = 1, n_{11ah} = 1\}$ the utilisation was 100%, but as the load was incremented to $\{n_{11ac} = 3, n_{11ah} = 3\}$ the simulation output decreased suddenly to a minimum of 67.86% before then increasing gradually to 74.76% at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$.

Between loads of $\{n_{11ac} = 1, n_{11ah} = 1\}$ and $\{n_{11ac} = 2, n_{11ah} = 2\}$ only a single link was attached to the bond, and therefore the resulting bandwidth utilisation was 100%. Then as the applied load was incremented to $\{n_{11ac} = 3, n_{11ah} = 3\}$, the bond throughput began to exceed that of the single fastest available link and so the inactive link was re-attached to the bond by the algorithm, also causing a sudden decrease in the bandwidth utilisation. Then as the load was further increased the performance disparity between bond links decreased which gradually increased the achieved bandwidth utilisation.

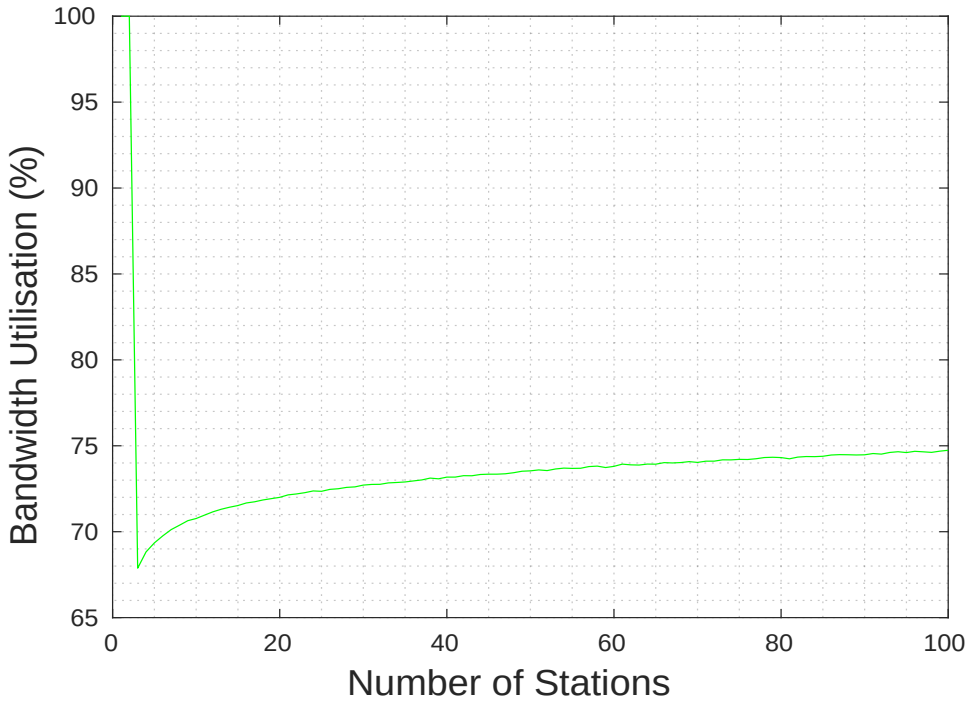

Figure 4.83: Test 2A MT Algorithm Link Bandwidth Utilisation Results

The bond delay results for the MT algorithm under the balanced load scenario in Test 2A are presented in Figure 4.84 below as a function of the applied bond load. The mean bond delay for all applied load combinations considered in the test was 85297.98 μs. As observed below the bond delay followed the slower 11ah link delay throughout which increased non-linearly from a minimum of 0.01568 μs at $\{n_{11ac} = 1, n_{11ah} = 1\}$ to a maximum of 195500 μs at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$. The overall mean link delay difference for all applied load combinations considered was 35089.9 μs.

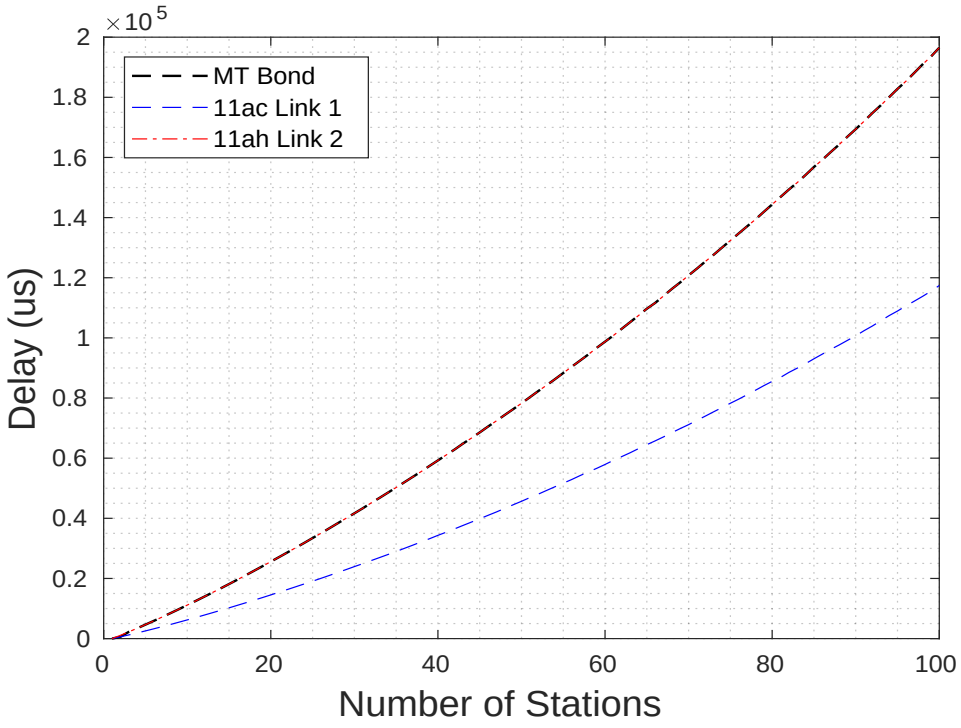

Figure 4.84: Test 2A MT Algorithm Bond Delay Results

4.7.2.2 MT Test 2A: Discussion

The main finding from the balanced load scenario in Test 2A was that the dynamic predicted throughput algorithm marginally increased the mean bond throughput by 1.92% and the mean link utilisation by 1% compared to the static bond configuration. The mean throughput under the MT algorithm was 1.324 Mbps with a link utilisation of 73.7% while the achieved means for the static method were 1.299 Mbps and 72.9% respectively. The marginal improvement can be seen below at an applied load of $\{n_{11ac} = n_{11ah} = 1\}$ where the MT algorithm exceeded the round robin throughput by approximately 17.5%. Immediately as the load was incremented to $\{n_{11ac} = n_{11ah} = 2\}$ the achieved MT throughput decreased to the same value as round robin and then continued to follow the same exponentially decreasing output curve as the applied load was further raised towards $\{n_{11ac} = n_{11ah} = 100\}$.

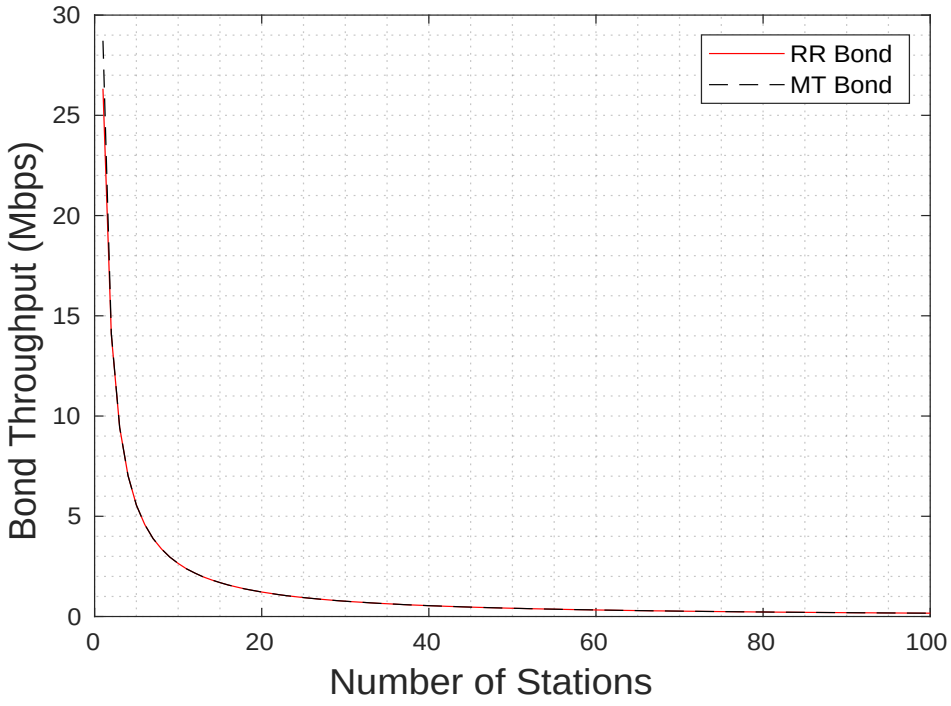

Figure 4.85: Test 2A MT vs RR Algorithm Bond Throughput

The difference in performance between the dynamic link selection method and the traditional static round robin approach is seen in Figure 4.86 at loads of $\{n_{11ac} = n_{11ah} = 1\}$ where the MT algorithm condition for removing a problematic slave link is satisfied, i.e. the predicted bond throughput plotted as the dashed black line was less than or equal to the throughput of the fastest available link, which was in this case the 11ac link plotted as the solid magenta line. However as the load was incremented to $\{n_{11ac} = n_{11ah} = 2\}$ the 11ac link throughput decreased below that of the predicted bond throughput and the MT algorithm responded by reattaching the slower 11ac link. By resorting to the single fastest slave device the dynamic MT algorithm link selection algorithm avoided the damaging scenario at $\{n_{11ac} = n_{11ah} = 1\}$ and marginally increased the the overall mean bond throughput and mean link bandwidth utilisation compared to the static round robin configuration.

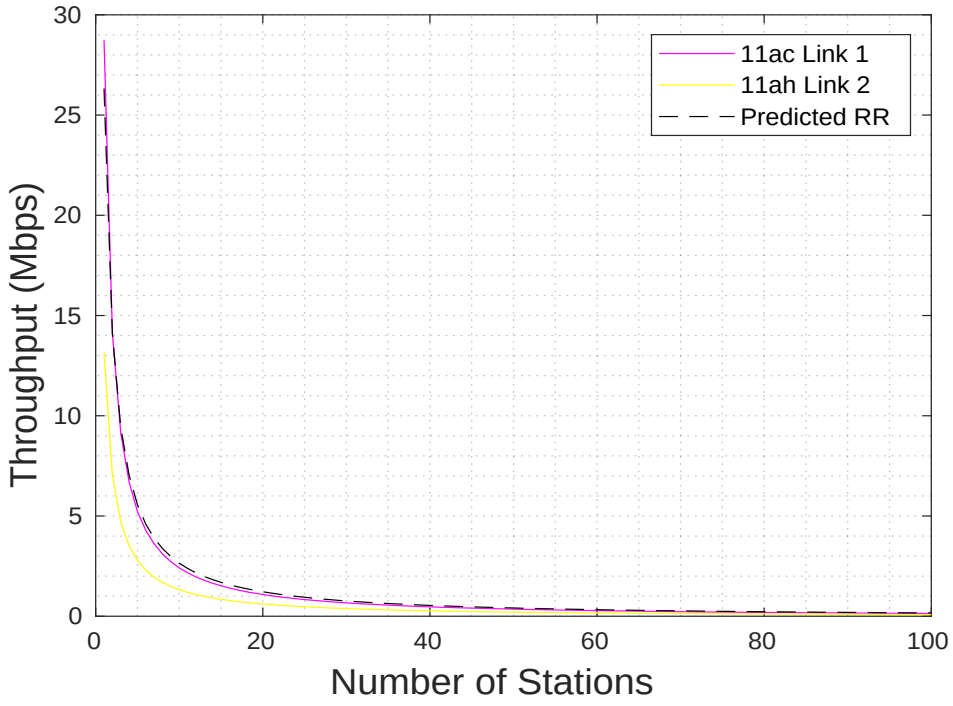

Figure 4.86: Test 2A MT vs Link Throughput Performance Comparison

4.7.2.3 MT Test 2B: Results

The 11ac-11ah bond throughput for the MT algorithm and the unbalanced load scenario in Test 2B is plotted in Figure 4.87 as the solid green line alongside the maximum theoretical bond throughput as the solid black line. The overall mean achieved bond throughput for all applied load combinations considered was 13.8775 Mbps. In general the MT throughput was lower than the theoretical maximum and both outputs decreased exponentially as the applied load was raised. At the lightest load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ the achieved MT bond throughput was approximately 28.74 Mbps and the maximum theoretical bond throughput was 41.9 Mbps. The MT bond throughput then reduced gradually to a minimum of 13.29 Mbps at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 1\}$ while the maximum theoretical bond throughput reduced to 13.3 Mbps. The overall mean link throughput difference for all applied loads considered in the test was 12.1047 Mbps.

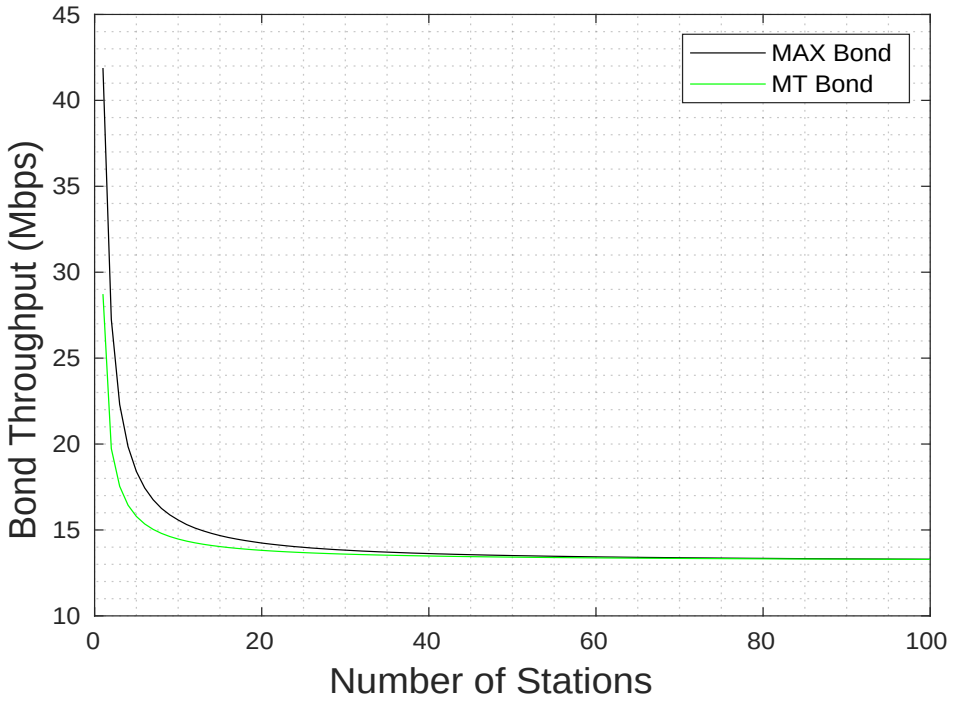

Figure 4.87: Test 2B MT Algorithm Bond Throughput Results

The 11ac-11ah link bandwidth utilisation results for the MT algorithm in Test 2B are given in Figure 4.88. The overall mean bandwidth utilisation for all applied load combinations considered was 97.734%. As seen below the achieved utilisation was 100% at an applied load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ but as the load was incremented $\{n_{11ac} = 2, n_{11ah} = 1\}$ the output reduced suddenly to an overall test low of 72.37%. As the applied load was was further raised the utilisation of the bond increased non-linearly to a final value of 99.94% at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 1\}$.

At the start of the test only a single link was attached to the bond, but as the applied load incremented to $\{n_{11ac} = 2, n_{11ah} = 1\}$ the performance disparity between links reduced and the bond throughput increased above that of the fastest single link, and so the MT algorithm reacted by re-attaching the second link. This also had the impact of suddenly decreasing the bandwidth utilisation. As the load was further increased, the performance disparity was reduced which had the effect of gradually increasing the utilisation.

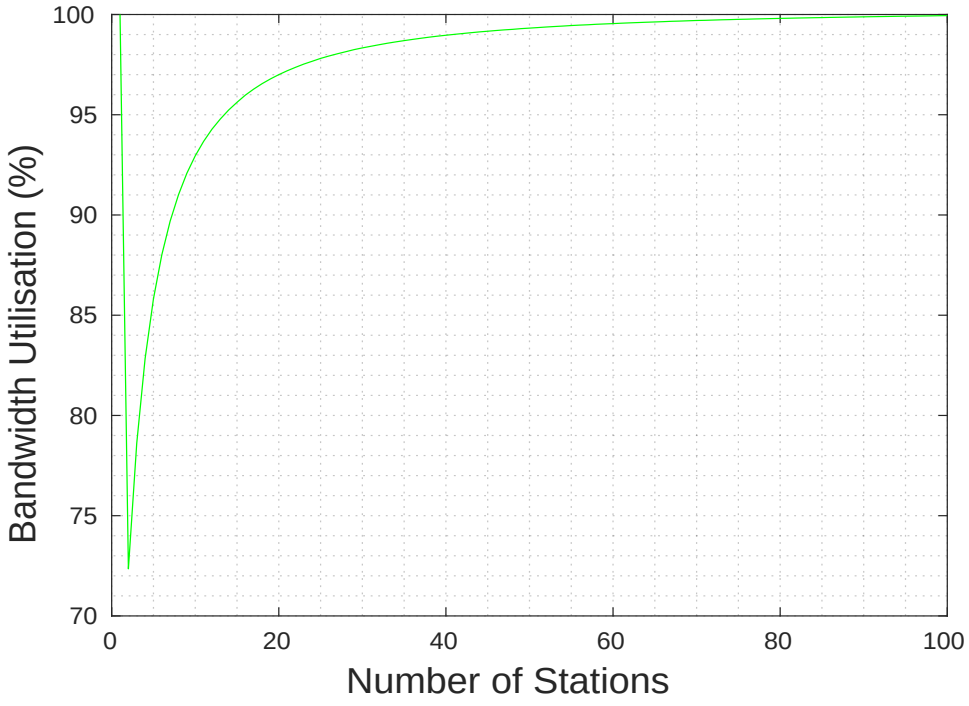

Figure 4.88: Test 2B MT Algorithm Link Bandwidth Utilisation Results

The 11ac-11ah bond delay results for the MT algorithm in Test 2B are presented in Figure 4.89 below. The overall mean bond delay for all applied load combinations considered was 50197.7533 μs. As seen the bond delay was equal to the 11ah delay at $\{n_{11ac} = 1, n_{11ah} = 1\}$ but as the load incremented to $\{n_{11ac} = 2, n_{11ah} = 1\}$ the bond delay began to follow the 11ac link which increased non-linearly from 0.284 μs to a maximum of 116700 μs at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 1\}$. The overall mean link delay difference for all applied load combinations considered was 50197.746 μs

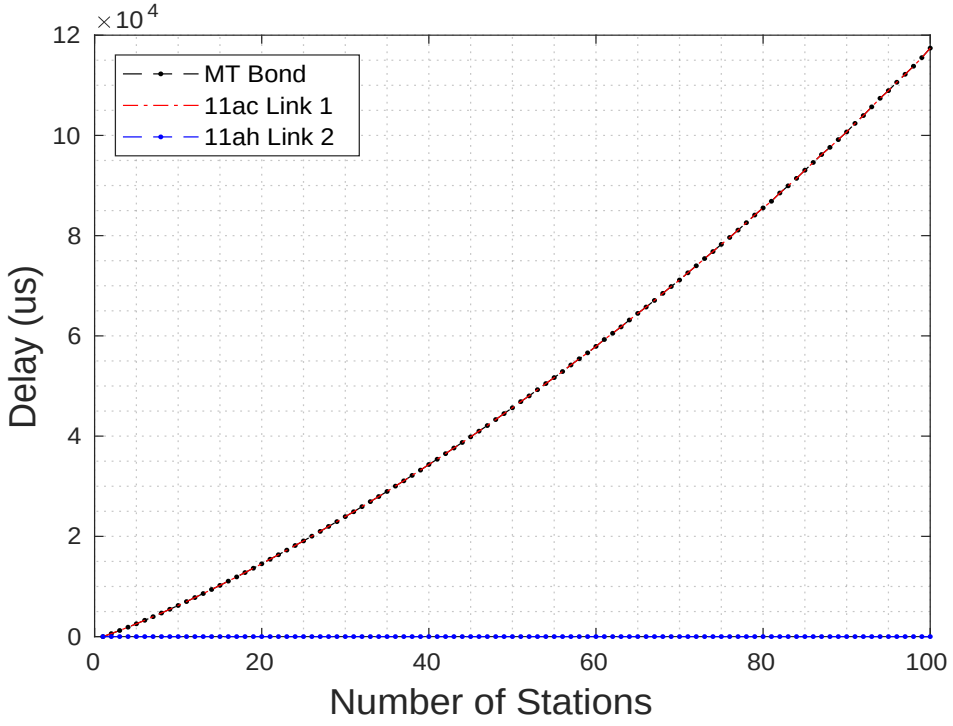

Figure 4.89: Test 2B MT Algorithm Bond Delay Results

4.7.2.4 MT Test 2B: Discussion

The main finding from the first unbalanced scenario in Test 2B was that the dynamic MT algorithm marginally outperformed the static configuration by 0.17 % in terms of the mean bond throughput and by 0.38% in terms of mean link bandwidth utilisation. The mean MT bond throughput was 13.8775 Mbps with a mean throughput efficiency of 97.734% while the achieved means for round robin were 13.8537 Mbps and 97.363% respectively. As seen in Figure 4.95 at $\{n_{11ac} = 1, n_{11ah} = 1\}$ the MT algorithm throughput of 26.33 Mbps was slightly higher than the round robin throughput of 26.28 Mbps, but as the load incremented to $\{n_{11ac} = 2, n_{11ah} = 1\}$ the MT throughput was decreased to that of round robin. Then as the applied load was further raised towards $\{n_{11ac} = 100, n_{11ah} = 1\}$ the MT output continued to follow that of the static round robin configuration such that their performance was equivalent for $\{n_{11ac} >= 2, n_{11ah} = 1\}$ and reducing non-linearly to a minimum of 13.31 Mbps.

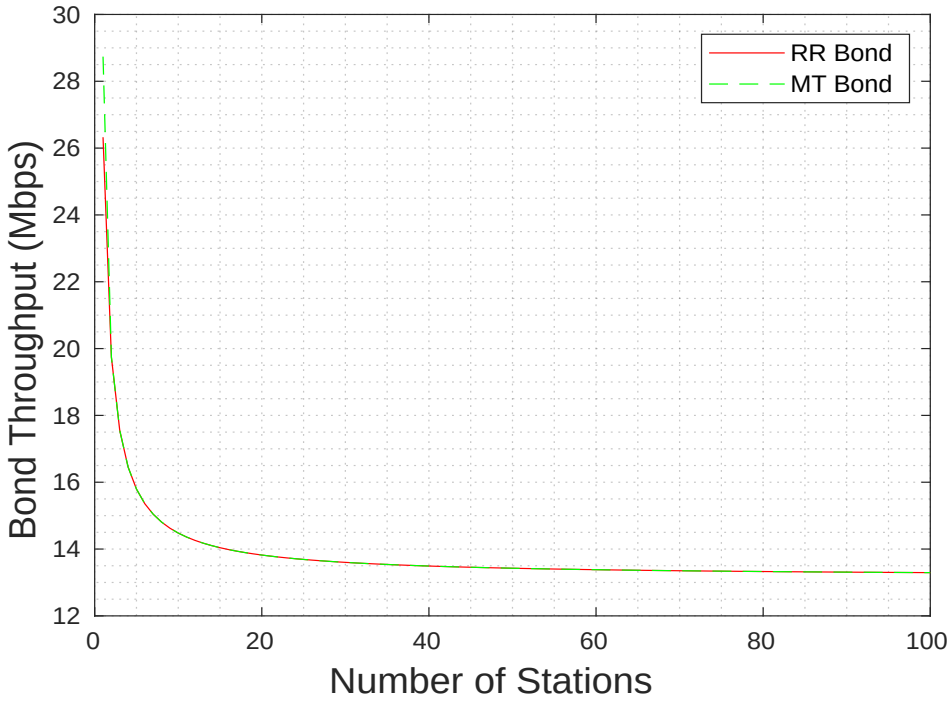

Figure 4.90: Test 2B MT vs RR Algorithm Bond Throughput Performance Comparison

L As seen in Figure 4.91 at $\{n_{11ac} = 1, n_{11ah} = 1\}$ only the 11ac slave link was in use as the predicted bond throughput was less than the throughput of the single fastest available link. Immediately as the load incremented to $\{n_{11ac} = 2, n_{11ah} = 1\}$ the predicted throughput reduced to 13.16 Mbps and due to the above condition no longer being true the slower 11ah link was re-attached to the bond. As seen the predicted throughput did not drop again below the throughput of the fastest link and therefore both links were used where $\{n_{11ac} >= 2, n_{11ah} = 1\}$. Here by avoiding the counter productive use of bonding at $\{n_{11ac} = 1, n_{11ah} = 1\}$ and reverting to the single 11ac link the dynamic MT algorithm was able to marginally increase the mean bond throughput and link utilisation compared to the static link selection approach.

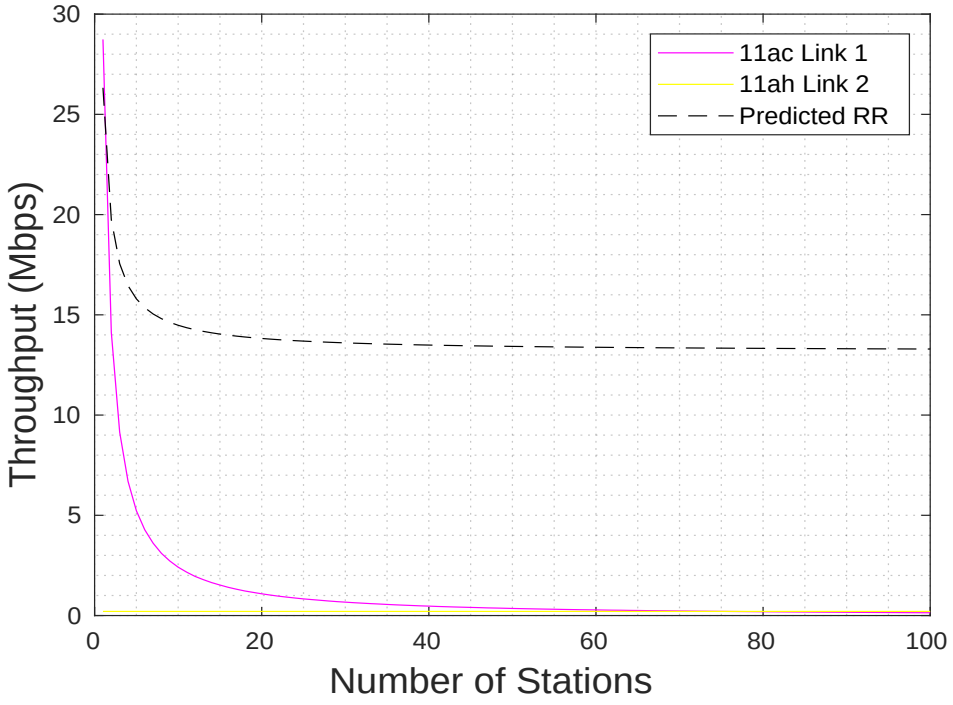

Figure 4.91: Test 2B MT vs Link Throughput Performance Comparison

4.7.2.5 MT Test 2C: Results

The 11ac-11ah bond throughput results for the MT algorithm in Test 2C are presented in Figure 4.92 below. The achieved MT algorithm throughput is plotted as the solid green line and the maximum theoretical bond throughput as the solid black line. The overall mean bond throughput for all applied load combinations considered was 1.3074 Mbps. As seen from the graph a familiar non-linear decreasing curve was seen for both metrics as the applied bond load was increased with a visible difference between the two outputs. At the initial applied load of $\{n_{11ac} = 1, n_{11ah} = 50\}$ the achieved throughput was measured as 28.74 Mbps while the maximum theoretical throughput was slightly higher at 28.94 Mbps. As the applied load was increased both throughput metrics reduced exponentially. At the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 50\}$ the achieved bond throughput reduced to a minimum of 0.2042 Mbps and the maximum theoretical bond throughput was reduced to 0.343 Mbps. The overall mean link throughput difference for all applied load scenarios considered was 3.4689 Mbps.

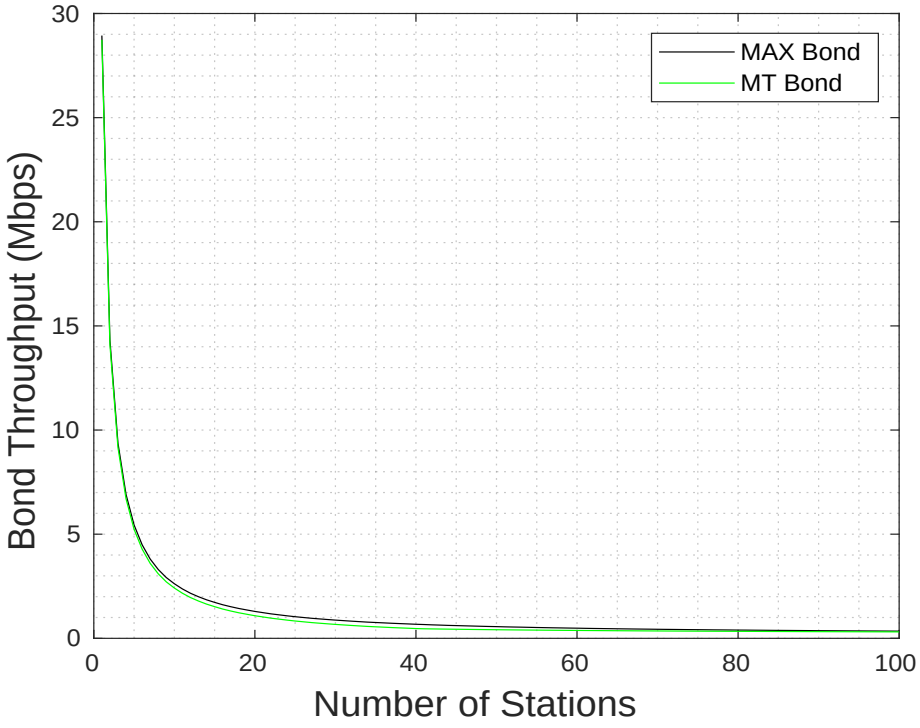

Figure 4.92: Test 2C MT Algorithm Bond Throughput Results

The link bandwidth utilisation results for the MT algorithm in Test 2C are presented in Figure 4.93 below. The overall mean link utilisation for all applied load combinations considered was 88.389%. For applied loads from $\{n_{11ac} = 1, n_{11ah} = 50\}$ to $\{n_{11ac} = 40, n_{11ah} = 50\}$ the bond utilisation was 100% but as the load incremented to $\{n_{11ac} = 41, n_{11ah} = 50\}$ the output decreased suddenly to 68.95%. As the load was further raised towards the heaviest prescribed level of $\{n_{11ac} = 100, n_{11ah} = 50\}$ the throughout efficiency increased non-linearly to a final value of 89.4%.

Initially only a single link was attached to the bond, but as the load incremented to $\{n_{11ac} = 41, n_{11ah} = 50\}$ the bond throughput began to exceed that of the fastest single link, to which the algorithm reacted by re-adding the slower link. This had the impact of suddenly decreasing the bandwidth utilisation. Then as the load was further increased, the performance disparity between links was reduced which had the effect of gradually increasing the utilisation.

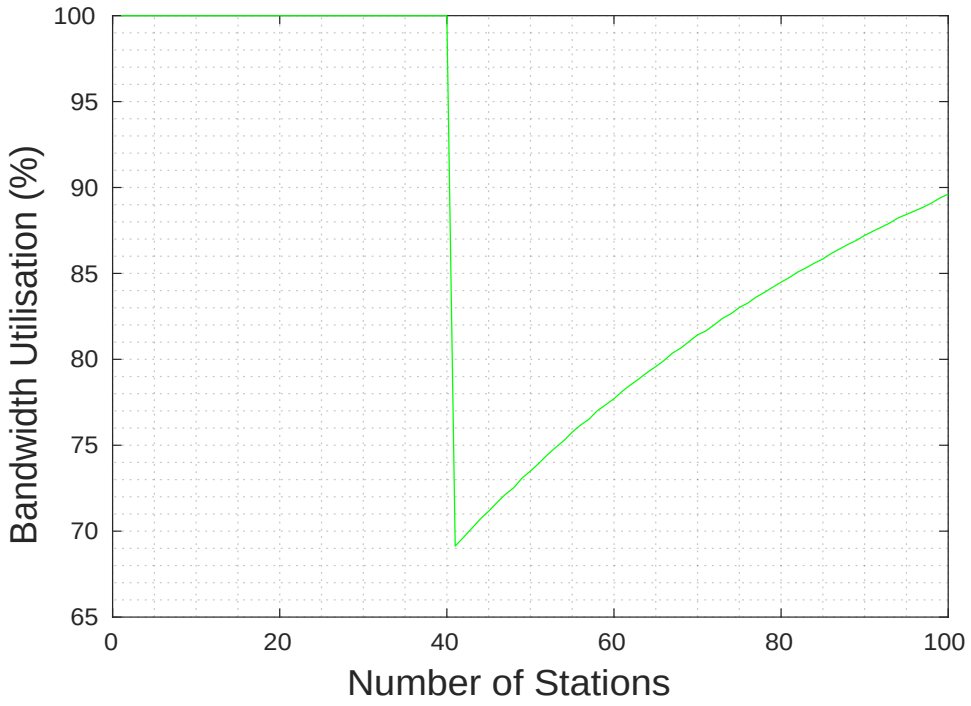

Figure 4.93: Test 2C MT Algorithm Link Bandwidth Utilisation Results

The bond delay results for the MT algorithm in Test 2C are presented in Figure 4.94 below. The overall mean bond delay for all applied load combinations considered was 58093.1452 μs. As seen between applied loads of $\{n_{11ac} = 1, n_{11ah} = 50\}$ and $\{n_{11ac} = 40, n_{11ah} = 50\}$ the bond delay followed the 11ac link delay which increased non-linearly from 0.0032 μs to 34330 μs. Then as the load was incremented to $\{n_{11ac} = 41, n_{11ah} = 50\}$ the bond delay began to follow the 11ah link delay which was constant at 78330 μs. At a load of $\{n_{11ac} = 76, n_{11ah} = 50\}$ the bond delay then began to follow the 11ac link delay again and increased non-linearly to a maximum of 116600 μs at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 50\}$. The overall mean link delay difference for all applied loads considered was 12719.779 μs.

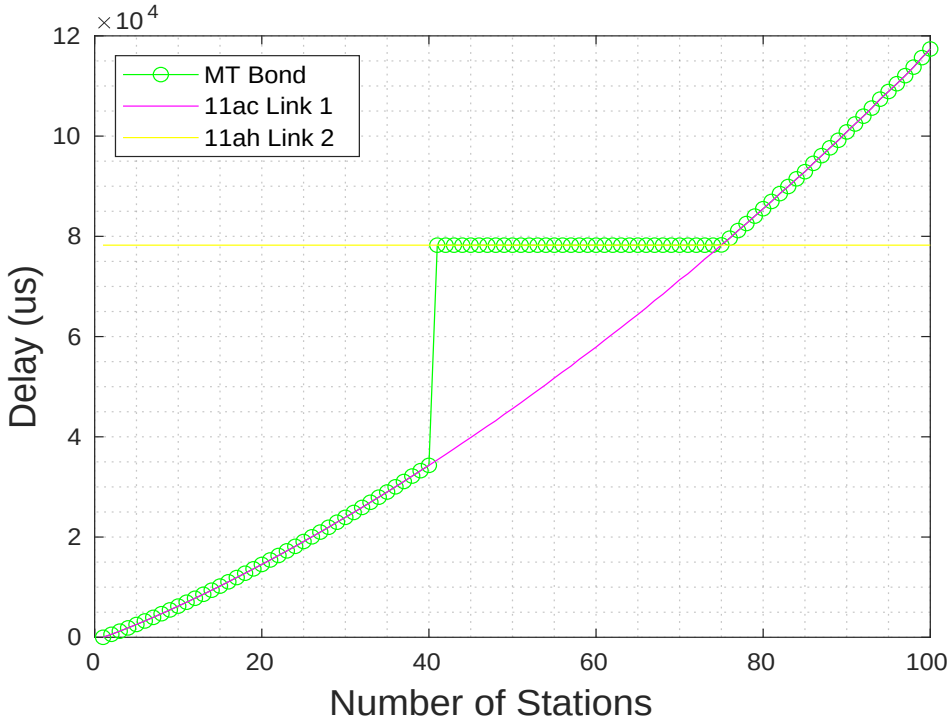

Figure 4.94: Test 2C MT Algorithm Bond Delay Results

4.7.2.6 MT Test 2C: Discussion

The main finding from Test 2C was that the dynamic MT algorithm provided a substantial mean bond throughput improvement of 68.04 % and a link bandwidth utilisation gain of 8.91% compared to the traditional static round robin configuration. The mean throughput for the MT algorithm was 1.2297 Mbps with a mean utilisation of 76.76% while the achieved round robin throughput was just 0.7406 Mbps with a utilisation of 70.479%. As seen from the graph below in Figure 4.95 at the initial applied load of $\{n_{11ac} = 1, n_{11ah} = 50\}$ the MT throughput of 28.94 Mbps was significantly higher compared to the achieved round robin throughput of 10.5 Mbps. As the load was increased to $\{n_{11ac} = 40, n_{11ah} = 50\}$ the MT throughput reduced non-linearly to 0.4694 Mbps which was slightly higher than the 0.4630 Mbps achieved by round robin at the same load. Then as the load incremented to $\{n_{11ac} = 41, n_{11ah} = 50\}$ the throughput of both methods became equal at approximately 0.45 Mbps. As the load was further increased the MT output reduced at a faster rate compared to round robin with both decreasing to 0.206 Mbps and 0.309 Mbps respectively at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 50\}$.

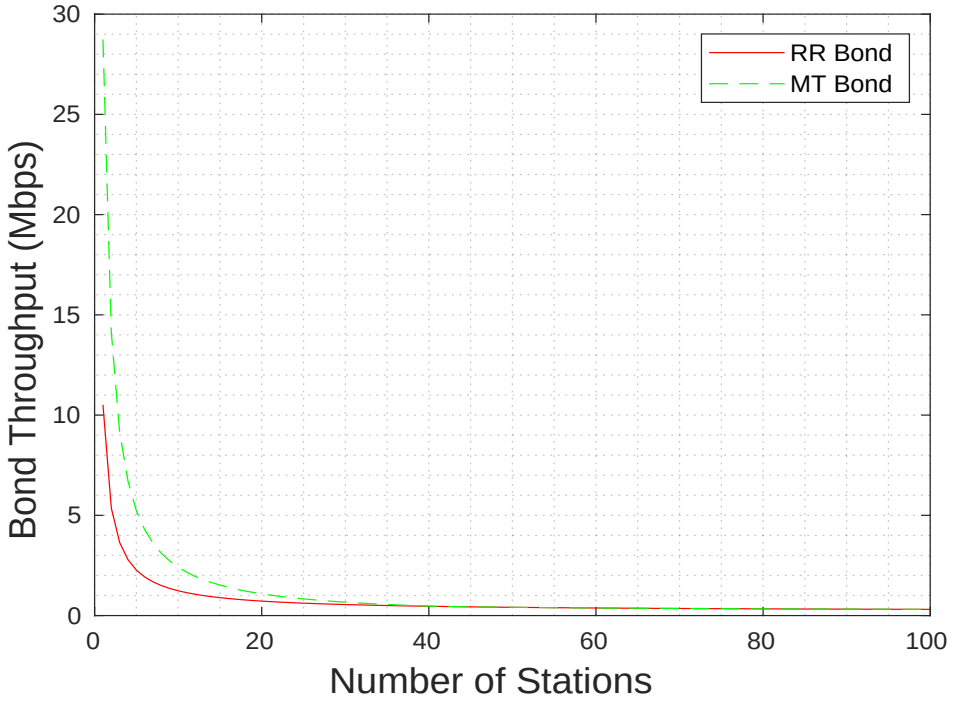

Figure 4.95: Test 2C MT vs RR Algorithm Bond Throughput Performance Comparison

In Test 2C we saw the dynamic link selection algorithm react to the predicted bond throughput increasing above the throughput of the fastest single link by reattaching the slower 11ah device. During the test for low loads only the single 11ac slave was in use because as seen below in Figure 4.96 the predicted bond throughput was lower than the specified threshold for applied loads from $\{n_{11ac} = 1, n_{11ah} = 50\}$ to $\{n_{11ac} = 40, n_{11ah} = 50\}$. Then as the applied load incremented to $\{n_{11ac} = 41, n_{11ah} = 50\}$ the predicted throughput increased above that of the fast 11ac link and the MT algorithm responded as desired by reattaching the problematic 11ah link. Because the predicted throughput did not reduce again below the given threshold both 11ac and 11ah links were used for the remainder of the simulation. By resorting to the use of a single fast 11ac link the MT algorithm avoided the counter productive use of bonding at applied loads where $\{n_{11ac} <= 40, n_{11ah} = 50\}$ and by doing so increased the overall mean bond throughput and link utilisation.

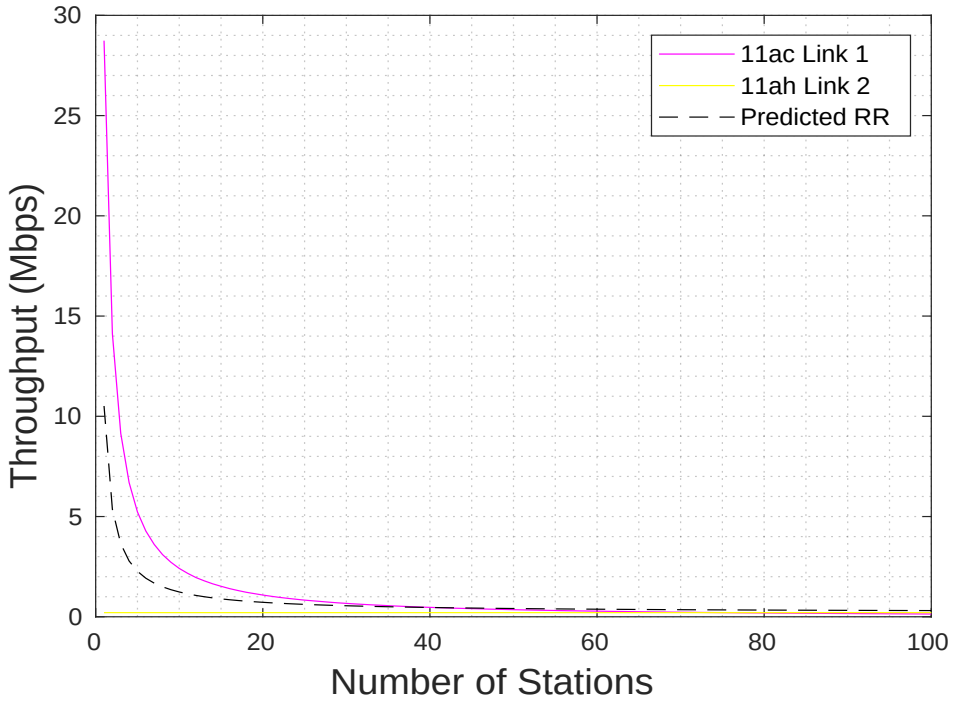

Figure 4.96: Test 2C MT vs Link Throughput Performance Comparison

4.7.2.7 MT Test 2D: Results

The bond throughput results for the MT algorithm in Test 2D are presented in Figure 4.97. The achieved MT bond throughput is plotted as the solid green line and the maximum theoretical bond throughput as the solid black line. The overall mean achieved bond throughput for all applied load combinations used was 1.2427 Mbps. As seen below both throughput metrics decreased non-linearly as the applied bond load was raised with the achieved MT throughput keeping close to the maximum theoretical throughout. At the initial applied load of $\{n_{11ac} = 1, n_{11ah} = 100\}$ the achieved throughput was measured at 28.74 Mbps and the maximum theoretical throughput was slightly higher at 28.82 Mbps. As the bond load was increased the achieved throughput decreased exponentially to a minimum of 0.1655 Mbps at $\{n_{11ac} = 100, n_{11ah} = 100\}$ while the maximum theoretical bond throughput was reduced to 0.221 Mbps. The overall mean link throughput difference for all applied loads considered was 0.0146 Mbps.

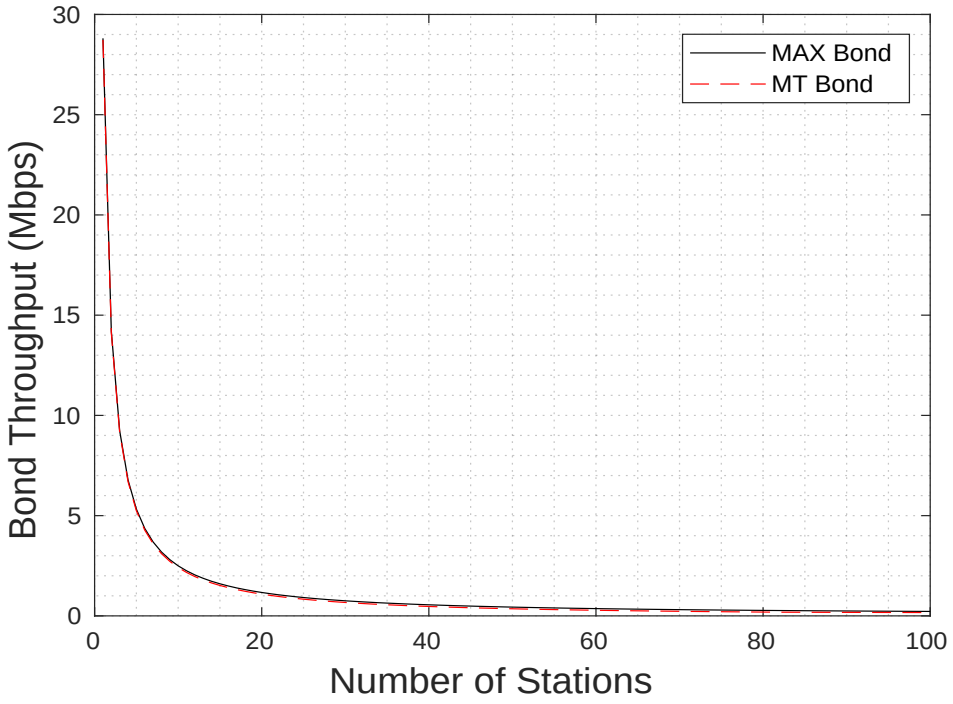

Figure 4.97: Test 2D MT Algorithm Bond Throughput Results

The link bandwidth utilisation results for Test 2D are shown in Figure 4.98. As seen the 11ac-11ah bond utilisation was 100% for applied loads of $\{n_{11ac} <= 2, n_{11ah} = 100\}$ but as the load incremented to $\{n_{11ac} = 83, n_{11ah} = 100\}$ the output reduced suddenly to a minimum of 69.37% before then increasing non-linearly to 74.55% at the heaviest considered bond of $\{n_{11ac} = 100, n_{11ah} = 100\}$. The overall mean link bandwidth utilisation for all applied load combinations considered was 94.673%.

Between applied loads of $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 82, n_{11ah} = 100\}$ only a single link was attached to the bond, but as the load incremented to $\{n_{11ac} = 82, n_{11ah} = 100\}$ the bond throughput began to exceed that of the fastest single link, and the algorithm reacted by re-attaching the slower second link. This also had the impact of suddenly decreasing the bandwidth utilisation. Then as the load was further increased, the performance disparity between links was reduced which had the effect of gradually increasing the utilisation.

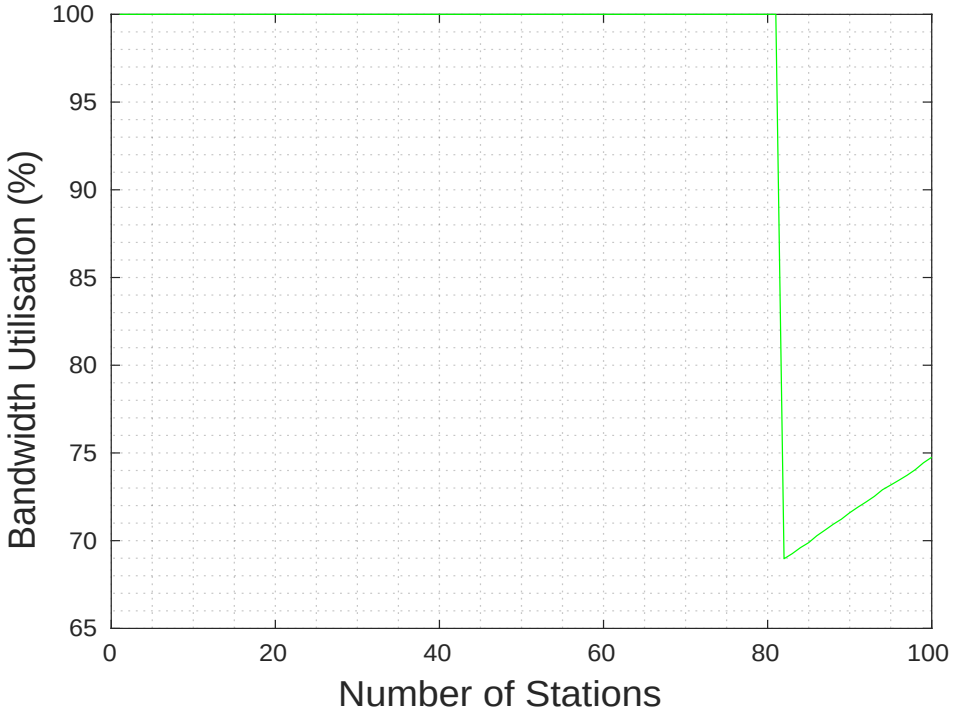

Figure 4.98: Test 2D MT Algorithm Link Bandwidth Utilisation Results

The bond delay results for the MT algorithm in Test 2D are shown in Figure 4.99. The overall mean 11ac-11ah bond delay for all applied load combinations considered was 67905.41 μs. For loads between $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 82, n_{11ah} = 100\}$ the bond delay followed the 11ac delay and increased non-linearly from 0.0284 μs to 88220 μs. However as the load incremented to $\{n_{11ac} = 83, n_{11ah} = 100\}$ the bond delay began to follow the 11ac link which remained constant at 196000 μs as the load was further increased towards $\{n_{11ac} = 100, n_{11ah} = 100\}$. The overall mean link delay difference for all applied loads used was 17688.43 μs.

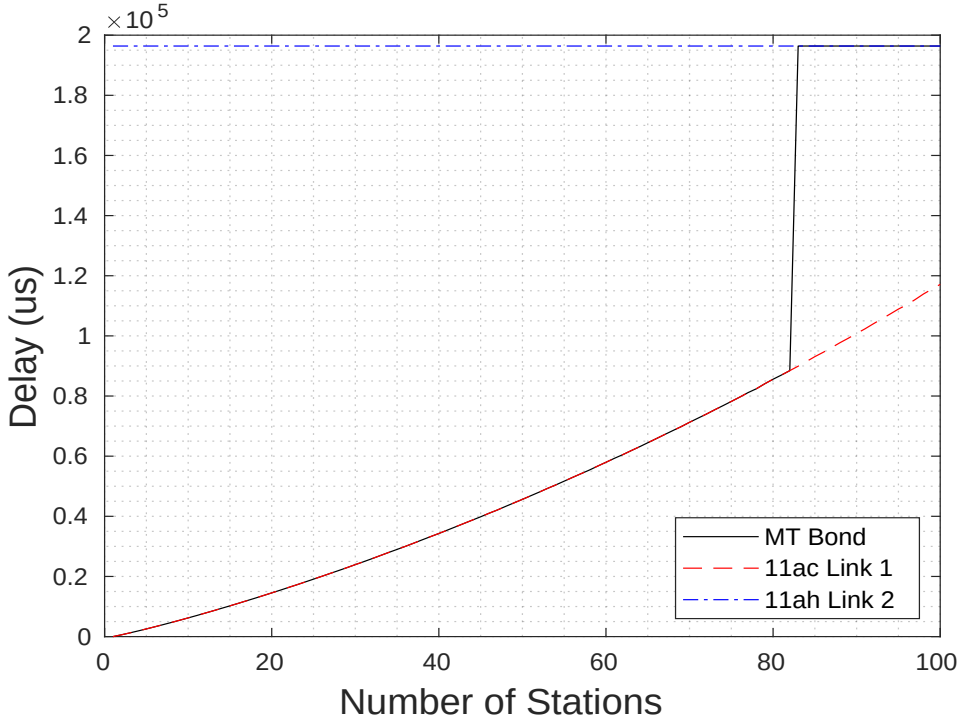

Figure 4.99: Test 2D MT Algorithm Bond Delay Results

4.7.2.8 MT Test 2D: Discussion

The results of the MT evaluation in Test 2D showed a substantial performance improvement of 142.01% in terms of the achieved bond throughput and also increased link utilisation by 27.6% compared to the traditional static round robin configuration. The mean bond throughput for the MT algorithm was 1.2427 Mbps with a mean link utilisation of 94.673% while the achieved mean throughput under round robin was 0.5135 Mbps with a mean utilisation of 74.197%. The throughput performance of both approaches is compared in Figure 4.100 with the MT throughput plotted as the dashed green line and the round robin throughput as the solid red line. As seen the MT throughput was higher than round robin for applied loads from $\{n_{11ac} = 1, n_{11ah} = 100\}$ to $\{n_{11ac} = 82, n_{11ah} = 100\}$, but as the load was incremented to $\{n_{11ac} = 83, n_{11ah} = 100\}$ the output of both methods reduced to approximately 0.1866 Mbps. As the applied load was further raised the MT throughput continued to follow round robin as both decreased non-linearly to a minimum of 0.1655 Mbps at the heaviest considered load of $\{n_{11ac} = 100, n_{11ah} = 100\}$.

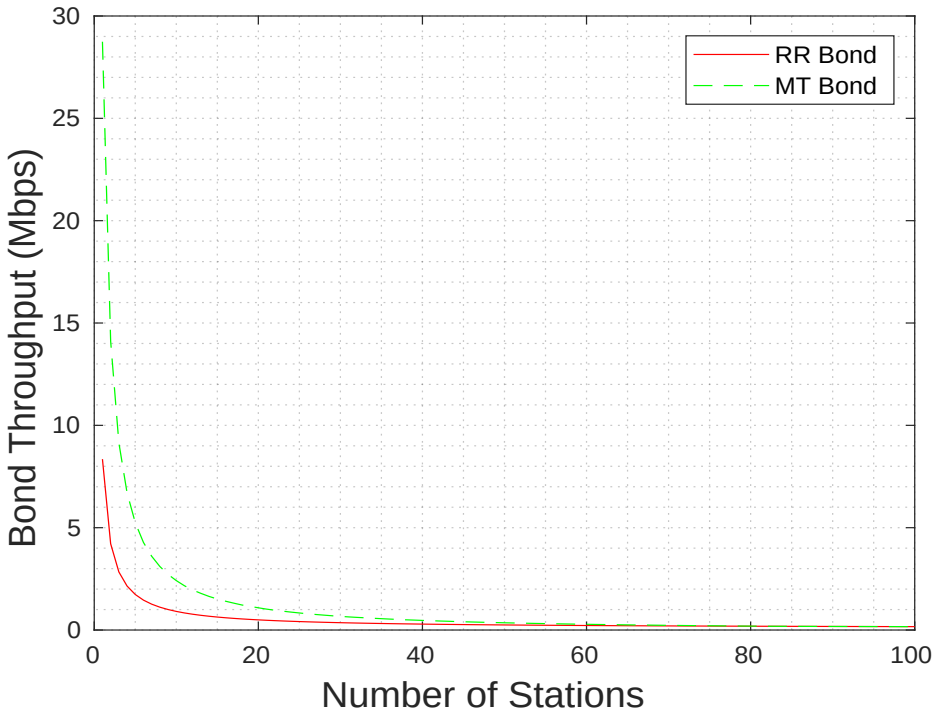

Figure 4.100: Test 2D MT vs RR Algorithm Bond Throughput Performance Comparison

In Test 2D the MT algorithm reacted to the predicted bond throughput increasing above the achievable throughput of the single fastest available link by re-attaching the slower 11ah slave device. As seen in Figure 4.101 below for applied loads between $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 82, n_{11ah} = 100\}$ only the single 11ac link was attached to the bond, but as the load was incremented to $\{n_{11ac} = 83, n_{11ah} = 100\}$ the above condition was met and the previously problematic slower 11ah slave was re-added. By avoiding the counter productive and wasteful use of bonding seen at loads from $\{n_{11ac} = 1, n_{11ah} = 100\}$ to $\{n_{11ac} = 82, n_{11ah} = 100\}$ and resorting to a single link the dynamic link selection algorithm significantly increased the overall mean bond throughput and mean bond throughput efficiency by comparison to the traditional static link configuration.

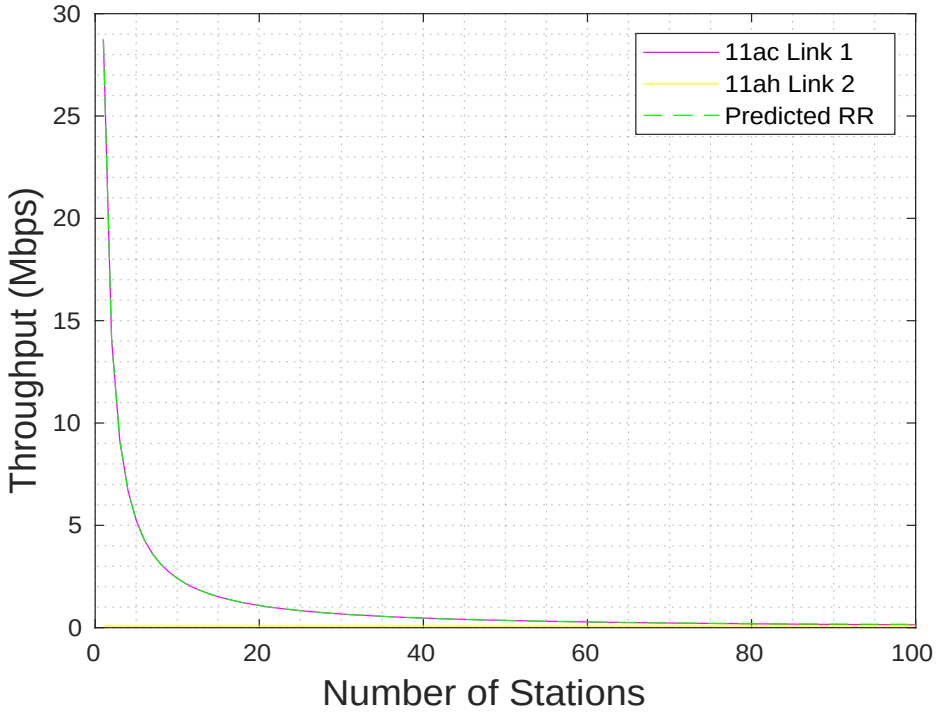

Figure 4.101: Test 2D MT vs Link Throughput Performance Comparison

4.8 MT Results Evaluation

This section provides an extensive overview and evaluation of the results from the MT algorithm testing. First the MT algorithm results for the dual-11ac and heterogeneous 11ac-11ah interface bonds are discussed individually and the performance compared against the traditional static bond configuration as per the round robin study in Chapter 3. Then in Section 4.8.3 there is a detailed comparison of the proposed algorithm performance against the state of the art throughput-gap-based approach to link selection by Nam.

4.8.1 Homogeneous 11ac-11ac Bond

First we present the MT algorithm results for the homogeneous 11ac-11ac interface bond. The main finding was that in all four tests the predicted-throughput-based dynamic link selection algorithm provided a substantial performance gain of 42.18% in terms of the mean bond throughput and 15.17% for the mean link bandwidth utilisation when compared to the static round robin bond configuration. The overall achieved mean bond throughput for the dual-11ac bond was 8.6517 Mbps with a mean link bandwidth utilisation of 96.797%.

In general the MT algorithm provided a substantial performance gain and was able to increase both the mean bond throughput and the mean link bandwidth utilisation in all of the test conducted for the homogeneous 11ac-11ac bond. The mean results for the key performance metrics are summarised in Table 4.6 below, alongside the 99.9% confidence interval.

	Test 1A	**Test 1B**	**Test 1C**	**Test 1D**
Bond T.put (Mbps)	2.48 ±0.1038	29.34 ±0.423	1.491 ±0.018	1.3 ±0.018
T.put Diff. (Mbps)	0.0018 ±4.07 * 10^{-6}	1.234 ±0.0548	0.195 ±0.0037	0.076 ±0.002
B.width Util. (%)	99.899 ±0.0084	99.65 ±0.0038	93.5 ±0.0055	94 ±0.008
Bond Delay (us)	50263±602	134.774±1.746	56864 ±1104	79075 ±1002
Delay Diff. (us)	115 ±31	1134.773 ±1.62	12744 ±237	28858 ±215

Table 4.6: Summary of Dynamic MT Algorithm Results for Homogeneous 11ac-11ac Bond

The main findings from each individual sub-test performed for the homogeneous 11ac-11ac bond are summarised as follows:

- Test 1A: The performance of the dynamic MT algorithm in the balanced load benchmark test was equivalent to that of the static bond configuration with an achieved mean bond throughput of 2.478 Mbps and mean link utilisation of 99.99%. As per the round robin experiments in Chapter 3 the balanced load used in Test 1A provided the ideal conditions for the 802.11 bond throughput, and therefore as desired the MT algorithm threshold for removing a problematic bond link was not met and the performance was equivalent to the static bond configuration throughout.

- Test 1B: In the first unbalanced applied load scenario the dynamic MT link selection algorithm outperformed the static configuration by 48.46% in terms of the mean bond throughput and also increased the link bandwidth utilisation by 53.2%. In this test the predicted bond throughput was below that of the fastest bond link for applied loads of $\{n_1 >= 8, n_2 = 1\}$ for which the MT algorithm reverted to the use of a single link and by doing so increased the mean bond throughput and link bandwidth utilisation significantly.

- Test 1C: In the second unbalanced load scenario the MT algorithm outperformed the static round robin bond in terms of mean throughput by 20.06% and also increased the mean link bandwidth utilisation by 5.04%. Here the predicted bond throughput was lower than the throughput of the single fastest 802.11 link for applied loads of $\{n_{11ac} <= 14, n_{11ah} = 50\}$ where MT algorithm responded by removing the offending slower link in order to increase the overall mean bond throughput and link bandwidth utilisation.

- Test 1D: In the last unbalanced scenario for the dual-11ac interface bond the dynamic MT algorithm outperformed the static configuration by 51.51% in terms of the mean bond throughput and by 15.36% in terms of link bandwidth utilisation. In this case the predicted bond throughput was below that of the fastest bond link for applied loads of $\{n_{11ac} <= 39, n_{11ah} = 100\}$ and therefore the algorithm switched to the use of a single link and again increased the mean bond throughput and link bandwidth utilisation significantly.

Notable, due to the predicted-throughput-based decision making approach, the dynamic MT algorithm was able to successfully guard against the counter productive scenario where for certain levels of applied load the aggregated 802.11 bond throughput is reduced to below the throughput of the single fastest available link.

4.8.2 Heterogeneous 11ac-11ah Bond

Next we present the MT evaluation results for the homogeneous 11ac-11ac interface bond. Here the main finding was that across all four tests the predicted-throughput-based dynamic MT link selection algorithm provided a substantial performance gain of 8.19% in terms of the mean bond throughput and increased the mean link bandwidth utilisation by 12.53% compared to the traditional static bond configuration. The average achieved mean bond throughput in all four tests was 4.4379 Mbps with a mean link bandwidth utilisation of 88.62%.

The mean values of the key bond-related performance metrics are summarised in Table 4.7 below. Again as seen during the simulations the proposed MT algorithm was able to successfully increase both the mean achieved bond throughput and the overall mean link bandwidth utilisation in each of the individual applied load tests performed.

	Test 2A	**Test 2B**	**Test 2C**	**Test 2D**
Bond T.put (Mbps)	1.324 ±0.0178	13.87 ±0.82	1.307 ±0.0296	1.2427 ±0.0148
T.put Diff. (Mbps)	0.59 ±0.062	12.1 ±0.677	3.47 ±0.00098	0.0146 ±0.00085
B.width Util. (%)	73.7 ±0.0076	97.734 ±0.0092	88.39 ±0.0068	94.67 ±0.0094
Bond Delay (us)	85299 ±1594	50197.75 ±1617	58093 ±562	67905 ±884
Delay Diff. (us)	35090 ±592	50197.74 ±592	12720 ±251	17688 ±442

Table 4.7: Summary of Dynamic MT Algorithm Results for Heterogeneous 11ac-11ah Bond

The key findings from each individual sub-test for the heterogeneous 11ac-11ah interface bond are summarised as follows:

- Test 2A: For the balanced applied load the dynamic MT algorithm marginally increased the mean bond throughput by 1.92% and the link utilisation by 1% compared to the static bond configuration. In contrast to the homogeneous bond in Test 1A the heterogeneous DCF parameters creates a performance disparity between each 802.11 type even for the same level of applied link load which significantly reduced the achieved bond throughput and link bandwidth utilisation. For applied loads of $\{n_{11ac} = n_{11ah} = 1\}$ and $\{n_{11ac} = n_{11ah} = 2\}$ the predicted bond throughput was reduced to below that of the fastest single link and so the algorithm responded by removing the problematic and slower 11ah link from the bond.

- Test 2B: In first unbalanced load test the dynamic MT algorithm marginally outperformed the static bond by 0.17% in terms of the mean bond throughput and by 0.38% in terms of mean bandwidth utilisation. The predicted bond throughput was lower than the throughput of the fastest single link for applied loads where $\{n_{11ac} = 1, n_{11ah} = 1\}$ and so the MT algorithm acted as intended by reverting to the use of a single link thereby increasing the mean bond throughput and link bandwidth utilisation significantly.

- Test 2C: In the second unbalanced test the dynamic MT algorithm provided a mean bond throughput improvement of 68.04% and a link bandwidth utilisation gain of 8.91% compared to the traditional static bond configuration. In this case the predicted bond throughput was below that of the fastest bond link for applied loads of $\{n_{11ac} = 40, n_{11ah} = 50\}$ and therefore the algorithm switched to the use of a single link and again increased the mean bond throughput and link bandwidth utilisation significantly.

- Test 2D: In the final unbalanced load test the MT algorithm showed a performance increased of 142.01% in terms of the achieved bond throughput and also increased link utilisation by 27.6% when compared to the static round robin bond. Here the predicted bond throughput was lower than the throughput of the fastest individual 802.11 link for applied loads of $\{n_{11ac} <= 82, n_{11ah} = 100\}$, and therefore the MT algorithm acted as desired by removing the offending link in order to increase the mean bond throughput and link bandwidth utilisation.

Again as per the MT tests for the homogeneous dual-11ac interface bond, by virtue of the predicted-throughput-based method the dynamic MT algorithm was able to explicitly guard at all times against the counter productive scenario where at certain levels of applied load the slow-down effect is so pronounced that the compound bond throughput is reduced to below the throughput of the single fastest available 802.11 link.

4.8.3 Algorithm Performance Comparison

In this section we provide a detailed comparison between the performance of the proposed MT algorithm and that of the state of the art dynamic link selection approach by Nam.

The main research finding from the performance comparison was that in all eight tests the predicted-throughput-based dynamic MT algorithm outperformed the contemporary throughput-gap-based Nam solution by 6.16% in terms of mean bandwidth utilisation and by 0.79% in terms of the mean bond throughput. The MT algorithm provided an overall increase of 13.89% and 28.42% respectively when compared to the traditional static round robin bond configuration. The overall mean performance results for the two dynamic link selection methods are compared in Table 4.8 below.

	Bond Throughput (Mbps)	**Link Bandwidth Utilisation (%)**
Static RR Bond	5.09	81.40
Dynamic Nam Bond	6.48	87.32
Dynamic MT Bond	6.54	92.71

Table 4.8: Summary of Dynamic vs Static Bond Performance Results

Although designed for use with MPTCP and end-to-end interface bonds at the Transport Layer, the throughput-gap-based link selection algorithm by Nam performed well at Layer 2 for the Link Layer point-to-point topology assumed in the simulations, achieving a 7.28% increase to the link bandwidth utilisation and a 26.81% increase to the bond throughput when compared to the static bond configuration.

However the Nam algorithm failed to guard specifically against the counter productive scenario where the achieved bond throughput is reduced below that of the single fastest available link due to the in-order delivery requirement and associated slow-down effect. For Nam there is no reliable way to identify this damaging scenario from the value of the bond throughput gap metric used and the algorithm is left vulnerable to the condition. As shown in the following sections the strength of the presented MT algorithm lies in its knowledge of the predicted bond throughput allowing it to identify and avoid the use of the technique at known problematic loads.

The benchmark balanced applied load scenarios in Tests 1A and 2A are of particular interest as they serve to demonstrate the difference in performance between the homogeneous 11ac-11ac and heterogeneous 11ac-11ah bonds, as well as the difference between the two dynamic link selection approaches. In the benchmark testing for the homogeneous 11ac-11ah bond the two link selection algorithms responded in a similar way to the balanced applied load used:

- Nam Test 1A: For all applied loads considered the homogeneous 11ac-11ac bond throughput gap was negligible such that the Nam threshold of 0.9 for triggering the removal of

problematic bond links was never met and both 11ac slave links were used at all times during the simulations. The dynamic link selection algorithm performance was therefore equivalent to static round robin with an overall mean bond throughput of 2.47 Mbps and mean link utilisation of 99.99%.

- MT Test 1A: As show in Figure 4.66 in Section 4.7.1.2 above for all applied loads considered in the test the bond throughput was maximised and always faster than the single fastest available link such that the threshold defined by the algorithm for removing problematic bond links was never met and both 11ac slave links were used throughout. Similar to Nam Test 1A, the dynamic MT algorithm performance was therefore equivalent to the static round robin performance with both achieving an overall mean throughput of 2.47 Mbps and mean link utilisation of 99.99%.

By contrast in Test 2A when the balanced load was applied on the heterogeneous 11ac-11ah bond the two link algorithms responded differently:

- Nam Test 2A: As seen in Figure 4.36 of Section 4.4.2.1 above, for the balanced scenario the maximum value of the bond throughput gap in was 0.54, as such the threshold of 0.9 set for removing problematic bond links was never exceeded and both 11ac and 11ah slave links were used throughout the simulation. Similar to Test 1A the dynamic Nam algorithm performance was therefore equivalent to the static bond configuration with both achieving an overall mean bond throughput of 1.29 Mbps and mean bandwidth utilisation of 72.9%.
- MT Test 2A: By contrast in the MT evaluation in Test 2A at applied loads where $\{n_{11ac} = n_{11ah} <= 2\}$ the predicted bond throughput was less than the throughput the single fastest available link, as seen in Figure 4.86 of Section 4.7.2.2 above. Therefore the MT algorithm removed the slower 11ah link at the above loads which increased the achieved mean bond throughput by 5.72% and also increased the mean bandwidth utilisation significantly by 33.05% compared to the throughput-gap-based Nam approach.

Test	Applied Loads
1A	$\{n_1 = n_2 = n\}$
1B	$\{n_1 = n, n_2 = 1\}$
1C	$\{n_1 = n, n_2 = 50\}$
1D	$\{n_1 = n, n_2 = 100\}$
2A	$\{n_{11ac} = n_{11ac} = n\}$
2B	$\{n_{11ac} = n, n_{11ah} = 1\}$
2C	$\{n_{11ac} = n, n_{11ah} = 50\}$
2D	$\{n_{11ac} = n, n_{11ah} = 100\}$

Table 4.9: Summary of Applied Loads Used in Evaluation

Bond Metric	**Description**
Maximum Bond Throughput	Maximum throughput of the bond, given as the sum of the individual link throughputs
Achieved Bond Throughput	Bond throughput in Mbps calculated by counting the number of payload bits received over both links and dividing by the maximum link transmission time
Link Throughput Difference	Absolute difference in achieved throughput between the individual bond links given in Mbps
Link Bandwidth Utilisation	Calculated by dividing the achieved bond throughput by the maximum throughput
Bond Delay	The overall bond delay given in μs as the maximum of the individual link access delays
Link Delay Difference	The absolute difference in access delay between the individual links as a function of the applied load
Bond Throughput Gap	The bond throughput gap used as the main decision-making by authors Nam et al.

Table 4.10: Summary of Bond Performance Metrics

	Nam et al.	**Current Work**
Bonding Topology	End-to-end	Point-to-point
Bonding Layer	Layer 4	Layer 2
Experiment Type	Small hardware testbed	MatLab simulation
Bonding Technology	Multipath TCP	Protocol Agnostic
802.11 Standards	11g and 11n	11ac and 11ah
Bond Function	Dynamic Interface Selection	Dynamic Interface Selection
Bond Scheduling	Static Round Robin	Static Round Robin
Link Manipulation	Linux Traffic Control	Varying # Competing Nodes

Table 4.11: Comparison of Nam Experiment Setup vs Current Work

4.8.3.1 Homogeneous 11ac-11ac Bond

First we present the comparison for the dual-11ac interface bond. The main finding was that in all four tests for the predicted-throughput-based MT algorithm outperformed the Nam approach in terms of the mean bond throughput by 42.18% and in terms of the mean bandwidth utilisation by 14.39%. The mean MT bond throughput was 8.6517 Mbps with a mean link bandwidth utilisation of 96.797%, compared to 6.085 Mbps and 84.62% for the Nam algorithm. The results are summarised in Table 4.12 below, and the performance of both link selection algorithms in each individual applied load test is discussed and compared on the following pages.

	Bond T.put (Mbps)	**T.put Diff. (Mbps)**	**B.width Util. (%)**	**Bond Delay (us)**	**Delay Diff (us)**
Test 1A Nam	2.478	0.0018405	99.898	50256.711	105.65
Test 1A MT	2.4783	0.0018053	99.899	50263.23	115
Test 1B Nam	29.3289	1.5502	99.507	181.6635	182.6612
Test 1B MT	29.3371	1.2338	99.625	134.593	134.593
Test 1C Nam	1.4754	0.32843	91.932	62142.14	26525.44
Test 1C MT	1.4914	0.19539	93.479	56864.06	12743.688
Test 1D Nam	1.2725	0.21799	88.394	115962.75	49014.35
Test 1D MT	1.3004	0.076292	94.188	79074.58	28857.60

Table 4.12: Nam vs MT Performance Comparison for Homogeneous 11ac-11ac Bond

4.8.3.1.1 Test 1A Comparison The main finding from the benchmark case was that the MT algorithm performance was equivalent to the Nam performance for the balanced load and homogeneous 11ac-11ac and 65Mbps-65Mbps bond used in the test. Both dynamic link selection methods achieved an overall mean bond throughput of 2.47 Mbps and mean link throughput utilisation of 99.99%.

The achieved throughputs are plotted in Figure 4.102. As per the simulation testing in Chapter 3 (see requirements (1), (2), and (3) from the round robin evaluation), the balanced applied load used in Test 1A produces the ideal conditions for the homogeneous 11ac-11ac and 65Mbps-65Mbps interface bond due to the equivalent link throughput performance at equal levels of applied load.

As a consequence there was no observable throughput slow-down and the respective thresholds defined by each algorithm for triggering link removal were not met at any of the applied loads. Therefore in Test 1A as desired and expected the performance of both dynamic link selection algorithms was equivalent and equivalent to the static bond configuration.

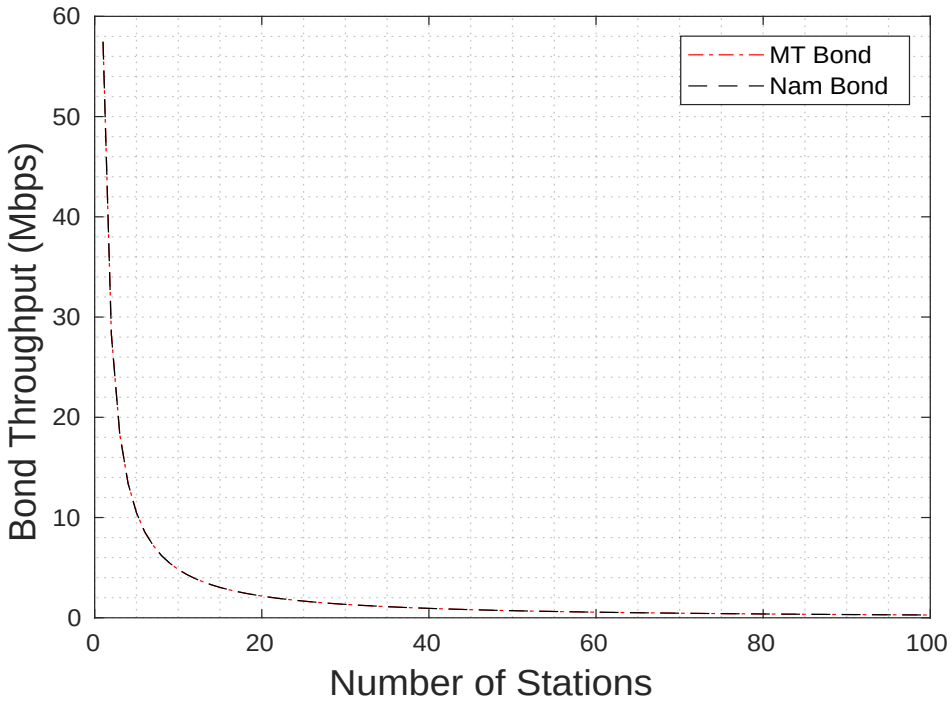

Figure 4.102: Test 1A Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.1.2 Test 1B Comparison The main finding from the unbalanced scenario in Test 1B was that the MT algorithm marginally outperformed Nam by 0.3% in terms of the mean bond throughput and also increased the bandwidth utilisation marginally by 0.12%. The mean bond throughput under the MT algorithm was 29.3371 Mbps with a mean link utilisation of 99.625% while the mean bond throughput achieved under Nam was 29.3289 Mbps with mean utilisation of 99.507%.

The results for both dynamic link selection algorithms are compared in Figure 4.103 below. For Nam at low loads from $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 8, n_2 = 50\}$ both 11ac links of the bond were in use as the bond throughput gap was less than the specified threshold. However as the applied load was incremented to $\{n_1 = 9, n_2 = 50\}$ the throughput gap exceeded 0.9 and the algorithm responded as expected by removing the slower 11ac link from the bond.

Similarly for the MT algorithm at low loads both 11ac bond links were attached but in contrast the slower 11ac link was removed slightly sooner at a load of $\{n_1 = 8, n_2 = 50\}$ when the predicted bond throughput was reduced below that of the fastest available single link. In delaying the re-attachment the MT algorithm avoided the counter productive scenario at $\{n_1 = 7, n_2 = 50\}$ and marginally increased *both* the overall mean bond throughput and mean bond link bandwidth utilisation.

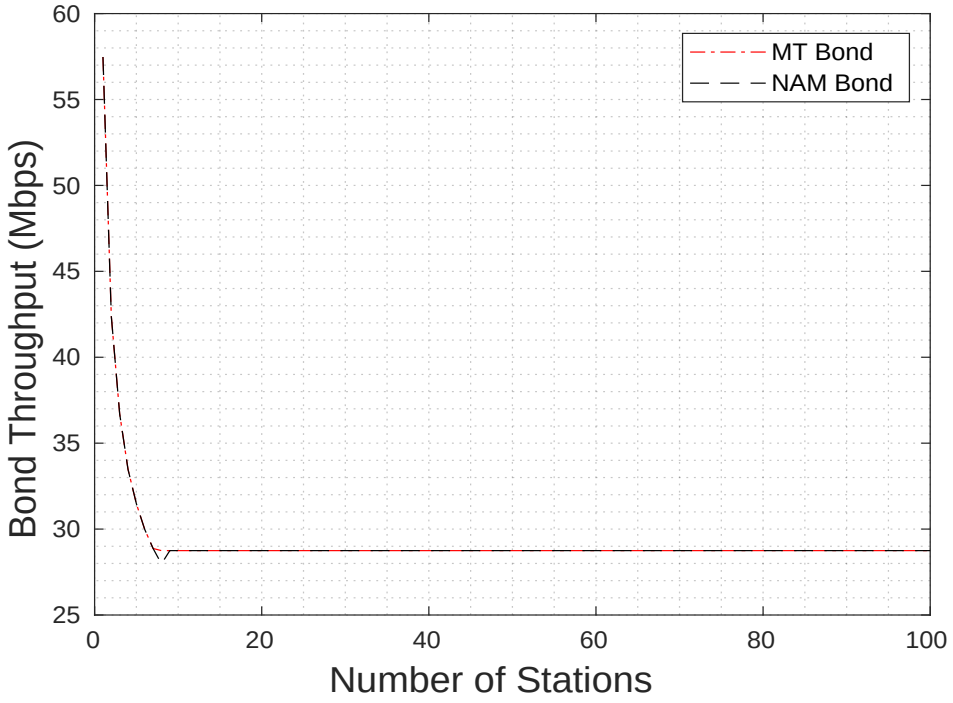

Figure 4.103: Test 1B Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.1.3 Test 1C Comparison The main finding from Test 1C was that the predicted-throughput-based MT link selection method outperformed Nam by 1.08% in terms of the mean bond throughput and also increased the link bandwidth utilisation by 1.68%. The overall mean bond throughput for the MT algorithm was 1.4914 Mbps with a mean bandwidth utilisation of 93.479%, while the mean throughput under Nam was 1.4754 Mbps with mean utilisation of 91.932%.

The results for both link selection algorithms are compared in Figure 4.104 below. In the Nam test for applied loads between $\{n_1 = 1, n_2 = 50\}$ and $\{n_1 = 7, n_2 = 50\}$ the measured bond throughput gap was decreasing yet still greater than 0.9 such that only the faster first 11ac bond link was in use. Then as the load was incremented to $\{n_1 = 8, n_2 = 50\}$ the throughput gap reduced below the threshold and the slower 11ac link was re-attached to the bond. However for applied loads from $\{n_1 = 8, n_2 = 50\}$ to $\{n_1 = 14, n_2 = 50\}$ the impact of the throughput slow-down effect due to the in-order delivery requirement was still so severe that the bond throughput was reduced below the achievable throughput of the fastest available 11ac link, thereby making the use of the technique not only pointless but also counter productive.

By contrast the MT algorithm was more cautious in its approach and in order to avoid the above scenario it delayed re-attaching the slower 11ac slave until an applied load of $\{n_1 = 15, n_2 = 50\}$ and in doing so increased the overall mean 11ac-11ac bond throughput as well as the mean bandwidth utilisation.

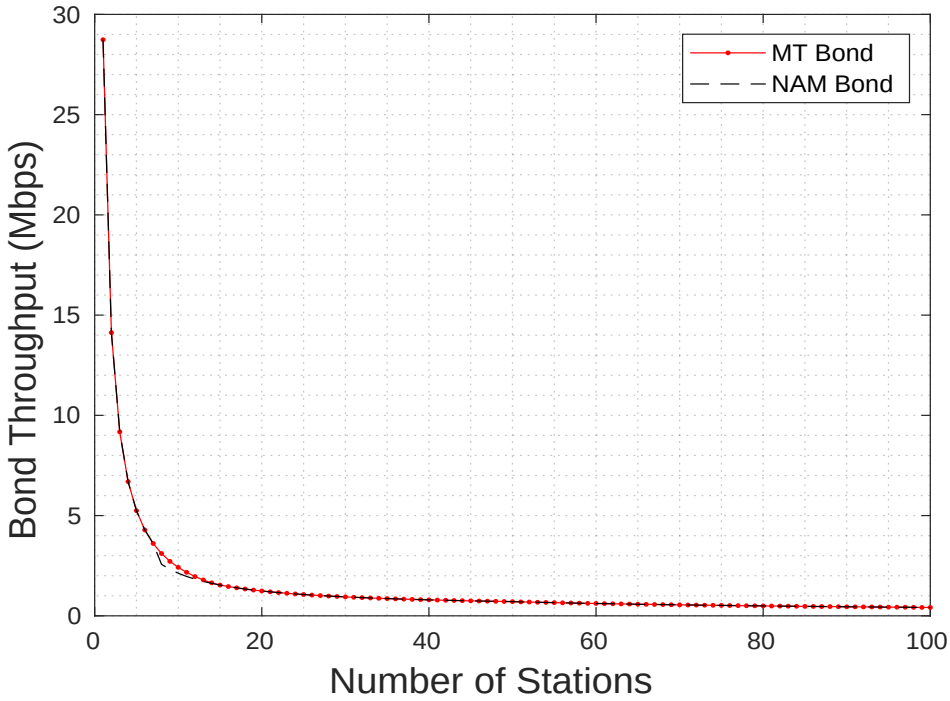

Figure 4.104: Test 1C Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.1.4 Test 1D Comparison The main finding from Test 1D was that the MT method outperformed the Nam algorithm by 5.75% in terms of the mean bond throughput and also increased the mean link utilisation by 33.65%. The overall mean bond throughput for the MT algorithm during the test was 1.242 Mbps with a mean link bandwidth utilisation of 94.9%, while the mean throughput under Nam was 1.174 Mbps with mean bandwidth utilisation of 71%.

The results for both link selection methods are compared in Figure 4.105 below. In the Nam test for applied loads between $\{n_1 = 1, n_2 = 100\}$ and $\{n_1 = 16, n_2 = 100\}$ the measured bond throughput gap was decreasing but greater than 0.9 and therefore only the faster first 11ac bond link was in use. As the applied load was incremented to $\{n_1 = 17, n_2 = 100\}$ the throughput gap reduced below the above threshold and the algorithm responded by re-attaching the slower 11ac slave link.

By contrast the MT algorithm was more cautious in its approach and did not re-attach the slower 11ac link until an applied load of $\{n_1 = 39, n_2 = 100\}$ when the predicted bond throughput increased above the throughput of the fastest available single link. In doing so the MT algorithm avoided the counter productive and wasteful use of the bond technique for applied loads from $\{n_1 = 17, n_2 = 100\}$ to $\{n_1 = 38, n_2 = 100\}$ which increased the overall mean bond throughput and mean link bandwidth achieved by the dual-11ac bond used.

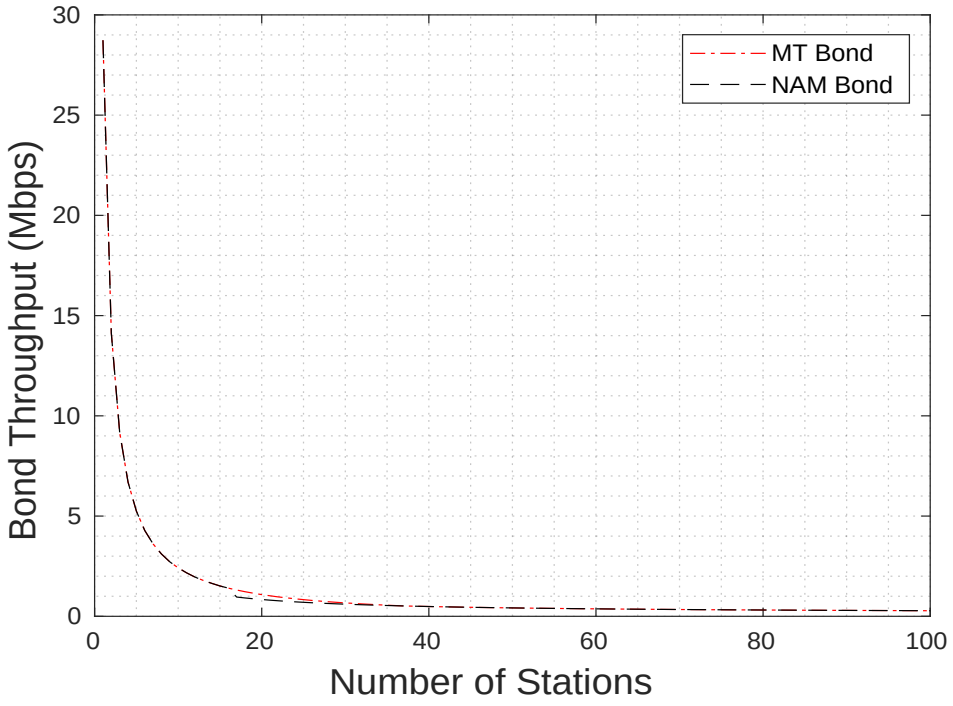

Figure 4.105: Test 1D Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.2 Heterogeneous 11ac-11ah Bond

Next we present the algorithm performance comparison for the heterogeneous 11ac-11ah interface bond. The main finding was that in all four tests for the heterogeneous 11ac-1ah bond the presented MT algorithm outperformed the Nam method in terms of the mean bond throughput by 2.49% and in terms of mean bandwidth utilisation by 11.17%. The achieved mean MT bond throughput was 4.4379 Mbps with a mean link bandwidth utilisation of 88.62% compared to 4.33 Mbps and 79.718% respectively for Nam. The results are summarised in Table 4.13 below, and the performance in each individual applied load scenario is discussed and compared in detail in the following sections.

	Bond T.put (Mbps)	**T.put Diff. (Mbps)**	**B.width Util. (%)**	**Bond Delay (us)**	**Delay Diff. (us)**
Test 2A Nam	1.299	0.5901	72.98	85312.46	35122.9
Test 2A MT	1.324	0.59008	73.705	85297.98	35089.9
Test 2B Nam	13.61	1.5315	98.06	50197.75	838.1
Test 2B MT	13.87	12.1047	97.734	50197.75	50197.74
Test 2C Nam	1.229	0.253	76.76	57953.63	12682.37
Test 2C MT	1.307	3.4689	88.38	58093.13	12719.77
Test 2D Nam	1.175	0.178	71.11	193429.54	98549.15
Test 2D MT	1.242	0.0146	94.673	67905.41	17688.43

Table 4.13: Nam vs MT Performance Comparison for Heterogeneous 11ac-11ah Bond

4.8.3.2.1 Test 2A Comparison The main finding from Test 2A was that the MT algorithm marginally outperformed Nam in terms of the mean bond throughput by 1.85 % and also increased the mean link utilisation marginally by 0.99 %. The overall achieved mean 11ac-11ah bond throughput under the MT algorithm was 1.324 Mbps with a mean bandwidth utilisation of 73.705% while the mean throughput for Nam was 1.299 Mbps with mean utilisation of 72.98%. The results for both approaches are plotted together in Figure 4.106 below.

The difference in behaviour was due to the comparatively more protective strategy of the MT algorithm apparent at an applied load $\{n_{11ac} = 1, n_{11ah} = 1\}$ where the slower 11ah link was removed due to the predicted bond throughput being reduced to less than that of the faster 802.11ac slave link. Then as the applied load incremented to $\{n_{11ac} = 2, n_{11ah} = 2\}$ the predicted bond throughput increased so that the above condition no longer held and the slower 11ah link was re-attached and stayed in active use for the remainder of the simulation.

By contrast the Nam throughput gap threshold of 0.9 for removing a problematic bond link was *never* exceeded and therefore both slave links were used throughout the test. The MT algorithm therefore avoided the severe slow-down effect due to the in-order delivery requirement at applied loads of $\{n_{11ac} = n_{11ah} = 1\}$ by resorting to a single link and thereby increased the mean bond throughput and mean link utilisation compared to the Nam approach.

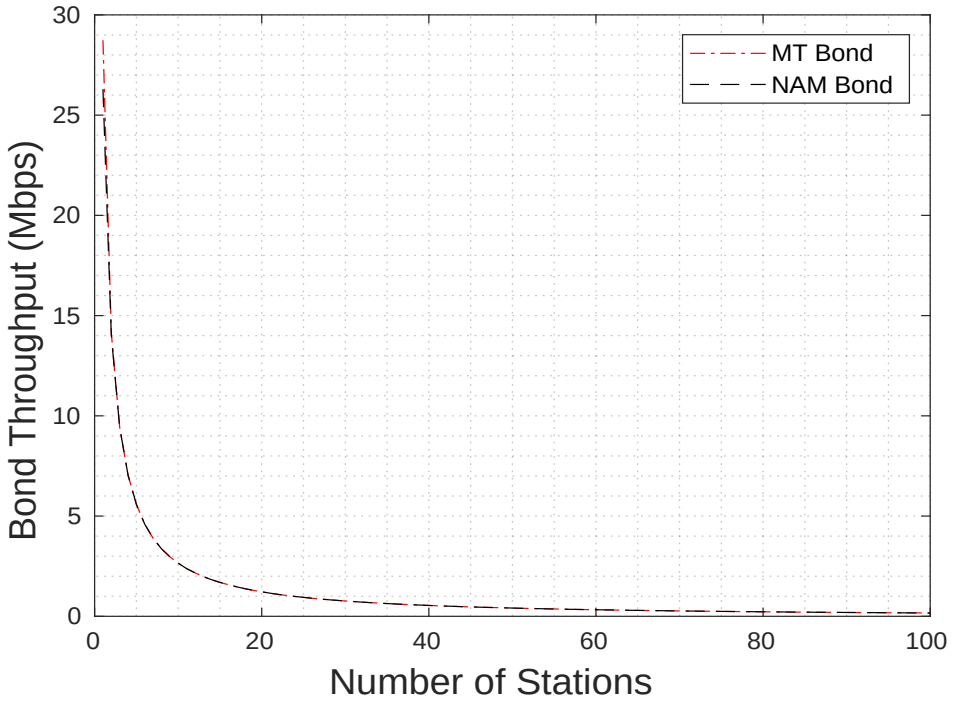

Figure 4.106: Test 2A Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.2.2 Test 2B Comparison In Test 2B the MT algorithm outperformed the Nam algorithm by 1.91% in terms of the mean bond throughput while decreasing the link bandwidth utilisation marginally by 0.33%. The overall mean bond throughput for the MT algorithm was 13.87 Mbps with a mean bandwidth utilisation of 97.734%, while the achieved mean throughput under Nam was 13.61 Mbps with mean utilisation of 98.06%.

The results for both link selection approaches are plotted in Figure 4.107 below. In the Nam test at applied loads between $\{n_{11ac} = 1, n_{11ah} = 1\}$ and $\{n_{11ac} = 16, n_{11ah} = 1\}$ both 11ac and 11ah slave links were attached as the measured bond throughput gap was still under the prescribed threshold. Then as the load was incremented to $\{n_{11ac} = 17, n_{11ah} = 1\}$ the throughput gap reduced below 0.9 and the algorithm responded as expected by removing the slower 11ah slave link.

By contrast for the MT algorithm at $\{n_{11ac} = 1, n_{11ah} = 1\}$ the predicted bond throughput was less than that of the fastest single link and therefore the slower 11ah slave link was not used. As the load was incremented to $\{n_{11ac} = 2, n_{11ah} = 1\}$ the above condition then no longer applied and the 11ah link was re-attached for the remainder of the test. By taking this step with no other action, the MT algorithm avoided the counter productive use of bonding at $\{n_{11ac} = 1, n_{11ah} = 1\}$ and increased the overall mean bond throughput while marginally reducing the overall link utilisation.

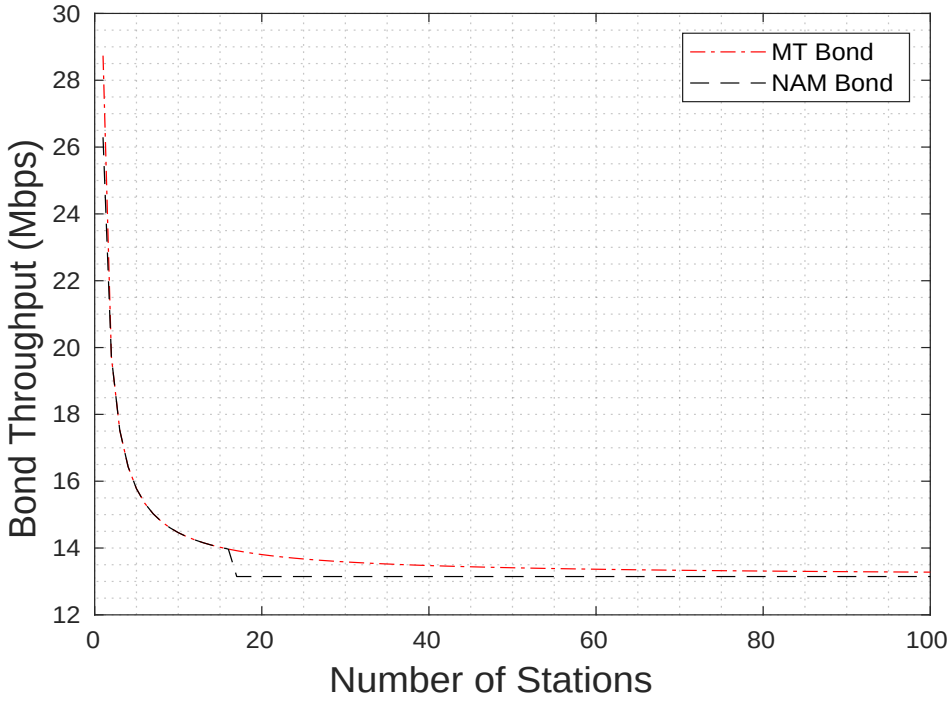

Figure 4.107: Test 2B Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.2.3 Test 2C Comparison The main finding from Test 2C was that the MT method outperformed the Nam approach by 6.32% in terms of the mean bond throughput and also increased the link bandwidth utilisation by 15.15%. The achieved mean bond throughput under the MT algorithm was 1.307 Mbps and the mean bandwidth utilisation was 88.389%, compared to 1.2297 Mbps and 76.76% respectively for the Nam algorithm.

The achieved throughput results for both approaches are compared in Figure 4.108 below. In the Nam test for the initial load of $\{n_{11ac} = 1, n_{11ah} = 1\}$ only the 11ac bond link was in use as the measured bond throughput gap was in excess of the prescribed threshold. Then as the load was increased to $\{n_1 = 12, n_2 = 1\}$ the bond throughput gap decreased below 0.9 and the algorithm responded as expected by re-attaching the slower 11h link.

By contrast the MT algorithm was more cautious in its approach. Similar to Nam, at relatively low levels of applied load only the 11ac link was attached as the predicted bond throughput was less than the throughput of the fastest available single link. The predicted bond throughput did not increase above this value until an applied load of $\{n_1 = 42, n_2 = 1\}$ to which point the algorithm reacted by re-attaching the slower 11ah link. In doing so it avoided the counter productive use of bonding between applied loads of $\{n_1 = 12, n_2 = 1\}$ and $\{n_1 = 41, n_2 = 1\}$ and increased the overall mean bond throughput and mean link bandwidth utilisation when compared to Nam.

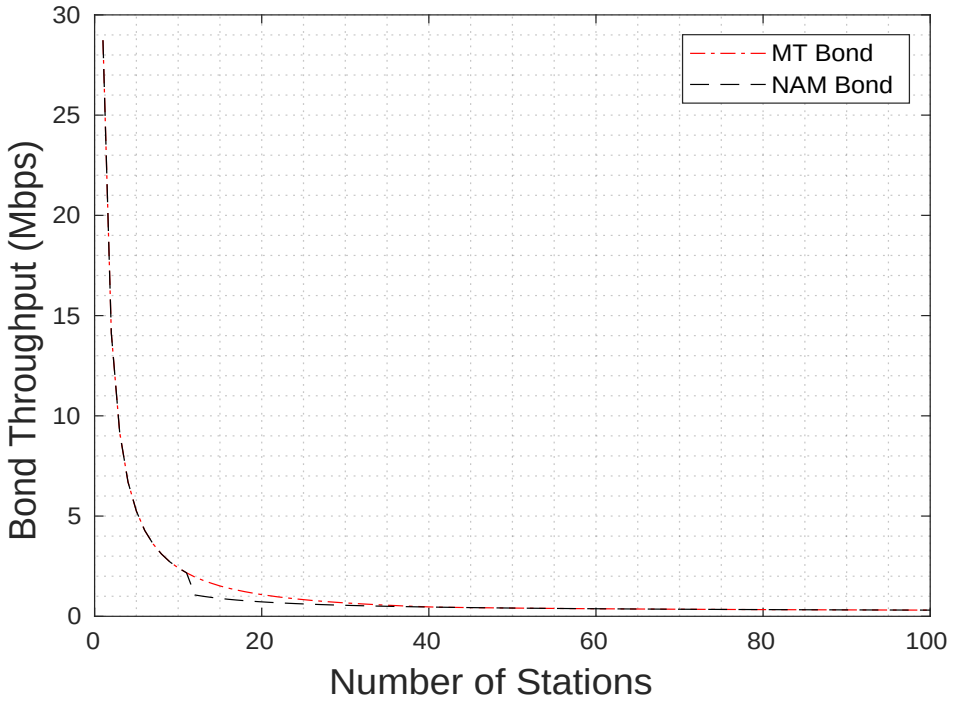

Figure 4.108: Test 2C Nam vs MT Algorithm Bond Throughput Comparison

4.8.3.2.4 2D: Comparison In Test 2D the main finding from the comparison was that the MT method outperformed the Nam algorithm by 5.72% in terms of the mean bond throughput and also increased the link bandwidth utilisation significantly by 33.05%. The overall mean bond throughput for the MT algorithm was 1.2427 Mbps with a mean bandwidth utilisation of 94.673%, compared to just 1.1755 Mbps and 71.154% respectively for the throughput-gap-based approach by Nam.

The results for both link selection methods are compared in Figure 4.105 below. In the Nam test for applied loads between $\{n_{11ac} = 1, n_{11ah} = 100\}$ and $\{n_{11ac} = 25, n_{11ah} = 100\}$ the bond throughput gap was decreasing but still greater than 0.9 such that only the faster first 11ac slave link was in use. Then as the applied load was incremented to $\{n_{11ac} = 26, n_{11ah} = 100\}$ the bond throughput gap reduced below the specified threshold and the Nam algorithm responded by re-attaching the slower 11ac slave link to the bond.

By contrast the MT algorithm was more cautious in its approach and did not re-attach the slower 11ac link until an applied load of $\{n_1 = 83, n_2 = 100\}$ when the predicted bond throughput increased above the throughput of the fastest available single link. In doing so the MT algorithm avoided the counter productive and wasteful use of bonding between applied loads of $\{n_{11ac} = 26, n_{11ah} = 100\}$ and $\{n_{11ac} = 82, n_2 = 100\}$, and thereby increased the mean bond throughput while significantly increasing the mean bandwidth utilisation.

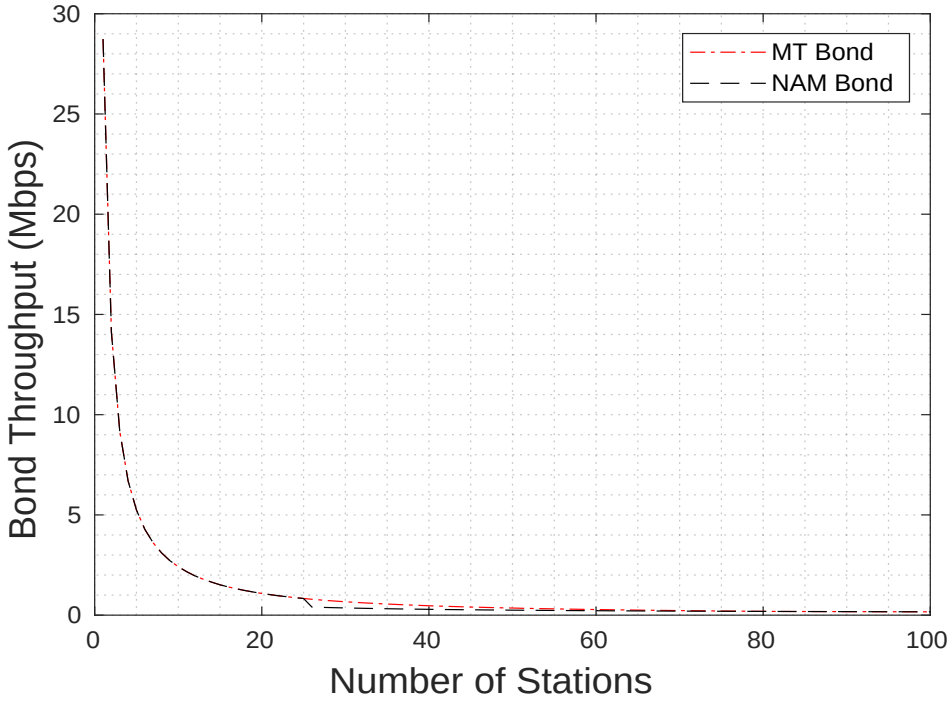

Figure 4.109: Test 2D Nam vs MT Algorithm Bond Throughput Comparison

4.9 Chapter Conclusion

This chapter investigated state of the art approaches for dynamic link selection for heterogeneous 802.11 devices leveraging the principles of SDN.

An overview of traditional wired SDN as defined in high performance data centre networks was provided in Section 4.2.1. In Section 4.2.2 there was an overview of SDN in wireless networks, including cellular WWAN, 802.11 WLANs, and wireless sensor networks and IoT. Section 4.2.3 introduced the mean thesis research of SDN-assisted interface bonding. The basis concepts were described, such as the centralisation of bond functionality in the SDN controller, as well as the main design challenges such as the dynamic wireless channel, the heterogeneous link throughput performance, and the inherent slow-down effect for in-order delivery.

Closely related research work was presented in Section 4.2.4, including approaches at all layers of the network protocol stack. There was a particular focus on the main thesis reference work: the architecture and decision-making framework for dynamic MPTCP link selection over heterogeneous 802.11 links using SDN by Nam et al, which was independently evaluated in Section 4.3 under a range of applied link load scenarios and individual achieved link throughputs. While the algorithm was shown to substantially increased the bond performance by 26.81% in terms of the mean bond throughput and by 7.28% in terms of the mean link bandwidth utilisation compared to the traditional static bond configuration, it did not provide specific protection against the counter productive scenario where the achieved bond throughput is reduced below the throughput of the fastest available link. The reason for this was the bond throughput gap used as the main performance metric, which was shown not to be capable of identifying, predicting, and avoiding the aforementioned damaging scenario, thereby reducing the achieved mean bond throughput and link bandwidth utilisation.

To address these limitations, novel, predicted-throughput-based algorithm was presented for the dynamic selection of heterogeneous 802.11 devices. The proposed algorithm was designed specifically to avoid the counter-productive bond throughput scenario seen at certain levels of applied load, in order to address the shortcomings of the contemporary approach taken by Nam et al, while at least matching the performance at other times. During the evaluation testing, the proposed algorithm was found to negligibly increase the mean bond throughput by 0.79%, but was found to increase the mean link bandwidth utilisation significantly by 6.16%.

CHAPTER 5

CONCLUSION

This thesis investigated dynamic 802 interface bonding for heterogeneous 802.11 devices leveraging Software Defined Networking (SDN), in particular, the centralised decision-making algorithms and bond related performance metrics used.

5.1 Research Benefits

To the best knowledge of the author, the work presented in the thesis provides the most extensive analysis of point-to-point 802.11 interface bonding at the Link layer seen so far in the published research literature. The research is of interest to designers and developers of dynamic interface bonding systems, with or without the assistance of SDN.

Chapter 3 presented an extensive quantitative and qualitative analysis of the inherent slow-down effect for the traditional static approach to bond configuration when there is an in-order delivery assumption and heterogeneous individual link throughput performance. Previous works such as [171] measured the bond throughput at the Transport Layer, with the aim of evaluating the adverse impact of short-term variations in the 802.11 access delay on protocols such as TCP and UDP. However, this work measured the long-term performance of 802.11 bonding at the MAC layer, with a particular focus in the impact of the 802.11 DCF channel access mechanism. Perhaps the most useful finding in Chapter 3 was the identification of a series of problematic applied loads where the aggregate bond throughput was reduced below that of the fastest available slave link, thereby rendering the use of the technique counter productive to the aim of boosting wireless access speeds. Notably, the effect was found to be a commonly occurring due to the condition of the underlying 802.11 MAC and without the instability effects at the Transport Layer as documented by Shidik et al. in the above work.

This was found to be of particular relevance when evaluating SDN-assisted approaches to dynamic interface selection in Chapter 4. For example, the bond throughput gap metric used in the dynamic MPTCP solution by Nam et al. in [130] was shown to be unable to identify the damaging and counter productive throughput scenario. The results presented therefore provides important information for protocol designers seeking to implement efficient dynamic interface bonding systems, and highlights the need for an effective identification and avoidance of the above condition.

5.2 Research Limitations

As per the main thesis research objectives, the presented work provides insight into the impact of the 802.11 MAC on the interface bonding performance, but only the main decision-making algorithms and related performance metrics are accounted, without considering the wider SDN-assisted bonding architectures, programming interfaces, and physical implementations required for a practical deployment. For example, external factors such as the additional control channel latency introduced by the physical control/data plane decoupling are not considered.

Furthermore, the implemented MatLab simulator, while providing great flexibility in terms of the 802.11 technologies used and other test parameters such as the PHY data rate and applied load, uses a simplified bonding mechanism that is limited to traditional round robin bond scheduling, and therefore cannot be extended to accommodate more advanced algorithms such as weighted round robin and minimum round trip time etc. In addition, due to the bond throughput being measured at the MAC layer, the adverse impact of 802.11 bonding on Transport Layer protocol such as TCP and UDP are not considered.

In terms of the individual 802.11 link simulation, the MatLab-based implementation is also fundamentally limited by the analytical modelling used in the validation testing. In many ways, the framework can be considered as a practical implementation of the Bianchi DCF modelling presented in [30], along with all the relevant modelling assumptions and limitations such as infinite re-transmissions attempts, ideal RF conditions, and a flat, fully-connected 802.11 topology without any hidden or exposed nodes present.

5.3 Chapter 2 Summary

The foundation for the main thesis research was provided in Chapter 2, which included all relevant literature surveys, numerical analysis, and software implementation carried with respect to the single-link 802.11 performance evaluation.

Section 2.2.1 provided an overview of the globally predominant standard for WLANs: the IEEE 802.11 protocol. The history and early development of the standard was discussed and all major revisions described individually, with a particular focus on the 802.11 channel access mechanism at the MAC later, known as the DCF.

To provide a reliable method of validating the implemented MatLab DCF simulator, an overview of MAC modelling concepts and literature review of related techniques was given in Section 2.2.2. There was a detailed description of the well established Markov-based saturation modelling provided by Bianchi in Section 2.2.2.3, and the access delay extensions by Chatzimisios in Section 2.2.2.4, which were chosen for the simulation validation.

To provide example output, a numerical evaluation of the Bianchi and Chatzimisios modelling for the 802.11-1997 FHSS PHY and the 802.11b DSSS PHY was provided in Section 2.3.

In order to provide a baseline for the 802.11 interface bonding analysis in Chapter 3 and Chapter 4, in Section 2.3.3 we analysed and compared the MAC layer performance between two notable versions of 802.11: the 5 GHz 11ac and the sub-1 GHz 11ah standards. Despite both Wi-Fi types using the same raw PHY data rate of 65 Mbps with an identical applied load, the achieved channel throughput of the 11ah variant was considerably lower compared to 11ac, which was due to the difference in DCF parameters such as the slot duration and inter-frame spacing lengths between the two standards. The measured difference in the average channel saturation throughput and expected access delay was 23.74% and 67.14% respectively.

The design, implementation, and validation of the developed DCF evaluation framework in MatLab was presented in Section 2.4. The validation test results showed that the implemented DCF simulator was able to model the DCF channel throughput performance with a mean relative accuracy of $\pm 1\%$, and the tagged STA node throughput with a mean relative accuracy of $\pm 2.02\%$, under a wide range of a considered applied loads.

5.4 Chapter 3 Summary

Chapter 3 built on the simulations and research presented in Chapter 3 to analyse the throughput performance of a statically configured, dual 802.11, point-to-point interface bond at Layer 2 of the protocol stack.

Relevant background concepts, design challenges, and research work related to 802.11 interface bonding was presented and discussed in Section 3.2. We saw the difference in approaches taken at all layers of the protocol stack. We also described functional components of a bonding systems, for example the link monitoring unit and interface selector, and the difference between point-to-point and end-to-end interface bonding used at Layer 2 and Layer 4 respectively. An overview of related standards and technologies was also provided, including MPTCP, Linux bonding, and the LACP, as well as a non-exhaustive list of related research work such as the Layer 2 802.11 solution for Linux by the authors So et al in [180], and various early hardware based experimental works such as that presented by Shidik et al. in [171].

Section 3.3 described the design and implementation of the MatLab bonding layer, which was constructed on top of the single-link 802.11 DCF framework validated in the previous chapter. In Sections 3.4 to 3.6, the performance of the traditional static approach to 802.11 link selection was evaluated using the implemented bonding simulator. Under a wide range of different applied load combinations, the long-term bond throughput, delay, and link bandwidth utilisation were measured, and the impact of the inherent slow-down effect for in-order delivery was quantified and qualified. The test results demonstrated the in efficiency of the traditional static configuration for heterogeneous link performance. In particular, on multiple occasions throughput the simulations, the impact of the throughput slow-down was so severe that the aggregate bond throughput was reduced below that of the fastest available individual 802.11 link, thereby rendering the use of the bonding technique counter productive to the aim of increasing the overall throughput. Notably, by measuring the bond throughput at the MAC, this condition was observed without considered the adverse affects on upper layer protocols such as TCP and UDP, as described by Shidik et al. in [171].

In Section 3.6.3, the throughout and bandwidth utilisation performance was analysed and compared between the dual-11ac and heterogeneous 11ac-11ah interface bonds considered in the experiments. The test results showed that the general aggregate throughput performance of the heterogeneous 11ac-11ah bond was significantly lower when compared to that of the homogeneous 11ac-11ac bond, which was in spite of the identical applied loads and 65Mbps raw PHY date used by both types. This was due to the difference in DCF parameters, such as slot duration and DIFS, between the two 802.11 versions. The overall achieved mean dual-11ac bond throughput for static round robin was 6.085 Mbps, compared to just 4.1 Mbps for the heterogeneous 11ac-11ah bond. Furthermore the mean link bandwidth utilisation was also substantially lower at 78.75% compared to 84.62% to the homogeneous bond.

5.5 Chapter 4 Summary

Chapter 4 presented the main thesis research: the investigation of dynamic interface bonding leveraging SDN.

An overview of SDN was given in Section 4.2.1, where the main architectural principles and components were described, for example the SDN controller, network hypervisors, southbound interfaces, and software switches, along with design challenges such as the control/data plane scalability and the additional latency introduced by the relocation of network intelligence. There was also a non-exhaustive literature review of related SDN research work covering subjects such as network slicing and isolation, security, load balancing, and the software-defined WAN.

Section 4.2.2 discussed the application of SDN techniques in wireless networks. In particular, we saw how the vastly expanded control/data plane functionality differed significantly from the original form of SDN, which was designed primarily to speed up provisioning in high large-scale, high performance data centres. We saw how SDN was being leveraged in the wireless domain to assist in functions such as seamless handover and the dynamic allocation of radio resource, in order to address to the co-channel interference problems associated with increasing network density.

The central concepts and design challenges related to the main thesis research theme of SDN-assisted interface bonding were presented in Section 4.2.3. In particular, we saw how the bond functions, for example link monitoring and interface selection, were centralised in the SDN controller which is then able to exploit its global network view in order to dynamically update the bond configuration in response to changes in link state. A review of closely related, SDN-assisted interface bonding research was also provided, with a focus on the main thesis reference work, the architecture and decision-making framework for dynamic link selection in end-to-end MPTCP interface bonding using heterogeneous dual 802.11 links, as presented by the authors Nam et al. in [130].

In Sections 4.3 to 4.5, the contemporary work by Nam et al. was independently evaluated using the custom 802.11 bonding framework implemented in MatLab. Using a variety of different applied load scenarios and combinations, the throughput-gap-based dynamic link selection was shown to outperform the traditional static bond configuration by 26.81% in terms of the mean bond throughput and by 7.28% in terms of mean link bandwidth utilisation. However, the performance was not robust for all applied loads considered, and was shown to be unable to guard against the damaging scenario where the bond throughput is reduced below that of the fastest available single link, therefore rendering the use of bonding counter-productive to aim of increasing wireless access speeds. The scenario was found to be a common event which occurred at various applied loads in 7 out of the 8 tests performed, and was seen on far more occasions for the heterogeneous 11ac-11ah bond compared to the dual-11ac bond.

In Sections 4.6 to 4.8, a novel, predicted-throughput-based dynamic link selection algorithm was proposed that was designed specifically to avoid the above counter productive scenario, while

at least matching the throughput-gap-based Nam algorithm performance at all other times. Comparing the MT algorithm results directly against Nam in all eight individual tests conducted, the overall mean bond throughput was increased by 0.79% and the mean link bandwidth utilisation by 6.16%.

5.6 Concluding Remarks

Towards the long term goal of seamless and highly efficient link aggregation assisted by software defined networking, this thesis investigated wireless interface bonding performance using heterogeneous 802.11 device types. To the best knowledge of the author, the work represents a first in the literature with respect to the detailed and extensive analysis of the 802.11 bond throughput under a large number of applied load combinations and individual achieved link throughputs, as well as the similarly rigorous evaluation of dynamic SDN-assisted link selection algorithms.

The presented research has helped to quantify the potential performance benefit of the SDN-assisted interface technique in 802.11 networks, while also identifying the relevant design challenges and technical limitations of the approach. The thesis also attempted to draw together the related SDN bonding research works published in a coherent manner, which has not yet been attempted in the literature. While there is still some way to go before the above goal is reached, it is hoped that research contained in this thesis may be able to assist future researchers and engineers towards their aims in this regard.

BIBLIOGRAPHY

[1] *packet delay analysis of ieee 802.11 mac protocol*.

[2] *"IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications"*, IEEE Std 802.11-1997, (1997), pp. 1–445.

[3] *"IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band"*, IEEE Std 802.11a-1999, (1999), pp. 1–102.

[4] *"IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band"*, IEEE Std 802.11b-1999, (2000), pp. 1–96.

[5] *"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: Amendment 6: Medium Access Control (MAC) Security Enhancements"*, IEEE Std 802.11i-2004, (2004), pp. 1–190.

[6] *"IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements"*, IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003), (2005), pp. 1–212.

[7] *"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"*, IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999) - Redline, (2007), pp. 1–1238.

[8] *"IEEE Standard for Local and metropolitan area networks–Link Aggregation"*, IEEE Std 802.1AX-2008, (2008), pp. 1–163.

[9] *"IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput"*, IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), (2009), pp. 1–565.

[10] *"IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments"*, IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009), (2010), pp. 1–51.

[11] *"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 Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band"*, IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012), (2012), pp. 1–628.

[12] *"IEEE Standard for Information technology– Telecommunications and information exchange between systemsLocal and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz."*, IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012), (2013), pp. 1–425.

[13] *"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 Amendment 2: Sub 1 GHz License Exempt Operation"*, IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016), (2017), pp. 1–594.

[14] A. A. ABDULLAH, F. GEBALI, AND L. CAI, *Modeling the throughput and delay in wireless multihop ad hoc networks*, in GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference, Nov 2009, pp. 1–6.

[15] M. N. ABOURRAJA, M. OUDANI, M. Y. SAMIRI, J. BOUKACHOUR, A. E. FAZZIKI, A. BOUAIN, AND M. NAJIB, *An improving agent-based engineering strategy for minimizing unproductive situations of cranes in a rail–rail transshipment yard*, SIMULATION, 94 (2018), pp. 681 – 705.

[16] P. AGENEAU AND N. BOUKHATEM, *"Multipath TCP over network coding for wireless networks"*, in 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC), Jan 2017, pp. 373–376.

[17] I. AKYILDIZ, P. WANG, AND S.-C. LIN, *Softair: A software defined networking architecture for 5g wireless systems*, Computer Networks, 85 (2015).

[18] A. M. AL-SADI, A. AL-SHERBAZ, J. XUE, AND S. TURNER, *"Routing algorithm optimization for software defined network WAN"*, in 2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), May 2016, pp. 1–6.

[19] F. AL-TURJMAN, L. MOSTARDA, E. EVER, A. DARWISH, AND N. SHEKH KHALIL, *Network experience scheduling and routing approach for big data transmission in the internet of things*, IEEE Access, PP (2019), pp. 1–1.

[20] S. ALABADY, A. SALAH, M. F. M. F. SALLEH, AND A. HASIB, *"Throughput and Delay Analysis of IEEE 802.11 DCF in the Presence of Hidden Nodes for Multi-hop Wireless Networks"*, Wirel. Pers. Commun., 79 (2014), pp. 907–927.

[21] H. ALI-AHMAD, C. CICCONETTI, A. DE LA OLIVA, M. DRÄXLER, R. GUPTA, V. MANCUSO, L. ROULLET, AND V. SCIANCALEPORE, *Crowd: An sdn approach for densenets*, in 2013 Second European Workshop on Software Defined Networks, 2013, pp. 25–31.

[22] N. ALSBOU AND H. REFAI, *"Two-dimensional Markov model for throughput analysis of IEEE 802.11 Distributed Coordination Function in non ideal channel with capture effects"*, in 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC), Aug 2012, pp. 316–321.

[23] S. ASADOLLAHI, B. GOSWAMI, AND M. SAMEER, *"Ryu controller's scalability experiment on software defined networks"*, in 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), Feb 2018, pp. 1–5.

[24] M. ASLAN AND A. MATRAWY, *"SDN-VSA: Modeling and Analysis of SDN Control Applications Using Vector Spaces"*, in 2018 IEEE International Conference on Communications (ICC), May 2018, pp. 1–6.

[25] E. AU, *Ieee 802.11be: Extremely high throughput [standards]*, IEEE Vehicular Technology Magazine, 14 (2019), pp. 138–140.

[26] M. BANSAL, J. MEHLMAN, S. KATTI, AND P. LEVIS, *Openradio: A programmable wireless dataplane*, HotSDN'12 - Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, (2012).

[27] V. BAÑOS-GONZALEZ, S. AFAQUI, E. LOPEZ-AGUILERA, AND E. GARCIA-VILLEGAS, *Ieee 802.11ah: A technology to face the iot challenge*, Sensors, 16 (2016).

[28] C. BERNARDOS, A. DE LA OLIVA, P. SERRANO, A. BANCHS, L. CONTRERAS, H. JIN, AND J. ZUNIGA, *An architecture for software defined wireless networking*, Wireless Communications, IEEE, 21 (2014), pp. 52–61.

[29] G. BHANAGE, D. VETE, I. SESKAR, AND D. RAYCHAUDHURI, *Splitap: Leveraging wireless network virtualization for flexible sharing of wlans*, 01 2011, pp. 1 – 6.

[30] G. BIANCHI, *"Performance analysis of the IEEE 802.11 distributed coordination function"*, IEEE Journal on Selected Areas in Communications, 18 (2000), pp. 535–547.

[31] G. BIANCHI AND I. TINNIRELLO, *"Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network"*, in IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), vol. 2, March 2003, pp. 844–852 vol.2.

[32] N. BIZANIS AND F. A. KUIPERS, *"SDN and Virtualization Solutions for the Internet of Things: A Survey"*, IEEE Access, 4 (2016), pp. 5591–5606.

[33] J. BOTELHO, *"The LTE-U Vs. WiFi Debate"*, 2019.
[Online]. Available: https://www.networkcomputing.com/wireless-infrastructure/lte-u-vs-wifi-debate [Accessed Jan. 15, 2012].

[34] C. BOURAS, A. KOLLIA, AND A. PAPAZOIS, *"Teaching network security in mobile 5G using ONOS SDN controller"*, in 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), July 2017, pp. 465–470.

[35] M. M. CARVALHO AND J. J. GARCIA-LUNA-ACEVES, *"Delay analysis of IEEE 802.11 in single-hop networks"*, in 11th IEEE International Conference on Network Protocols, 2003. Proceedings., Nov 2003, pp. 146–155.

[36] M. CHAMANIA, T. SZYRKOWIEC, M. SANTUARI, D. SIRACUSA, A. AUTENRIETH, V. LOPEZ, P. SKOLDSTROM, AND S. JUNIQUE, *"Intent-based in-flight service encryption in multi-layer transport networks"*, in 2017 Optical Fiber Communications Conference and Exhibition (OFC), March 2017, pp. 1–2.

[37] Z. CHANG, O. ALANEN, T. HUOVINEN, T. NIHTILA, E. H. ONG, J. KNECKT, AND T. RISTANIEMI, *"Performance Analysis of IEEE 802.11ac DCF with Hidden Nodes"*, in 2012 IEEE 75th Vehicular Technology Conference (VTC Spring), May 2012, pp. 1–5.

[38] P. CHATZIMISIOS, A. C. BOUCOUVALAS, AND V. VITSAS, *"IEEE 802.11 packet delay-a finite retry limit analysis""*, in GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489), vol. 2, Dec 2003.

[39] K. CHEBROLU, R. KAMESWARI, R. BHASKARAN, AND R. RAMESH, *"A Network Layer Approach to Enable TCP over Multiple Interfaces"*, Wireless Networks, 11 (2005), pp. 637–650.

[40] CISCO, *Cisco Annual Internet Report - Cisco Annual Internet Report 2018–2023) White Paper*, 2020.
[Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html [Accessed Jan. 15, 2012].

[41] D. COURCY, K. RASHID, AND D.ERIC, *"If You Like Bonding, You Will Love Teaming"*, 2014.
[Online]. Available: https://rhelblog.redhat.com/tag/team-driver [Accessed Mar. 1, 2019].

[42] J. COX, *Apple iOS 7 surprises as first with new multipath TCP connections*, 2013.
[Online]. Available: https://www.networkworld.com/article/2170068/apple-ios-7-surprises-as-first-with-new-multipath-tcp-connections.html [Accessed Jan. 15, 2012].

[43] J. H. COX, S. DONOVAN, R. J. CLARKY, AND H. L. OWEN, *"Ryuretic: A modular framework for Ryu"*, in MILCOM 2016 - 2016 IEEE Military Communications Conference, Nov 2016, pp. 1065–1070.

[44] N. L. S. DA FONSECA AND R. BOUTABA, *Openflow and SDN for Clouds*, IEEE, 2015, pp. 129–152.

[45] J. DAI AND C. SHEN, *A modified lbt mechanism and performance enhancement for lte-u/wifi co-existence*, in 2017 IEEE/CIC International Conference on Communications in China (ICCC), Oct 2017, pp. 1–6.

[46] T. DAVIES, *"Linux Ethernet Bonding Driver HOWTO"*, 2000.
[Online]. Available: https://www.kernel.org/doc/Documentation/networking/bonding.txt [Accessed Mar. 1, 2019].

[47] A. DE GANTE, M. ASLAN, AND A. MATRAWY, *Smart wireless sensor network management based on software-defined networking*, in 2014 27th Biennial Symposium on Communications (QBSC), 2014, pp. 71–75.

[48] T. DE SCHEPPER, J. STRUYE, E. ZELJKOVIC, S. LATRE, AND J. FAMAEY, *Software-defined multipath-tcp for smart mobile devices*, in 2017 13th International Conference on Network and Service Management (CNSM), Nov 2017, pp. 1–6.

[49] J. DeLisle, *What's the Difference Between IEEE 802.11af and 802.11ah?*, 2015. [Online]. Available: https://www.mwrf.com/technologies/active-components/article/21846205/whats-the-difference-between-ieee-80211af-and-80211ah [Accessed Jan. 15, 2012].

[50] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, *Cloudmac — an openflow based architecture for 802.11 mac layer processing in the cloud*, in 2012 IEEE Globecom Workshops, 2012, pp. 186–191.

[51] D. Deng, K. Chen, and R. Cheng, *"IEEE 802.11ax: Next generation wireless local area networks"*, in 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Aug 2014, pp. 77–82.

[52] B. Dezfouli, V. Esmaeelzadeh, J. Sheth, and M. Radi, *A review of software-defined wlans: Architectures and central control mechanisms*, IEEE Communications Surveys Tutorials, 21 (2019), pp. 431–463.

[53] M. Dilmore, A. Doufexi, and G. Oikonomou, *Analysing interface bonding in 5g wlans*, in 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Sep. 2018, pp. 1–5.

[54] L. F. Dong, Y. T. Shu, H. M. Chen, and M. D. Ma, *"Packet delay analysis on IEEE 802.11 DCF under finite load traffic in multi-hop ad hoc networks"*, Science in China Series F: Information Sciences, 51 (2008), pp. 408–416.

[55] J. Duan, Z. Wang, and C. Wu, *Responsive multipath tcp in sdn-based datacenters*, in 2015 IEEE International Conference on Communications (ICC), June 2015, pp. 5296–5301.

[56] A. Duda, *Understanding the performance of 802.11 networks*, in 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008, pp. 1–6.

[57] M. Durvy, J. Abeillé, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels, *Poster abstract: Making sensor networks ipv6 ready*, (2021).

[58] P. E. Engelstad and O. N. Osterbo, *"Analysis of the Total Delay of IEEE 802.11e EDCA and 802.11 DCF"*, in 2006 IEEE International Conference on Communications, vol. 2, June 2006.

[59] C. Fancy and M. Pushpalatha, *"Performance evaluation of SDN controllers POX and floodlight in mininet emulation environment"*, in 2017 International Conference on Intelligent Sustainable Systems (ICISS), Dec 2017, pp. 695–699.

[60] E. FELEMBAN AND E. EKICI, *"Single Hop IEEE 802.11 DCF Analysis Revisited: Accurate Modeling of Channel Access Delay and Throughput for Saturated and Unsaturated Traffic Cases""*, IEEE Transactions on Wireless Communications, 10 (2011).

[61] J. FLINN, T. GIULI, B. HIGGINS, B. NOBLE, A. REDA, AND D. WATSON, *The case for intentional networking*, 01 2009.

[62] A. B. FLORES, R. E. GUERRA, E. W. KNIGHTLY, P. ECCLESINE, AND S. PANDEY, *"IEEE 802.11af: a standard for TV white space spectrum sharing"*, IEEE Communications Magazine, 51 (2013).

[63] A. FORD, C. RAICIU, M. HANDLEY, AND O. BONAVENTURE, *"TCP Extensions for Multipath Operation with Multiple Addresses"*, RFC 6824, IETF, January 2013.
[Online]. Available: http://www.rfc-editor.org/rfc/rfc6824.txt [Accessed Mar. 1, 2019].

[64] B. FORUM, *"CPE WAN Management Protocol"*, 2019.
[Online]. Available: https://cwmp-data-models.broadband-forum.org/ [Accessed Jan. 15, 2012].

[65] N. FOSTER, M. J. FREEDMAN, R. HARRISON, J. REXFORD, M. L. MEOLA, AND D. WALKER, *"Frenetic: a high-level language for OpenFlow networks"*, in PRESTO '10, 2010.

[66] S. FRANKEL AND S. KRISHNAN, *"IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap"*, RFC 6071, IETF, February 2011.
[Online]. Available: http://www.rfc-editor.org/rfc/rfc6071.txt [Accessed Mar. 1, 2019].

[67] T. FRATCZAK, M. BROADBENT, P. GEORGOPOULOS, AND N. RACE, *Homevisor: Adapting home network environments*, 10 2013, pp. 32–37.

[68] G. BIANCHI AND I. TINNIRELLO, *"Remarks on IEEE 802.11 DCF performance analysis"*, IEEE Communications Letters, 9 (2005), pp. 765–767.

[69] P. L. GALLEGOS-SEGOVIA, J. F. BRAVO-TORRES, P. E. VINTIMILLA-TAPIA, J. O. ORDONEZ-ORDONE, R. E. MORA-HUIRACOCHA, AND V. M. LARIOS-ROSILLO, *"Evaluation of an SDN-WAN controller applied to services hosted in the cloud"*, in 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), Oct 2017, pp. 1–6.

[70] L. GALLUCCIO, S. MILARDO, G. MORABITO, AND S. PALAZZO, *Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks*, in 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 513–521.

[71] M. GARETTO AND C.-F. CHIASSERINI, *Performance analysis of 802.11 wlans under sporadic traffic*, 05 2005, pp. 1343–1347.

[72] M. GARETTO, T. SALONIDIS, AND E. W. KNIGHTLY, *Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks*, IEEE/ACM Transactions on Networking, 16 (2008), pp. 864–877.

[73] P. GAWLOWICZ AND A. ZUBOW, *Demo abstract: Practical cross-technology radio resource management between lte-u and wifi*, in IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), April 2018, pp. 1–2.

[74] P. GORJA AND R. KURAPATI, *"Extending open vSwitch to L4-L7 service aware OpenFlow switch"*, in 2014 IEEE International Advance Computing Conference (IACC), Feb 2014, pp. 343–347.

[75] Y. GOTO, H. MASUYAMA, B. NG, W. K. G. SEAH, AND Y. TAKAHASHI, *"Queueing Analysis of Software Defined Network with Realistic OpenFlow Based Switch Model"*, in 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Sep. 2016, pp. 301–306.

[76] K. GOVINDARAJAN, *"A literature review on Software-Defined Networking (SDN) research topics, challenges and solutions"*, in 2013 Fifth International Conference on Advanced Computing (ICoAC), Dec 2013, pp. 293–299.

[77] A. GUDIPATI, D. PERRY, L. E. LI, AND S. KATTI, *Softran: Software defined radio access network*, in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN '13, New York, NY, USA, 2013, Association for Computing Machinery, p. 25–30.

[78] K. GUO, S. SANADHYA, AND T. WOO, *"ViFi: Virtualizing WLAN Using Commodity Hardware"*, in Proceedings of the 9th ACM Workshop on Mobility in the Evolving Internet Architecture, MobiArch '14, New York, NY, USA, 2014, ACM, pp. 25–30.

[79] K. HABAK, A. HARRAS, AND M. YOUSSEF, *"Bandwidth aggregation techniques in heterogeneous multi-homed devices: A survey"*, Computer Networks, 92 (2015), pp. 168 – 188.

[80] K. HABAK, A. K. HARRAS, AND M. YOUSSEF, *"OPERETTA: An optimal energy efficient bandwidth aggregation system"*, vol. 1, 06 2012, pp. 121–129.

[81] K. HABAK, K. HARRAS, AND M. YOUSSEF, *"OSCAR: A Collaborative Bandwidth Aggregation System"*, in "Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS '14)", 01 2014.

[82] K. HABAK, M. YOUSSEF, AND K. A. HARRAS, *"G-DBAS: A green and deployable bandwidth aggregation system"*, in 2012 IEEE Wireless Communications and Networking Conference (WCNC), April 2012, pp. 3290–3295.

[83] B. HAFAIEDH, I. E. KORBI, L. A. SAIDANE, AND A. KOBBANE, *"LTE-U and WiFi coexistence in the 5 GHz unlicensed spectrum: A survey""*, in 2017 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), Nov 2017, pp. 1–7.

[84] E. HALEPLIDIS, D. JOACHIMPILLAI, J. H. SALIM, D. LOPEZ, J. MARTIN, K. PENTIKOUSIS, S. DENAZIS, AND O. KOUFOPAVLOU, *"ForCES Applicability to SDN-Enhanced NFV"*, in 2014 Third European Workshop on Software Defined Networks, Sep. 2014, pp. 43–48.

[85] J. HAMILTON, *"Datacenter Networks are in my Way"*, 2010.
[Online]. Available: https://perspectives.mvdirona.com/2010/10/datacenter-networks-are-in-my-way/ [Accessed Jan. 15, 2012],.

[86] Y. HAN, J. LI, D. HOANG, J. YOO, AND J. W. HONG, *"An intent-based network virtualization platform for SDN"*, in 2016 12th International Conference on Network and Service Management (CNSM), Oct 2016, pp. 353–358.

[87] I. T. HAQUE AND N. ABU-GHAZALEH, *Wireless software defined networking: A survey and taxonomy*, IEEE Communications Surveys Tutorials, 18 (2016), pp. 2713–2737.

[88] K. D. HUANG, K. R. DUFFY, D. MALONE, AND D. J. LEITH, *"Investigating the validity of IEEE 802.11 MAC modeling hypotheses"*, in 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, Sept 2008, pp. 1–6.

[89] E. HUSNI AND A. BRAMANTYO, *"Design and Implementation of MPLS SDN Controller Application based on OpenDaylight"*, in 2018 International Symposium on Networks, Computers and Communications (ISNCC), June 2018, pp. 1–5.

[90] J. HWANG AND J. YOO, *"Packet scheduling for Multipath TCP"*, in 2015 Seventh International Conference on Ubiquitous and Future Networks, July 2015, pp. 177–179.

[91] Y. JARRAYA, T. MADI, AND M. DEBBABI, *"A Survey and a Layered Taxonomy of Software-Defined Networking"*, IEEE Communications Surveys Tutorials, 16 (2014), pp. 1955–1980.

[92] A. JAYASURIYA, S. AUST, P. DAVIS, A. YAMAGUCHI, AND S. OBANA, *Aggregation of wi-fi links: When does it work?*, 12 2007, pp. 318 – 323.

[93] B. JIN, S. KIM, D. YUN, H. LEE, W. KIM, AND Y. YI, *"Aggregating LTE and WiFi: Toward Intra-Cell Fairness and High TCP Performance"*, IEEE Transactions on Wireless Communications, 16 (2017), pp. 6295–6308.

[94] X. JIN, L. LI, L. VANBEVER, AND J. REXFORD, *Softcell: Scalable and flexible cellular core network architecture*, 12 2013, pp. 163–174.

[95] D. C. K. I. C. J. A. K. F. H. J. Y. L. L. E. K. K. MENG, A. JONES, *Ieee 802.11 real time applications tig report*, tech. rep., IEEE, 2018.

[96] M. KANG, E. KANG, D. HWANG, B. KIM, K. NAM, M. SHIN, AND J. CHOI, *"Formal Modeling and Verification of SDN-OpenFlow"*, in 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, March 2013, pp. 481–482.

[97] S. KAUR, J. SINGH, AND N. GHUMMAN, *Network programmability using pox controller*, 08 2014.

[98] S. KENT AND K. SEO, *"Security Architecture for the Internet Protocol"*, RFC 4301, IETF, December 2005.
[Online]. Available: http://www.rfc-editor.org/rfc/rfc4301.txt [Accessed Mar. 1, 2019].

[99] M. KETEL, *"Enhancing BYOD Security Through SDN"*, in SoutheastCon 2018, April 2018, pp. 1–2.

[100] R. KHALAF, I. RUBIN, AND J. HSU, *Throughput and delay analysis of multihop ieee 802.11 networks with capture*, in 2007 IEEE International Conference on Communications, June 2007, pp. 3787–3792.

[101] E. KHOROV, I. LEVITSKY, AND I. F. AKYILDIZ, *Current status and directions of ieee 802.11be, the future wi-fi 7*, IEEE Access, 8 (2020), pp. 88664–88688.

[102] T. KIM, S. CHOI, J. MYUNG, AND C. LIM, *"Load balancing on distributed datastore in opendaylight SDN controller cluster"*, in 2017 IEEE Conference on Network Softwarization (NetSoft), July 2017, pp. 1–3.

[103] R. KOKKU, R. MAHINDRA, H. ZHANG, AND S. RANGARAJAN, *"CellSlice: Cellular wireless resource slicing for active RAN sharing"*, in 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), Jan 2013, pp. 1–10.

[104] D. KREUTZ, F. M. V. RAMOS, P. E. VERISSIMO, C. E. ROTHENBERG, S. AZODOLMOLKY, AND S. UHLIG, *"Software-Defined Networking: A Comprehensive Survey"*, Proceedings of the IEEE, 103 (2015), pp. 14–76.

[105] B. KRISHNA, V. SATHYA, AND B. R. TAMMA, *"A Dynamic Link aggregation Scheme for heterogeneous wireless networks"*, in 2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Jan 2014, pp. 1–6.

[106] P. KRONGBARAMEE AND Y. SOMCHIT, *"Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch"*, in 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), July 2018, pp. 1–5.

[107] P. KUMAR AND A. KRISHNAN, *"Throughput Analysis of the IEEE 802.11 Distributed Coordination Function Considering Capture Effects"*, in 2010 3rd International Conference on Emerging Trends in Engineering and Technology, 2010.

[108] A. LARA, B. RAMAMURTHY, E. POUYOUL, AND I. MONGA, *"WAN virtualization and dynamic end-to-end bandwidth provisioning using SDN"*, in 2015 Optical Fiber Communications Conference and Exhibition (OFC), March 2015, pp. 1–3.

[109] D. H. LEE, *"Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks"*, ICML 2013 Workshop : Challenges in Representation Learning (WREPL), (2013).

[110] L. E. LI, Z. M. MAO, AND J. REXFORD, *Toward software-defined cellular networks*, in 2012 European Workshop on Software Defined Networking, 2012, pp. 7–12.

[111] X. LIANG AND X. QIU, *"A software defined security architecture for SDN-based 5G network"*, in 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), Sep. 2016, pp. 17–21.

[112] L. LIAO, A. SHAMI, AND V. C. M. LEUNG, *"Distributed FlowVisor: a distributed FlowVisor platform for quality of service aware cloud network virtualisation"*, IET Networks, 4 (2015), pp. 270–277.

[113] H. LIN, L. SUN, Y. FAN, AND S. GUO, *Apply embedded openflow mpls technology on wireless openflow — openroads*, in 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012, pp. 916–919.

[114] F. LIU AND Z. YANG, *"Design of VMware vSphere Automatic Operation and Maintenance System Based on Python"*, in 2018 International Conference on Advanced Mechatronic Systems (ICAMechS), Aug 2018, pp. 283–286.

[115] J. LIU, S. ZHANG, N. KATO, H. UJIKAWA, AND K. SUZUKI, *Device-to-device communications for enhancing quality of experience in software defined multi-tier lte-a networks*, IEEE Network, 29 (2015), pp. 46–52.

[116] Y. LIU, W. WU, B. WANG, T. HE, S. YI, AND Y. XIA, *Measurement-based channel management in wlans*, in 2010 IEEE Wireless Communication and Networking Conference, 2010, pp. 1–6.

[117] D. LOPEZ-PEREZ, A. GARCIA-RODRIGUEZ, L. GALATI-GIORDANO, M. KASSLIN, AND K. DOPPLER, *Ieee 802.11be extremely high throughput: The next generation of wi-fi technology beyond 802.11ax*, IEEE Communications Magazine, 57 (2019), pp. 113–119.

[118] T. LUO, H. TAN, AND T. Q. S. QUEK, *Sensor openflow: Enabling software-defined wireless sensor networks*, IEEE Communications Letters, 16 (2012), pp. 1896–1899.

[119] D. MALONE, I. DANGERFIELD, AND D. LEITH, *Verification of common 802.11 mac model assumptions*, vol. 4427, 06 2007, pp. 63–72.

[120] S. MIURA, T. HANAWA, T. YONEMOTO, T. BOKU, AND M. SATO, *"RI2N/DRV: Multi-link ethernet for high-bandwidth and fault-tolerant network on PC clusters"*, in 2009 IEEE International Symposium on Parallel Distributed Processing, May 2009, pp. 1–7.

[121] N. MOHAMED AND J. AL-JAROODI, *"Self-Configuration Techniques for MuniSocket"*, in Third IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC 2007), Sep. 2007, pp. 11–20.

[122] S. V. MORZHOV AND M. A. NIKITINSKIY, *"Development and research of the PreFirewall network application for floodlight SDN controller"*, in 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), March 2018, pp. 1–4.

[123] H. MOURA, A. ALVES, J. BORGES, D. MACEDO, AND M. VIEIRA, *Ethanol: A software-defined wireless networking architecture for ieee 802.11 networks*, Computer Communications, 149 (2019).

[124] A. S. A. MUBARAK, H. ESMAIEL, AND E. M. MOHAMED, *New capwap architectures for ieee 802.11ad based wi-fi/wigig wlans*, in 2018 International Conference on Innovative Trends in Computer Engineering (ITCE), 2018, pp. 231–235.

[125] R. MURTY, J. PADHYE, R. CHANDRA, A. WOLMAN, AND B. ZILL, *Designing high performance enterprise wi-fi networks.*, 01 2008, pp. 73–88.

[126] R. MURTY, J. PADHYE, A. WOLMAN, AND M. WELSH, *Dyson: An architecture for extensible wireless lans*, in Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, USENIXATC'10, USA, 2010, USENIX Association, p. 15.

[127] R. MURTY, A. WOLMAN, AND M. WELSH, *An architecture for extensible wireless lans*, Proceedings of ACM HotNetsVII, (2008).

[128] K. NAKAUCHI AND Y. SHOJI, *Wifi network virtualization to control the connectivity of a target service*, IEEE Transactions on Network and Service Management, 12 (2015), pp. 308–319.

[129] K. NAKAUCHI, Y. SHOJI, AND N. NISHINAGA, *Airtime-based resource control in wireless lans for wireless network virtualization*, in 2012 Fourth International Conference on Ubiquitous and Future Networks (ICUFN), 2012, pp. 166–169.

[130] H. NAM, D. CALIN, AND H. SCHULZRINNE, *"Towards dynamic MPTCP Path control using SDN"*, in 2016 IEEE NetSoft Conference and Workshops (NetSoft), June 2016, pp. 286–294.

[131] H. NAM, K. KIM, J. Y. KIM, AND H. SCHULZRINNE, *Towards qoe-aware video streaming using sdn*, in 2014 IEEE Global Communications Conference, 2014, pp. 1317–1322.

[132] O. NARMANLIOGLU AND E. ZEYDAN, *"Learning in SDN-based multi-tenant cellular networks: A game-theoretic perspective"*, in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), May 2017, pp. 929–934.

[133] S. NASTIC, H. L. TRUONG, AND S. DUSTDAR, *"SDG-Pro: a programming framework for software-defined IoT cloud gateways"*, Journal of Internet Services and Applications, 6 (2015), p. 21.

[134] A. A. NEGHABI, N. JAFARI NAVIMIPOUR, M. HOSSEINZADEH, AND A. REZAEE, *"Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature"*, IEEE Access, 6 (2018), pp. 14159–14178.

[135] M. NETWORKS, *"Broadband Bonding Service"*, 2019.
[Online]. Available: https://www.mushroomnetworks.com/broadband-bonding-service [Accessed Mar. 1, 2019].

[136] K. NGUYEN, K. ISHIZU, AND F. KOJIMA, *"An evolvable, scalable, and resilient control channel for software defined wireless access networks"*, Computers and Electrical Engineering, 57 (2017), pp. 104 – 117.

[137] T. A. NGUYEN, T. EOM, S. AN, J. S. PARK, J. B. HONG, AND D. S. KIM, *"Availability Modeling and Analysis for Software Defined Networks"*, in 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC), Nov 2015, pp. 159–168.

[138] OFCOM, *"TV White Spaces: Approach to Coexistence"*, 2019.
[Online]. Available: https://www.ofcom.org.uk/consultations-and-statements/category-1/white-space-coexistence [Accessed Jan. 15, 2012].

[139] B. OLIVEIRA AND C. MARGI, *Distributed control plane architecture for software-defined wireless sensor networks*, 09 2016, pp. 85–86.

[140] A. PAL AND A. NASIPURI, *"Performance analysis of IEEE 802.11 distributed coordination function in presence of hidden stations under non-saturated conditions with*

infinite buffer in radio-over-fiber wireless LANs", in 2011 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN), Oct 2011, pp. 1–6.

[141] S. PARK, C. JOO, Y. PARK, AND S. BAHK, *"Minimizing Application-Level Delay of Multipath TCP in Wireless Networks: A Receiver-Centric Approach"*, in 2015 IEEE 23rd International Conference on Network Protocols (ICNP), Nov 2015, pp. 245–255.

[142] A. PATRO AND S. BANERJEE, *Coap: A software-defined approach for home wlan management through an open api*, SIGMOBILE Mob. Comput. Commun. Rev, 18 (2014).

[143] PEPLINK, *"SpeedFusion Bonding Technology"*, 2019.
[Online]. Available: https://www.peplink.com/technology/speedfusion-bonding-technology [Accessed Mar. 1, 2019].

[144] I. PETROV AND O. MORGUNOVA, *"Forwarding Rule Minimization for Network Statistics Analysis in SDN"*, in 2018 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC), Oct 2018, pp. 1–6.

[145] M. PETTY, *Calculating and using confidence intervals for model validation*, Fall Simulation Interoperability Workshop 2012, 2012 Fall SIW, (2012), pp. 37–45.

[146] D. PFAMMATTER, D. GIUSTINIANO, AND V. LENDERS, *A software-defined sensor architecture for large-scale wideband spectrum monitoring*, 04 2015, pp. 71–82.

[147] Y. QIAN, W. YOU, AND K. QIAN, *"FlowVisor vulnerability analysis"*, in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), May 2017, pp. 867–868.

[148] P. RAGHAV AND A. DUA, *"Enhancing flow security in Ryu controller through set operations"*, in 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Dec 2017, pp. 1265–1269.

[149] R. RAHMANI, H. RAHMAN, AND T. KANTER, *On performance of logical-clustering of flow-sensors*, 2014.

[150] A. L. RAMABOLI, L. FALOWO, E. OLABISI, AND A. H. CHAN, *"Review: Bandwidth Aggregation in Heterogeneous Wireless Networks: A Survey of Current Approaches and Issues"*, J. Netw. Comput. Appl., 35 (2012), pp. 1674–1690.

[151] A. RAO, M. VISALI, S. SHAILENDRA, B. PANIGRAHI, AND A. SIMHA, *"Reliable robotic communication using multi-path TCP"*, in 2017 9th International Conference on Communication Systems and Networks (COMSNETS), Jan 2017, pp. 429–430.

[152] S. M. RAZA, D. PARK, Y. PARK, K. LEE, AND H. CHOO, *"Dynamic Load Balancing of Local Mobility Anchors in Software Defined Networking Based Proxy Mobile IPv6"*, in Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, IMCOM '16, New York, NY, USA, 2016, ACM, pp. 106:1–106:4.

[153] J. REICH, C. MONSANTO, N. FOSTER, J. REXFORD, AND D. WALKER, *"Modular SDN programming with Pyretic"*, USENIX Login, 38 (2013), pp. 128–134.

[154] M. RICHART, J. BALIOSIAN, J. SERRAT, AND J. GORRICHO, *"Resource Slicing in Virtual Wireless Networks: A Survey"*, IEEE Transactions on Network and Service Management, 13 (2016), pp. 462–476.

[155] F. RICO, P. FONSECA, AND A. SOUSA, *"A testbed for developing, simulating and experimenting multipath aggregation algorithms"*, in Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Sept 2014, pp. 1–4.

[156] R. RIGGIO, *The empower mobile network operating system: demo*, 10 2016, pp. 87–88.

[157] R. ROCHA, M. A. M. VIEIRA, AND A. A. F. ANTONIO, *"Dynamic Link Aggregation in Software Defined Networking"*, in 2018 IEEE Symposium on Computers and Communications (ISCC), June 2018, pp. 615–620.

[158] E. ROZNER, Y. MEHTA, A. AKELLA, AND L. QIU, *Traffic-aware channel assignment in enterprise wireless lans*, in 2007 IEEE International Conference on Network Protocols, 2007, pp. 133–143.

[159] A. SADASIVARAO, S. SYED, P. PAN, C. LIOU, I. MONGA, C. GUOK, AND A. LAKE, *"Bursting Data between Data Centers: Case for Transport SDN"*, in 2013 IEEE 21st Annual Symposium on High-Performance Interconnects, Aug 2013, pp. 87–90.

[160] P. SAN SEGUNDO, *A new dsatur-based algorithm for exact vertex coloring*, Computers Operations Research, 39 (2012), pp. 1724 – 1733.

[161] S. T. SANY, S. SHASHIDHAR, M. P. GILESH, AND S. D. M. KUMAR, *"Performance evaluation and assessment of FlowVisor"*, in 2016 International Conference on Information Science (ICIS), Aug 2016, pp. 222–227.

[162] M. S. SAUD, H. CHOWDHURY, AND M. KATZ, *"Heterogeneous Software-Defined Networks: Implementation of a Hybrid Radio-Optical Wireless Network"*, in 2017 IEEE Wireless Communications and Networking Conference (WCNC), March 2017, pp. 1–6.

[163] J. SCHULZ-ZANDER, C. MAYER, B. CIOBOTARU, S. SCHMID, AND A. FELDMANN, *Opensdwn: programmatic control over home and enterprise wifi*, Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, (2015).

[164] J. SCHULZ-ZANDER, N. SARRAR, AND S. SCHMID, *Aeroflux: A near-sighted controller architecture for software-defined wireless networks*, in ONS, 2014.

[165] M. SEDDIKI, M. SHAHBAZ, S. DONOVAN, S. GROVER, M. PARK, N. FEAMSTER, AND Y. SONG, *Flowqos: Qos for the rest of us*, HotSDN 2014 - Proceedings of the ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking, (2014).

[166] M. SEYEDEBRAHIMI, F. BOUHAFS, A. RASCHELLÀ, M. MACKAY, AND Q. SHI, *Sdn-based channel assignment algorithm for interference management in dense wi-fi networks*, in 2016 European Conference on Networks and Communications (EuCNC), 2016, pp. 128–132.

[167] M. J. SHAMANI, W. ZHU, S. REZAIE, AND V. NAGHSHIN, *"Signal aware multi-path TCP"*, in 2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Jan 2016, pp. 1–4.

[168] G. SHARMA, A. GANESH, AND P. KEY, *"Performance Analysis of Contention Based Medium Access Control Protocols"*, IEEE Transactions on Information Theory, 55 (2009), pp. 1665–1682.

[169] K. SHARMA, *"An alleviated model for private cloud deployment using VMware"*, in 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), Aug 2017, pp. 1–3.

[170] S. SHENKER, *The Future of Networking and the Past of Protocols*, 2011.
[Online]. Available: https://www.youtube.com/watch?v=YHeyuD89n1 [Accessed Jan. 15, 2012].

[171] G. SHIDIK, N. MUHAMAD, AND A. Z. AZRI, *"Performance Analysis of Interface Bonding and Channel Bonding at Point to Point"*, Latin America Transactions, IEEE (Revista IEEE America Latina), 11 (2013), pp. 633–635.

[172] M. S. SIDDIQUI, E. ESCALONA, E. TROUVA, M. A. KOURTIS, D. KRITHARIDIS, K. KATSAROS, S. SPIROU, C. CANALES, AND M. LORENZO, *"Policy based virtualised security architecture for SDN/NFV enabled 5G access networks"*, in 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Nov 2016, pp. 44–49.

[173] D. SIMEONIDOU, R. NEJABATI, AND S. AZODOLMOLKY, *"Enabling the future optical Internet with OpenFlow: A paradigm shift in providing intelligent optical network services"*, in 2011 13th International Conference on Transparent Optical Networks, 2011, pp. 1–4.

[174] W. SIMPSON, *"The Point to Point Protocol (PPP)"*, Aug 1996.
[Online]. Available: hhttps://tools.ietf.org/html/rfc1990 [Accessed Mar. 1, 2019].

[175] P. SIRIPONGWUTIKORN, *Throughput analysis of an ieee 802.1lb multihop ad hoc network*, in TENCON 2006 - 2006 IEEE Region 10 Conference, Nov 2006, pp. 1–4.

[176] V. A. SIRIS, G. STAMATAKIS, AND E. TRAGOS, *A simple end-to-end throughput model for 802.11 multi-radio multi-rate wireless mesh networks*, IEEE Communications Letters, 15 (2011), pp. 635–637.

[177] V. SIVARAMAN, T. MOORS, H. HABIBI GHARAKHEILI, D. ONG, J. MATTHEWS, AND C. RUSSELL, *Virtualizing the access network via open apis*, in Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT '13, New York, NY, USA, 2013, Association for Computing Machinery, p. 31–42.

[178] K. SKLOWER, B. LLOYD, G. MCGREGOR, D. CARR, AND T. CARADETTI, *"The PPP Multilink Protocol (MP)"*, Aug 1996.
[Online]. Available: hhttps://tools.ietf.org/html/rfc1990 [Accessed Mar. 1, 2019].

[179] J. SMITH, J. WOODHAMS, AND R. MARG, *Controller-based wireless lan fundamentals: An end-to-end reference guide to design, deploy, manage, and secure 802.11 wireless networks*, 2010.

[180] J. SO, J. PARK, AND A. ELIAB, *"Wireless Interface Bonding Supporting In-Order Delivery and Automatic Load Balancing"*, IEEE Transactions on Vehicular Technology, 66 (2017), pp. 5190–5203.

[181] C. SONG, *Performance analysis of the ieee 802.11p multichannel mac protocol in vehicular ad hoc networks*, in Sensors, 2017.

[182] T. SONG AND T. KIM, *Performance analysis of synchronous multi-radio multi-link mac protocols in ieee 802.11be extremely high throughput wlans*, Applied Sciences, 11 (2021).

[183] W. SONG AND W. ZHUANG, *"Performance Analysis of Probabilistic Multipath Transmission of Video Streaming Traffic over Multi-Radio Wireless Devices"*, IEEE Transactions on Wireless Communications, 11 (2012), pp. 1554–1564.

[184] K. SOOD, S. YU, AND Y. XIANG, *"Performance Analysis of Software-Defined Network Switch Using M/Geo/1 Model"*, IEEE Communications Letters, 20 (2016), pp. 2522–2525.

[185] R. SOULE, S. BASU, P. J. MARANDI, F. PEDONE, R. KLEINBERG, E. G. SIRER, AND N. FOSTER, *"Merlin: A Language for Managing Network Resources"*, IEEE/ACM Transactions on Networking, 26 (2018), pp. 2188–220.

[186] M. G. S. SRIYANANDA, I. PARVEZ, I. GÜVENE, M. BENNIS, AND A. I. SARWAT, *Multi-armed bandit for lte-u and wifi coexistence in unlicensed bands*, in 2016 IEEE Wireless Communications and Networking Conference, April 2016, pp. 1–6.

[187] R. STEWART, *"Stream Control Transmission Protocol (SCTP)"*, Sep 2007. [Online]. Available: hhttps://tools.ietf.org/html/rfc4960 [Accessed Mar. 1, 2019].

[188] Y. SU, X. DU, L. HUANG, Z. GAO, AND M. GUIZANI, *"LTE-U and Wi-Fi Coexistence Algorithm Based on Q-Learning in Multi-Channel"*, IEEE Access, 6 (2018), pp. 13644–13652.

[189] T. SUBRAMANYA, R. RIGGIO, AND T. RASHEED, *"Intent-based mobile backhauling for 5G networks"*, in 2016 12th International Conference on Network and Service Management (CNSM), Oct 2016, pp. 348–352.

[190] Q. SUN, Y. XUE, S. LI, AND Z. ZHU, *"Design and Demonstration of High-Throughput Protocol Oblivious Packet Forwarding to Support Software-Defined Vehicular Networks"*, IEEE Access, 5 (2017), pp. 24004–24011.

[191] L. SURESH, J. SCHULZ-ZANDER, R. MERZ, A. FELDMANN, AND T. VAZAO, *Towards programmable enterprise wlans with odin*, HotSDN'12 - Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, (2012).

[192] T. SZYRKOWIEC, M. SANTUARI, M. CHAMANIA, D. SIRACUSA, A. AUTENRIETH, V. LOPEZ, J. CHO, AND W. KELLERER, *"Automatic intent-based secure service creation through a multilayer SDN network orchestration"*, IEEE/OSA Journal of Optical Communications and Networking, 10 (2018), pp. 289–297.

[193] T. TAKIGUCHI, H. MASUI, A. HIDAKA, Y. SUGIZAKI, O. MIZUNO, AND K. ASATANI, *A new application level link aggregation and its implementation on android terminals*, in 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC), 2012, pp. 8–12.

[194] A. S. TANENBAUM AND D. J. WETHERALL, *Computer Networks*, Prentice Hall Press, Upper Saddle River, NJ, USA, 5th ed., 2010.

[195] M. TANHA, D. SAJJADI, R. RUBY, AND J. PAN, *"Capacity-Aware and Delay-Guaranteed Resilient Controller Placement for Software-Defined WANs"*, IEEE Transactions on Network and Service Management, 15 (2018), pp. 991–1005.

[196] J. W. TANTRA, C. H. FOH, AND A. B. MNAOUER, *"Throughput and delay analysis of the IEEE 802.11e EDCA saturation"*, in IEEE International Conference on Communications, 2005. ICC 2005. 2005, vol. 5, May 2005, pp. 3450–3454 Vol. 5.

[197] D. TATANG, F. QUINKERT, J. FRANK, C. ROPKE, AND T. HOLZ, *"SDN-Guard: Protecting SDN controllers against SDN rootkits"*, in 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Nov 2017, pp. 297–302.

[198] P. THUBERT, M. R. PALATTELLA, AND T. ENGEL, *"6TiSCH centralized scheduling: When SDN meet IoT"*, in 2015 IEEE Conference on Standards for Communications and Networking (CSCN), Oct 2015, pp. 42–47.

[199] O. TICKOO AND B. SIKDAR, *"Queueing analysis and delay mitigation in IEEE 802.11 random access MAC based wireless networks"*, in IEEE INFOCOM 2004, vol. 2, March 2004, pp. 1404–1413 vol.2.

[200] I. TINNIRELLO, *"Kalman filter estimation of the contention dynamics in error-prone IEEE 802.11 networks"*, in 2008 3rd International Symposium on Communications, Control and Signal Processing, March 2008, pp. 671–676.

[201] I. TRAJKOVSKA, P. AESCHLIMANN, C. MARTI, T. M. BOHNERT, AND J. SALVACHÚA, *Sdn enabled qos provision for online streaming services in residential isp networks*, in 2014 IEEE International Conference on Consumer Electronics - Taiwan, 2014, pp. 33–34.

[202] C. TSELIOS, I. POLITIS, AND S. KOTSOPOULOS, *"Enhancing SDN security for IoT-related deployments through blockchain"*, in 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Nov 2017, pp. 303–308.

[203] N. VAN TU, J. HYUN, AND J. W. HONG, *"Towards ONOS-based SDN monitoring using in-band network telemetry"*, in 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Sep. 2017, pp. 76–81.

[204] J. S. VARDAKAS, I. PAPAPANAGIOTOU, M. D. LOGOTHETIS, AND S. A. KOTSOPOULOS, *"On the End-to-End Delay Analysis of the IEEE 802.11 Distributed Coordination Function"*, in Second International Conference on Internet Monitoring and Protection (ICIMP 2007), July 2007, pp. 16–16.

[205] L. VDOVIN, P. LIKIN, AND A. VILCHINSKII, *"Network utilization optimizer for SD-WAN"*, in 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), Oct 2014, pp. 1–4.

[206] VIPRINET, *"Rugged VPN"*, 2019.
[Online]. Available: https://www.viprinet.com/en/technology/ruggedvpn [Accessed Mar. 1, 2019].

[207] A. VOELLMY AND P. HUDAK, *"Nettle: Taking the Sting out of Programming Network Routers"*, in Proceedings of the 13th International Conference on Practical Aspects of

Declarative Languages, PADL'11, Berlin, Heidelberg, 2011, Springer-Verlag, pp. 235–249.

[208] H. L. VU AND T. SAKURAI, *"Accurate delay distribution for IEEE 802.11 DCF"*, IEEE Communications Letters, 10 (2006), pp. 317–319.

[209] K. WANG, T. DREIBHOLZ, X. ZHOU, F. FA, Y. TAN, X. CHENG, AND Q. TAN, *"On the Path Management of Multi-path TCP in Internet Scenarios Based on the NorNet Testbed"*, in 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), March 2017, pp. 1–8.

[210] Q. WANG, K. JAFFRES-RUNSER, J. SCHARBARG, C. FRABOUL, Y. SUN, J. LI, AND Z. LI, *"A thorough analysis of the performance of delay distribution models for IEEE 802.11 DCF"*, Ad Hoc Networks, 24 (2015), pp. 21 – 33.
Modeling and Performance Evaluation of Wireless Ad-Hoc Networks.

[211] S. WANG, X. WU, H. CHEN, Y. WANG, AND D. LI, *An optimal slicing strategy for sdn based smart home network*, in 2014 International Conference on Smart Computing, 2014, pp. 118–122.

[212] Z. WANG, H. M. SHAWKAT, S. ZHAO, AND B. SHEN, *"An LTE-U coexistence scheme based on cognitive channel switching and adaptive muting strategy"*, in 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Oct 2017, pp. 1–6.

[213] P. XIAO, W. QU, H. QI, Z. LI, AND Y. XU, *"The SDN controller placement problem for WAN"*, in 2014 IEEE/CIC International Conference on Communications in China (ICCC), Oct 2014, pp. 220–224.

[214] J. XIE, F. R. YU, T. HUANG, R. XIE, J. LIU, C. WANG, AND Y. LIU, *"A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges"*, IEEE Communications Surveys Tutorials, 21 (2019), pp. 393–430.

[215] C. XU, W. JIN, G. ZHAO, H. TIANFIELD, S. YU, AND Y. QU, *"A Novel Multipath-Transmission Supported Software Defined Wireless Network Architecture"*, IEEE Access, 5 (2017), pp. 2111–2125.

[216] D. XU, T. SAKURAI, AND H. L. VU, *"An Access Delay Model for IEEE 802.11e EDCA"*, IEEE Transactions on Mobile Computing, 8 (2009), pp. 261–275.

[217] M. YAN, J. CASEY, P. SHOME, A. SPRINTSON, AND A. SUTTON, *Ætherflow: Principled wireless support in sdn*, (2015).

[218] Q. YAN, F. R. YU, Q. GONG, AND J. LI, *"Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges"*, IEEE Communications Surveys Tutorials, 18 (2016), pp. 602–622.

[219] M. YANG, B. LI, Z. YAN, AND Y. YAN, *Ap coordination and full-duplex enabled multi-band operation for the next generation wlan: Ieee 802.11be (eht)*, in 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), 2019, pp. 1–7.

[220] V. YAZICI, U. C. KOZAT, AND M. O. SUNAY, *A new control plane for 5g network architecture with a case study on unified handoff, mobility, and routing management*, IEEE Communications Magazine, 52 (2014), pp. 76–85.

[221] Y. YIAKOUMIS, M. BANSAL, A. COVINGTON, J. VAN REIJENDAM, S. KATTI, AND N. MCKEOWN, *Behop: A testbed for dense wifi networks*, in Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WiNTECH '14, New York, NY, USA, 2014, Association for Computing Machinery, p. 1–8.

[222] Y. YIAKOUMIS, K.-K. YAP, S. KATTI, G. PARULKAR, AND N. MCKEOWN, *Slicing home networks*, (2011).

[223] J. YU, X. WANG, J. SONG, Y. ZHENG, AND H. SONG, *"Forwarding Programming in Protocol-Oblivious Instruction Set"*, in 2014 IEEE 22nd International Conference on Network Protocols, Oct 2014, pp. 577–582.

[224] Y. ZAKI, C. GOERG, AND A. TIMM-GIEL, *"LTE wireless virtualization and spectrum management"*, in WMNC2010, Oct 2010, pp. 1–6.

[225] S. ZANNETTOU, M. SIRIVIANOS, AND F. PAPADOPOULOS, *"Exploiting path diversity in datacenters using MPTCP-aware SDN"*, in 2016 IEEE Symposium on Computers and Communication (ISCC), June 2016, pp. 539–546.

[226] M. ZEESHAN, A. ALI, A. NAVEED, A. LIU, A. WANG, AND H. QURESHI, *Modeling packet loss probability and busy time in multi-hop wireless networks*, EURASIP Journal on Wireless Communications and Networking, 2016 (2016).

[227] S. ZEHL, A. ZUBOW, M. DOERING, AND A. WOLISZ, *Resfi: A secure framework for self organized radio resource management in residential wifi networks*, 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), (2016), pp. 1–11.

[228] P. ZERFOS, G. ZHONG, J. CHENG, H. LUO, S. LU, AND J. J. LI, *Dirac: a software-based wireless router system*, in MobiCom '03, 2003.

[229] H. ZHAI, Y. KWON, AND Y. FANG, *Performance analysis of ieee 802.11 mac protocols in wireless lans*, Wireless Communications and Mobile Computing, 4 (2004), pp. 917–931.

[230] H. ZHANG, S. VRZIC, G. SENARATH, N. D. DÀO, H. FARMANBAR, J. RAO, C. PENG, AND H. ZHUANG, *5g wireless network: Mynet and sonac*, IEEE Network, 29 (2015), pp. 14–23.

[231] D. ZHAO, M. ZHU, AND M. XU, *Sdwlan: A flexible architecture of enterprise wlan for client-unaware fast ap handoff*, in Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1–6.

[232] H. ZHONG, Y. FANG, AND J. CUI, *"LBBSRT: An efficient SDN load balancing scheme based on server response time"*, Future Generation Computer Systems, 68 (2017), pp. 183 – 190.

[233] F. ZHOU, T. DREIBHOLZ, X. ZHOU, F. FU, Y. TAN, AND Q. GAN, *"The Performance Impact of Buffer Sizes for Multi-path TCP in Internet Setups"*, in 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), March 2017, pp. 9–16.

[234] P. ZHOU, K. CHENG, X. HAN, X. FANG, Y. FANG, R. HE, Y. LONG, AND Y. LIU, *"IEEE 802.11ay-Based mmWave WLANs: Design Challenges and Solutions"*, IEEE Communications Surveys Tutorials, 20 (2018), pp. 1654–1681.

[235] A. ZUBOW, S. ZEHL, AND A. WOLISZ, *Bigap — seamless handover in high performance enterprise ieee 802.11 networks*, in NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 445–453.