This electronic thesis or dissertation has been downloaded from Explore Bristol Research, http://research-information.bristol.ac.uk

Author:
Alferaidi, Khaled D

Title:
Multi-hop Wireless Networks with Network Coding Techniques

General rights
Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

Take down policy
Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact collections-metadata@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

Multi-hop Wireless Networks with Network Coding Techniques.

By

KHALED DHAIFALLAH M. ALFERAIDI

Department of Electric and Electronic Engineering
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.

SEPTEMBER 2018

Word count: 40324

Abstract

The aim of this thesis is to investigate network coding techniques on network-stack components such as Links and Transport Layers. As networks are required to provide reliability and efficiency in their performances in a cost effective manner, the capability of such network in achieving such objective is becoming highly significant in the relevant literature. Moreover, proving reliability and delivery of fast services in any networks settings is very challenging due to the nature of wireless links, mobility, collision and frequent topology changes. Nevertheless, a Multi-Hop network possesses specifics advantages such as easy deployment, extended coverage, and low implementation costs compared to other approaches. Therefore, this research sets out to explore the effective and practical use of Xor network coding in Multi-Hop wireless networks to achieve improved throughput and reduce latency in networks by investigating MAC, Transport and proposed Cross-layers in the presence of Xor network coding in Multi-Hop wireless networks. The findings of this thesis provide that one problem with the reliance of Wireless Multi-Hop networks on IEEE 802.11 technology is the inefficiency of Distributed Coordination Function (DCF) applied in IEEE 802.11 settings, leaving a room to employ a Network Coding technique to improve performance. DCF is found fair to nodes than carrying out Network Coding. Hence, applied a new approach based on DCF mechanism on MAC layer has provided more access to the medium, and also gave a better share of medium for relay nodes which enabled bidirectional traffic flows in saturated multi-hop network traffic's scenario. The proposed scheme has provided a maximum of 255% throughput improvement compared to COPE with the MAC layer priority over ten hops. Moreover, moving the focus from throughput to Ratio of Loss Packet (RLP), delay and jitters are found to be important factors as they are capable of explaining the impact on throughput and ratio of lost packets. Hence, the thesis has proposed a Transport layer-based scheme called WeNC-TCP. This scheme found to enable the use of network coding functionality, and permit better interactions between TCP at Transport layer and network coding functionality. The scheme was found to provide a better ratio of loss packets, and "end-to-end" Delays have been achieved. The average ratio of loss packets has improved up to 72%, and end-to-end Delays have reduced to up to 48%. Furthermore, TCP is found to provide better network congestion management in a wired network. However, in wireless networks, TCP is found not to work well when Xor network coding is applied. This is due to the absence of the MAC layer's role to assist TCP packet flow. Therefore, a cross-layer interaction solution is investigated. The simulated results have shown performance improvement concerning throughputs, delays, and jitter. The scheme has provided 100% RLP reduction compared to other TCP variants. The proposed scheme has also provided a minimum delay of 93%, and the jitters have also been reduced to the average of 42%.

Dedication and acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Robert Piechocki, for his guidance, encouragement, trust and enthusiasm throughout my exciting journey as PhD student. His invaluable advice and support has helped me become not only a better researcher but also a better person, and his belief in my abilities for research will undoubtedly continue to inspire my future career. During my studies in the Communication Systems and Networks Research Group at the University of Bristol I was fortunate to collaborate with many distinguished people, all of whom have influenced my work in many ways. I would like to take this opportunity to sincerely thank them for their help over the last few years.

Furthermore, I am truly grateful to all of my friends, who are too many to be named here, for making my stay in Bristol one of the most joyful and exciting periods of my life. My wife, Kopi, was a great support and a best friend who was around to pick me up afters hiccups during my PhD journey. Finally, I wish to thank my mother, Naja, and father, Daifallah, and my sisters and brothers, Khalood, Fasial, Dina, Mohammed and Ahmed. They have been my deep motivation to carry on in my journey, and deeply believed in my decision to pursue my dreams.

Copyrights and Publication

Attention is drawn to the fact that the copyright of this thesis rests with the author. This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author. This thesis may be made available for consultation within the University Library and may not be photocopied or lent to other libraries for the purpose of consultation.

List of Publication

Attention is drawn to the fact that the copyright of this thesis rests with the author. This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author. This thesis may be made available for consultation within the University Library and may not be photocopied or lent to other libraries for the purpose of consultation.

List of Publications

Conference publications

- K. Alferaidi and R. Piechocki, "Improving network coding throughput in multi-hop networks through DCF adoption in 802.11", 2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS), Bangkok, Thailand, 2017, pp. 1 –6 doi: 10.1109/ICETSS.2017.8324180.
- Alferaidi, Khaled, and Robert Piechocki. "TCP-MAC Cross Layer Integration for Xor Network Coding." In Science and Information Conference, pp. 860-875. Springer, Cham, 2018.
- K. Alferaidi, R. Piechocki and F. Alfordy, "Improving TCP Performance in Multi-Hop Coded Wireless Networks," 2019 2nd International Conference on Computer Applications and Information Security (ICCAIS), Riyadh, Saudi Arabia, 2019, pp. 1 –6 doi: 10.1109/CAIS.2019.8769518.

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: ....

List of Abbreviation

MAC MAC Media Access Control

CW Contention Window in DCF function at MAC layer

Alice and Bob Three nodes with Two hops away

DCF Distributed Coordination Function

RTT Round Trip Time

COPE Novel packet-level network coding technique

COPE with MAC priority, MAC-P packet-level network coding technique using MAC layer functionality

RLP The Ratio of Loss Packet

NS3 Network Simulator 3

TCP Transmission Control Protocol

CWND Congestion Window Size

MAX Window Maximum Window Size

Advertised Window Receiver's Window Size

AIMD Additive Increase/Multiplicative Decrease

ACK,Ack Acknowledgment message (Packet)

RTO Retransmission Timeout Timer

MSS Maximum segment size

TCP/NC Transmission Control Protocol/Network Coding

SACK Selective Acknowledgment

DupAcks Duplicate Acknowledgment (Three or Two)

BW Bandwidth

WeNC-TCP Westwood Network Coding-TCP

MACINT-TCP MAC Interaction-TCP

ssThresh Slow-Start Threshold

T Throughput

PLR Packet Lost Ratio

AODV Ad hoc On-demand Distance Vector

TABLE OF CONTENTS

List of Tables

List of Figures

Figure **Page**

Chapter 1

Introduction

Over the last decade, wireless technologies have shown tremendous growth. To meet increasing users demands, more and more applications have been invented. Users will need wireless connectivity to use these wireless technologies. Effective utilisation of network resources is, therefore, a prime requirement. Effective use of resources lessens the number of transmissions required, and reduces power consumption. Ahlswede et al. [2] defined network coding which intelligently codes packets at nodes, this might increase network throughput.

The code in [2] could be understood as the Max-flow Min-cut Theorem for network coding. The maximum amount of data flow from the source to the the destination (Max-flow) is equal to the minimum capacity. This capacity will not allow any flow to go throw from the source to the destination when the cut is applied in the network (Min-cut). If intermediate nodes are allowed to perform coding, then the maximum multicast rate will be equal to the minimum min-cut from the source to every destination. Moreover, if all recipients have the same min-cut from the source, then the network coding will allow all nodes to achieve the min-cut capacity synchronously. The capacity is related to the maximum flow rate, which is the same at each receiver in the network.

To appreciate the importance of network coding in a network, it is essential to compare network coding with the traditional routing paradigm. Network communication was designed on the principal of the store-and-forward model. In this model, information is seen as a commodity that relies on routing as a means of delivery in the network. It is achieved through replication at the intermediate nodes. As we are entered the new millennium, the concept of network coding fundamentally transformed the way a network can be utilised and operated. For example, the information can be exploited and merged in the network for better optimisation regarding transmission. Let's consider the wireless network in figure 1.1. The network is lined up in a

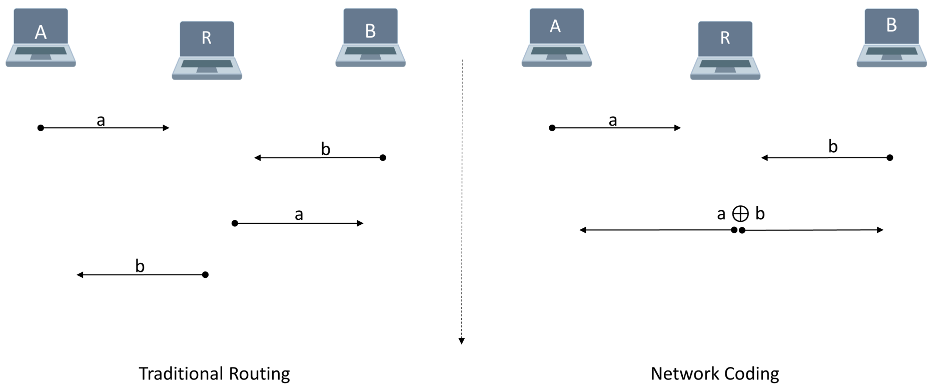


FIGURE 1.1. Traditional Routing vs. Network Coding.

chain topology. Nodes A and B are sending each other data through node R using the traditional routing principal of store-and-forward. Nodes A and B are interested in exchanging packets.

The number of necessary transmissions is four. Node A sends packet a1 to node R, the intermediate node that will store it and look into its routing table to find a route to node B. Meanwhile, node B has sent a packet to node R. The intermediate node will store both packets and decide to forward them to each end. Therefore, the intermediate node will need two transmissions: the first transmission to node B contains packet a1, and the one to node A carries packet b1. In network coding, only three transmissions are required. For example, the last transmission is saved by combining both packets and sending them during the third transmission. Nodes A and B can obtain the information by performing the same operation on the combined information.

1.1 Dawn of Network Coding for Wireless Networks

The emergence of network coding begins when Ahlswede expressed [2] a new class of problems called network information flow which was inspired by computer network applications (Multicast). Their result regarded as s the Max-flow Min-cut Theorem for network information flow. In addition, the work of Ahlswede [2] revealed that we should not regard the information to be multicast as a "fluid" which could simply be routed or replicated. But, by employing coding at the nodes, which it could in general save the bandwidth. Subsequently, this finding has had a significant impact on future networks.

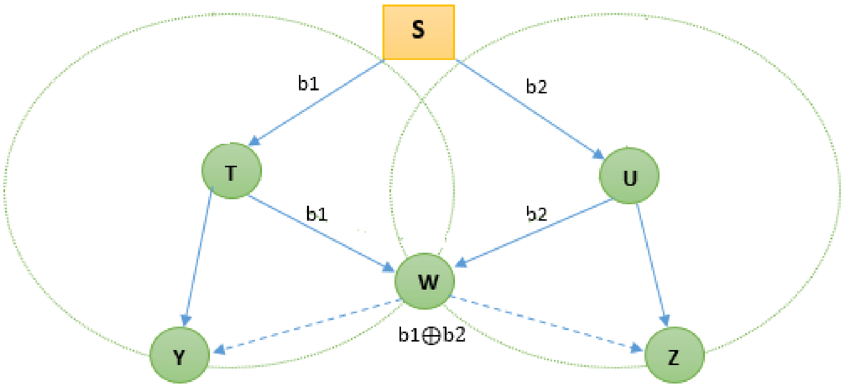


FIGURE 1.2. Xor Wireless Overhears Topology.

1.2 Xor Network Coding

The network coding technique permits the intermediate (relay) nodes to generate new packets by combining the received packets on their incoming edges. This approach offers manifold benefits, such as considerable increase in throughput, solid reliability and robustness in the network as the authors describe in [3], [2]. For example, in figure 1.2, Node S wants to send information to destination nodes (Y, Z). We assume that all edges of the network are one unit of capacity, in other words, each node can transmit one packet per time unit. S sends one packet per time unit to its next neighbour when packets arrive at node W, and the received packets are Xored $(b1 \oplus b2)$ and resent to the next nodes, to be disseminated again to their destinations (Y, Z). The intermediate node can maximise coding gain by making N packets combinations if all recipients already have $N-1$ packets of the same combination. Note that network coding using Xor operation depends on network topology, and overhearing; for example, nodes Z and Y can hear transmissions of U and T for packet $b2$ and $b1$.

1.3 Random Linear Network Coding (RLNC)

The random linear network coding approach adopts a linear combination of incoming information symbolised as an element over a finite field. The main difference to the previous approach is the replacement of exclusive OR operation with a linear combination of data (in terms of matrix multiplication) where coefficients of linear combination are chosen from a finite field equal to $GF(2^q)$. The advantage of using RLNC is the flexibility of mixing packets. It does not depend on receiving a specific packet but on having the sufficient number of independent (native) packets. Having the required number of native packets will lead to successful decoding.

1.3.0.1 Encoding

To understand the process of RLNC, let us say that $m_1, m_2, m_3 ... m_n$ are source packets that need to be broadcast to one or more receivers. In a classical transmission scheme, $m_1, m_2, m_3 ... m_n$ will be delivered to the intended destination using relay nodes by the simple store-forward approach. However, in network coding, it is combined by relay nodes between source and destination; after mixing packets, a sequence of linear combination vectors is then forwarded to the destination. Assuming that each packet has contents of b bits, it is important to note that some packets may be different sizes. The small packet will be padded with trailing 0s to round up all packets to have the same (b) size. A source packet, m_i, can be encoded as a vector $\underline{m}_i$ in finite Galois field $GF(2^q)$. Moreover, the packet size b bits and its vector representation will hold $[b/q]$ elements – each of them made up of q bits. As a result, output packets $\underline{y}$ are the linear combination of source packets. Multiplication and addition are possible operations over the finite field $GF(2^q)$. The encoded packets are presented in the following equation:

$$\underline{y} = \sum_{j=1}^{n} g_j \cdot \underline{m}_j \qquad (1.1.1)$$

The combination coefficients are grouped to create vector $\underline{g} = [g_1, g_2, g_3, ... g_n]$ which is then call global encoding vector. Destination nodes need $\underline{g}$ to be able to decode $\underline{y}$: without $\underline{g}$, the destination node will not be able to decode encoded information (as we will see later in the decoding process). The process of encoding can be recursive. For instance, the encoded packet received by the relay node between source and destination can re-encode packets with other previously or recently received packets. The new encoded packets will be forwarded to the next. For instance, node i has received and stored in its queue a set of encoded packets $C = \{\underline{c}_1, \underline{c}_2, \underline{c}_3, ... \underline{c}_m\}$ and their corresponding global encoding vectors $G = \left\{\underline{g}_1, \underline{g}_2, \underline{g}_3, ... \underline{g}_m\right\}$. Then a new set of globing encoding coefficients (such as $\xi_1, \xi_2, .. \xi_m$), generated by relay node, computes a new linear combination for new encoding packets $\underline{z}$,where $\underline{z} = \sum_{j=1}^{m} \xi_j \cdot \underline{c}_j$.

1.3.0.2 Decoding

The receiving node wishes to retrieve the information within received packets and has to solve a system of m equation with n unknown. Thus, node i needs $m \geq n$ to solve the system of received packets that must have be least as large as the possible number of source packets generated at sending nodes. The first condition must be (a) the number of received encoded packets is at least as large as the number of sources packets ($m \geq n$); and (b) the encoded packets must be linearly independent to one another. To attain these conditions, nodes must choose the coefficients uniformly and at random over the finite field of $GF(2^q)$. In order to have efficient network coding, the encode packets should be at least as low as the possible dependent packets. Otherwise, it will lead to the probability of highly unsuccessful decoding. The probability of successful decoding depends on the propriety of the Galois field size [4].

The works in [5], [6] show that this probability becomes insignificant even for small field sizes such as q = 8. Moreover, RLNC allows nodes to operate in a completely independent and decentralised manner, which is appropriate for wireless networks. Therefore, in this thesis, RLNC will henceforth be used and referred to as Network Coding (NC).

When an encoded packet is received, it is added to the last row of the decoding matrix (a triangular matrix, as a result of Gaussian's elimination algorithm). When an encoded packet is added to the decoding matrix, it increases the rank of the matrix (called an innovative packet). If the decoding matrix, at some time, has a row, e.g. $e_j = [0,...,1,...,0]$, where 1 is located at the jth position, the node realises that the source packet m_j affiliated to the jth position can now be recovered. Applying the Gaussian elimination algorithm, it can exist as a sub-matrix, which forms an upper triangular matrix. Such case will lead to the early decoding of some of the encoded packets. This can happen before the decoding matrix has attained full rank.

Limiting the decoding matrices is important. It will help to reduce the complexity of solving a linear system equation of n unknowns. Thus, encoded packets should be grouped into generations and put together in the same category [5]. The size of the Galois field has to be in accordance to generation size, and it results in less memory usage and less computational complexity than with Xor network coding. Xor network coding is a special case of random network coding where the size of the Galois field is equal to 2.

1.4 Benefits of Network Coding

Network coding is an elegant and novel scheme, which was introduced by authors in [2]. Network coding is a vital part of information theory. It can ameliorate network throughput and improve overall performance. Authors in [7] predicted that network coding would be a critical technology for networks in near future. Network coding is capable of enhancing throughput, wireless resources, security, robustness, network storage, complexity and reliability. Furthermore, network coding can enhance performance of battery life, bandwidth and delay in wireless networks. In different circumstances, there is a real challenge facing network coding: it depends upon the role of relay nodes. Such a role is necessary to perform actions on the data packets. Thus, implementing network coding in various networks will require changes and additions to the existing network OSI model. Introducing a network coding layer will not harm the integrity of data packets, and the underlying or upper layers will not need any drastic modification in OSI architecture, or to existed equipment and deployment of software [7]. It is essential to address the role of intermediate nodes, as they are vital to network coding techniques because their role is not just to forward data packets but also to encode (mix) received data packets from different links, as noted by the authors in [8], [9], [10], [11], [7]. Since network coding was formulated in [2], it has been subject to a great deal of development by scholars. It started with mathematical abstraction to real-time implementations with various network technologies. All works in literature follow the same principle: maximising a network's layers functionality and smartly exploiting available resources and operations. Authors in [7] describe this move as a result of the need to ease the burden on wireless resources at the edge node, and that is true, as we will see in the coming chapters.

The emergence of network coding has highlighted the importance of content distribution, expanding heterogeneity in networks, and better handling of data between core and edge nodes in networks. Thus, authors in [7] suggest we should rethink and revisit the way networks are designed. In their discussion in [7], they describe myths that researchers can encounter and encourage the 'busting' of 'facts' about throughput gain, COPE [12] and coding potential. For example, the throughput gain of a factor of 2 is the maximum that can be achieved in wireless networks. In fact, the authors in [7] disagree: there can be a much larger throughput gain. Their explanation was based on the misinterpretation of the work in [12]. Katabi and Katti demonstrated that for undirected networks, the maximum network coding gain by a factor of 2. This was an explanation of maximum throughput gain in an undirected network. However, the authors in [12] were discussing different types of undirected networks. In addition, Li and Li [13] describe networks that are "graph-theoretically" as undirected. This means that links could operate in two directions in a time-shared style per link [7]. But authors in [7] argue that wireless networks are not undirected in that way. Wireless networks can transmit in two directions in a time-shared style per node. As a result, network coding can achieve higher throughout gains in wireless networks.

The first attempt to bridge theory with practice was COPE [12], and it was implemented to addresses the issue of unicast traffic, bursty flows, and to integrate network coding in network stacks. The scholars in [7] argue that there are some myths related to the first implementation as COPE [12]. One of these myths is that COPE in [12] could eliminate congestion. In fact, COPE can ease the congestion and could take advantage of congestion to encode, or relieve the pressure on some of the network's resources. Furthermore, creating congestion to find and create coding opportunities causes more harm to throughput than what is benefitted from congestion. In addition, waiting for packets to be coded for the creation coding opportunities cannot help to improve throughput, as discussed in Chapter 3. Another suggestion made by authors in [7], is not to create coding opportunities at all costs. Instead, network engineers should judiciously choose when and at which nodes to find coding possibilities.

FIGURE 1.3. Chain topology;Two flows in reverse directions.

1.5 Research Aim and Motivation

This thesis will explore the effective and practical use of Xor network coding in Multi-Hop wireless networks. It will specifically focus on incorporating network coding-based frameworks for wireless Multi-Hop networks like AdHoc. The thesis's main goal is to achieve improved throughput and reduce latency in such networks with the help of Xor network coding. The thesis will look into Chain topology with Two or more flows in reverse directions. Figure 1.3 depicts the topology will consider in this thesis. The chain topology will be more than ten hops away, with each node will be located on the edge of each other. Each node will be able to hear the transmission of the next hop. The traffics will be generated by edge nodes. Moreover, this thesis will consider both TCP and UDP traffic packets.

1.5.1 Research Problem

1. The default Contention Window (CW) calculations cause a long waiting time. Consequently, low throughputs are achieved for a contending node. It is clear that there is an unexplained issue with the number of encoded packets, where there is full utilisation of medium or higher throughput. From this point, there was no priority to nodes to gain more access than edge nodes. Contention window (CW) for relay node had higher values and long waiting time that affected the total throughput. On the other hand, the edge nodes had similar CW value, which was large CW values. Therefore, CW had the negative impact on edge nodes that it constrained edge nodes access to the medium. Hence, CW doubles intensively at relay nodes and among other nodes in networks causing drains in overall throughput and long delays.

2. TCP function of the AIMD (Additive Increase/Multiplicative Decrease), had a significant impact on the benefits of network coding. Specifically, the AIMD mechanism is incredibly sensitive to packet losses and translates them as signs of congestion. Furthermore, upon detecting a loss, TCP halves its sending rate by reducing its congestion window size to half. In early simulation test, it was clear that there was AIMD impact on Xor coding network as in COPE [12]. Moreover, the TCP's AIMD has not been designed to accept interaction of intermediate nodes' roles of network coding; it was suitable for classical transmission paradigm. Again, TCP reacts to all such packet losses by reducing its transmission rate to half, and that results in low throughput and average utilisations. However, Xor net-

work coding does not benefit from TCP's high reliability and throughput performances. Meanwhile, the perspective of network coding has a more rigid approach that depends on relay nodes to perform coding rather than depending on senders' and receivers' exchange messages, which delegate network condition on current transmission.

3. Authors in [12] argued that TCP did not show any significant improvement regarding throughput (the average gain was 2-3%). Network congestion and coding opportunities increase, leading to higher good-put gains but it is not as satisfactory as with UDP. The previous explanation was lacking the impact of MAC layer on TCP. In addition, work in [12] has not investigated MAC impact when MAC priority is considered at centred node (bottleneck nodes).

1.5.2 Research Objectives

1. To explore how to maximise throughput through DCFs contention window propriety when Xor network coding is used in a Multi-Hop scenario.

2. To investigate how to improve throughput and reliability of TCP protocols in Multi-Hop scenarios via Xor network coding.

3. To evaluate the impact of MAC-layer vitality, TCP and MAC roles to improve overall performance in Xor network coding in wireless Multi-Hop networks.

1.6 Thesis contribution

The main contribution of this work is network coding throughput improvement using MAC-layer DCF. A new MAC layer contention window calculation technique that can maximise the throughput in Multi-Hop wireless networks is proposed. Some of the results and findings of this work has been presented in the IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS), in Bangkok Thailand. The complete framework of proposed scheme and the numerical results are presented in Chapter 3. The shortcomings of the current MAC-layer contention window in the DCF function in Multi-Hop wireless networks are also presented in Chapter 3. Previous works in Xor MAC layer were discussed and compared with the proposed contention window for Xor network coding at the MAC layer in Multi-Hop wireless networks.

The second contribution is TCP throughput and reliability improvement in Xor network coding networks. This work was presented and published in IEEE 2nd International Conference on Computer Applications, and Information Security ICCAIS' 2019, in Riyadh. Besides, our work illustrated TCP's components in wireless Multi-Hop networks. WeNC-TCP is used to overcome shortcomings in some TCP variant implementations, such as Tahoe, New Reno, Westwood and Westwood+. Chapter 4 presents the whole detailed work on the TCP layer. The proposed WeNC-TCP contributes to the improvement of TCP reliability without the interaction of other layers.

The WeNC-TCP is accompanied by the numerical results compared with other current TCP variants.

The third contribution of this work is TCP traffic management in Xor network coding wireless networks. This is based on the cross-layer integration solution between the TCP and MAC layers. The proposed MACINT-TCP is presented in chapter 5. The proposed scheme has been Simulated, and the results are discussed in more details. The work was published in Science and Information Conference, tilted Computing 2018, which was sponsored technically (TCS) by IEEE, it took place in Holiday Inn London, Regent Park.

1.7 Thesis Outline

The thesis is organised as follows:

1. Chapter 1: The introduction. A brief introduction to network coding, containing the research aim, problem and objectives. Moreover, this chapter introduces the background of network coding. It describes the philosophical aspects of network coding and proposes new notions that make network coding more applicable to practical networks.

2. Chapter 2: The literature review. Contains discussion of both Xor and random linear network coding. This chapter covers detailed works from the literature on MAC and transport layers related to network coding. Cross-layer approaches are also covered.

3. Chapter 3: Improving network coding throughput via MAC-layer DCF. It presents a new MAC layer contention window calculation scheme, which maximises throughput in Multi-Hop wireless networks. It starts with the shortcomings of the MAC-layer contention window in the DCF function in Multi-Hop wireless networks. Moreover, previous works in network coding at the MAC layer are classified and compared with the results of the proposed new MAC layer's contention window calculation scheme.

4. Chapter 4: Improving TCP throughput and reliability in network coding networks. This chapter shows how TCP is validated. In addition, it examines TCP's components in wireless Multi-Hop networks. Moreover, WeNC-TCP is introduced to overcome shortcomings in some TCP variant implementations, such as Tahoe, New Reno, Westwood and Westwood+. Finally, this chapter proposes a WeNC-TCP detailed implementation accompanied with the numerical results of simulation works.

5. Chapter 5: Introducing a cross-layer integration between the TCP and MAC layer. This chapter illustrates the need to bridge the gap found in integration between layers and proposes that an interaction can assist TCP traffic in Xor network coding. Hence, the proposed MACINT-TCP is presented, and simulation results are discussed in detail.

6. Chapter 6: The concluding chapter. This chapter summarises the main findings in chapters 3, 4 and 5. Finally, it suggests new directions for future works.

Chapter 2

Literature Review

In the communication context, a large number of networks around the globe use complex structures, for instance, wired telephone networks, ad-hoc networks, mobile networks, current internet and internet-based peer-to-peer networks. Current research focuses on the objective of information transfer in terms of throughput, reliability and robustness [14]. Network coding was seen as a technique to merge digital messages from different sources to receivers. Long ago, routers and switches carried on forwarding messages using the traditional transmission paradigm. The central ideology of network coding is mixing packet intelligibly to maximise throughput. The current switches and routers are equipped with coders and decoders to fulfil this ideology [14]. In recent years, much effort has been made to recognise the analytical gains of network coding in practical ways [2], [8], [15]. COPE [12] is considered the first implementation that made network coding a popular application in unicast sessions in Multi-Hop wireless networks.

2.1 Deterministic (Xor) Network Coding

Deterministic (Xor) network coding is part of network coding-aware routing, as classified by authors in [14]. The network coding-aware protocols form their decisions based on available coding opportunities across multiple links in wireless networks. Also, the network coding-aware routing approach has several advantages over traditional routing 'store-and-forward' techniques. It advances soaring throughput, high-reaching reliability and lower delays in wireless networks. In this category of approaches, COPE [12] and its variants will be discussed.

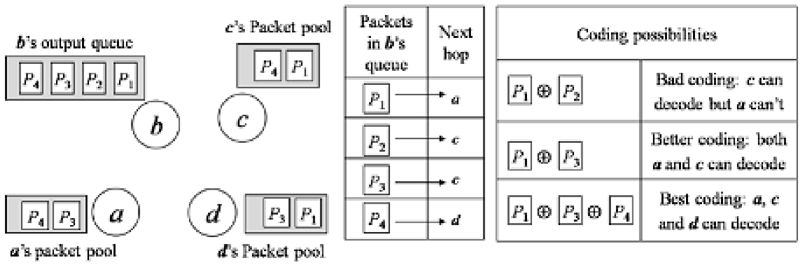


FIGURE 2.1. COPE Coding Process. Figure taken from [12].

2.1.1 COPE

COPE is the first proposed architecture for network coding in a wireless mesh network [12]. COPE applies network coding for broadcast, unicast and multicast flows. It permits nodes to mix many received packets from different neighbours into one encoded packet. However, the intermediate node handles only native packets. Therefore, they have to decode all the packets they receive. The use of the term 'deterministic' refers to a coding process where the node chooses which packets to combine through probability. It is used when packets are combined; for example, if the probability is greater than a certain threshold (0.8), the node sends a combined packet. Otherwise, it sends a native packet. In order to calculate the probability, the node needs to exchange messages with other nodes using the reception report. COPE appends a header between the frame and the IP headers which allows the node to give the identifiers of the packets mixed in the data field. If the receiver recovers native packets, it sends back to the sender node with a reception report that indicates the native packets contained in its packets pool. Figure 2.1 illustrates that a node can have any possibility of combinations, for example: node b has to send $P1$, $P2$,$P3$ and $P4$ to nodes a, b, c and d in this order. Node b realises that each of other nodes a,c,d holds in its packets pool ($P3$,$P4$), ($P1$,$P4$) and ($P1$,$P3$) [12].

2.1.1.1 Coding Process in COPE

Node b has many choices to combine the packets, but it has to choose the possibility that maximises the number of packets in a single transmission. The best choice is $P1 \oplus P3 \oplus P4$. This combination helps the three nodes to decode and recover the native packets, as in figure 2.2. The implementation of COPE in an ad-hoc network proves that throughput could increase by a factor of $\frac{4}{3}$= 1.33 (resp. 1.33) UDP flows. However, COPE [12] has a downside: it is a curbed coding structure, to two hops only. For instance, COPE will not be able to find coding opportunities when passing throughout multiple hops. COPE could use an opportunistic overhearing function, as it hops away from the central area. Therefore, there are few chances to encode flows bidirectionally.

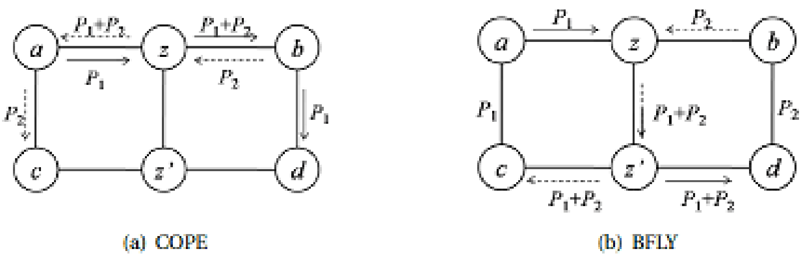


FIGURE 2.2. BFLY and COPE Coding Schemes. Figure taken from [16].

2.1.2 BFLY

When COPE is in use, each intermediate node decodes the received packets before storing and transmitting. Therefore, there are no combined packets directly forwarded to the destination. Based on this, BFLY in [16] is an improvement of COPE[12], which was proposed by scholars in [16]. It has been observed that BFLY could improve the gain in a butterfly network if BFLY allows the forwarding of a combined packet without decoding any of the packets. Figure 2.2 shows the difference between COPE and BFLY: a node a wants to send packet $P1$ to node d, and node b wants to send $P2$ to node c, COPE needs one more transmission to complete the process compared to BFLY. It is to be recalled that node z broadcasts the packet $P1 \oplus P2$ through the wireless links $z \rightarrow a$ and $z \rightarrow b$ and this is considered as one transmission. The authors in [16] suggest combining BFLY and COPE using a probabilistic model to differentiate the maximum throughput. However, the throughput depends on the probability condition of the links that are used. Thus, the node estimates whether BFLY network coding is worthy enough to send based on the link probability, otherwise, it uses the COPE [12] scheme.

2.1.3 DCAR

DCAR [17] extends the COPE scheme by considering two source nodes which are more designated than two hops away, and there is a node on the way to the destination. The figure 2.3 illustrates such a network. As authors in [17] explained, during the routing stage, the source node must inform its neighbours. The intermediate nodes then check if nodes on the way are involved in flows that participate in the DCAR scheme. The coding process is illustrated in figure 2.3. Let us consider two flows $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ and $5 \rightarrow 3 \rightarrow 6 \rightarrow 7$. Node 3 is in the middle of the network,it encodes packets from two flows and transmits to both node 6 and 4. On the other hand, node 6 cannot perform opportunistic listening for decoding. It can only pass the encoded packet to node 7, which can perform the opportunistic listening and decoding. Therefore, the ability to find coding opportunities is maximised, and this results in better bandwidth efficiency and throughput [17].

DCAR is proposed to overcome COPE [12] limitations, such as:

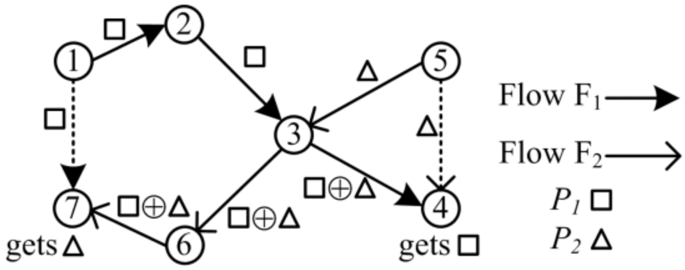


FIGURE 2.3. DCAR Coding Process. Figure taken from [17].

1. the coding opportunity was heavily dependent on the established routes; and

2. the coding structure was restricted to a two-hop zone.

DCAR is a distributed coding-aware routing mechanism which enables:

1. the discovery of possible paths between a source and destination;

2. the identification of potential network coding opportunities in a larger network scope.

Additionally, DCAR is capable of discovering high throughput paths with coding opportunities, compared to the conventional wireless network routing protocols. Also, DCAR can detect coding opportunities on the entire path, which removes the limitation of the two-hop coding in COPE. A novel routing metric called the coding-aware routing metric (CRM) is proposed; which accommodates the performance comparison between coding-possible and coding-impossible paths.

2.1.4 ROCX

The authors of [18] improved the network coding gain by considering coding opportunities during the routing phase. The routing with opportunistically coded exchanges (ROCX) algorithm used in mesh networks applies linear coding while minimising the overall number of transmissions in the network. The use of linear coding has been seen by researchers to be a practical approach when the paths to the destination are prolonged. However, it has complexities and downsides. For instance, linear coding cannot be applied in a network that has low computation capabilities. Moreover, it is not scalable because the linear coding complexity increases exponentially according to the number of nodes in the network. Another problem is that each node has to know which traffic is passing through which link in the network, but knowing all passing traffic is impractical. Moreover, ROCX is based on two hops only, and it is clear that there is no benefit of opportunistic listening in the ROCX scheme. In COPE[12], opportunistic listening is proven to be useful in order to find more coding opportunities and increase throughput. Meanwhile, ROCX considers only the packets that are transmitted to it, but not other packets can be heard in the network.

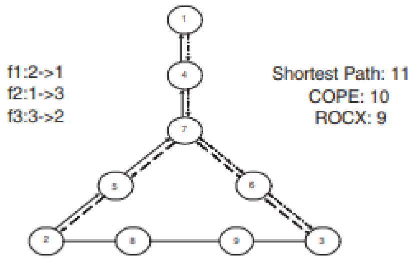


FIGURE 2.4. Figure taken from [18].

To understand the ROCX coding process [18], let us consider figure 2.4, with three flows $f_1 : 2 \rightarrow 1, f_2 : 1 \rightarrow 3, f_3 : 3 \rightarrow 2$. If all paths were efficient and with no losses, the shortest paths in traditional routing are $2 \rightarrow 5 \rightarrow 7 \rightarrow 4 \rightarrow 1, 1 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 3$, $3 \rightarrow 9 \rightarrow 8 \rightarrow 2$. Without coding, the overall number of transmissions necessary to deliver one packet is 11. In COPE [12], nodes 4, 5 and 6 are not able to code therefore, there is only one coding opportunity at node 4, as node 1 and node 7 exchange packets through node 4, and as a result the overall number of transmission is 10. In ROCX [18], if the path of f3 was diverted to $3 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 2$, the overall transmission number would be reduced to nine.

However, ROCX has its limitations, and it does not aid with bandwidth limitations and interference impact. As authors in [19] claimed, if interference was not considered in a wireless network, flow needed a particular bandwidth, the flow would experience a decrease in throughput. Therefore, quality of service (QoS) is required. Authors came up with a clique-based model of I2ILP in [20] to introduce IROCX, which is a routing algorithm for wireless mesh networks. It applies linear coding for routing while maximising the benefits of network coding and considering interference.

2.1.5 Improvement on COPE

Since COPE was implemented to test the practicality of network coding in unicast traffic, many studies made their focus on COPE [12]. The latest improvement on COPE introduced by authors in [21], focused on UDP unicast flows over coded wireless networks with constructive intersession network coding, such as COPE [12], based on prior work in [22]. In previous work, two mechanisms were developed: one, a network coding aware queue management (NCAQM) aimed at creating more network coding opportunities in the presence of congestion and the other, a network coding-aware MAC-level packet prioritisation (NCAPP) that assigns higher priority to coded over uncoded packets. Two mechanisms were compared, and achieved similar throughput gain. However, they were improved and combined into a novel network coding-aware priority queuing scheme, referred to as NCAPQ. The new scheme improved throughput compared to COPE by a factor of ten. However, it is claimed that this increase in throughput comes without significant loss in fairness.

The study in [23] tried to analyse the performance of COPE [12] by constructing a simple network model to provide insight into network design strategies. The author showed that the model could be used to address various approaches to network coding, such as MAC and multi-packet reception, and evaluated their effects on network throughput. They considered several topology components which exhibit the same non-monotonic saturation behaviour found within the COPE experiment [12]. It is proven that fairness allocation by the MAC could severely impact performance and cause non-monotonic saturation. Based on that they proposed a model that estimated the possible gains for the cross-layer design of network coding using multi-packet reception and MAC. The results of their study showed super-additive throughput gains six times higher than routing.

However, the authors in [24] argued that the key to COPE's success depends on the interaction between COPE and the MAC layer, and that the local fairness enforced by the MAC layer among competing nodes plays an essential role. Based on this analysis, authors in [24] proposed a simple modification to the COPE system that could further improve the network throughput through a single virtual queue for each input-output pair in the coding node. Implementing a virtual queue gives higher priority to coded packets. This modification to COPE is straightforward but has led to a significant gain in throughput when the offered load is high.

Existing network coding schemes such as COPE require the exchange of messages between neighbouring nodes in order to encode and decode data packets correctly. Nevertheless, such cooperation leads to high packet overheads and degrades system performance, especially when network traffic is high. For that reason, authors in [25] proposed a new adaptive scheme with the primary objective to control the waiting time of overheard packets that are stored in a buffer to achieve a trade-off between throughput and overhead. NS2 simulation results showed that there is tremendous performance improvement concerning throughput and overheads when the adaptive scheme is used as compared to a fixed scheme in high-load network traffic.

In favour of COPE, authors in [26] argued that linear network coding scheme was based on complicated linear operation, while the COPE was executed by way of pairwise Xor which was uncomplicated and low efficiency. Authors in [26] introduced group-based Xor network coding, which was putting forward. The scheme had the properties of high efficiency and low complexity. Their results and theoretical analysis showed that their proposed solution had higher throughput gains, lower delay and much fairer than traditional schemes.

In a very recent evaluation of COPE, the authors of [1] compared the analytical and experimental performance of COPE-style network coding in IEEE 802.11 Ad-hoc networks. In their experiments, they used a lightweight scheme called CATWOMAN that can run on standard wi-fi hardware. They presented an analytical model to evaluate the performance of COPE in simple networks. Their results showed a predictive quality of this model. By carefully examining the performance in two simple topologies, (Cross and Alice and Bob), their observation proved that the coding gain resulted from the interaction between network coding and the MAC protocol, and the gap between the theoretical and practical gains is due to the different channel qualities of sending nodes. They claimed their study could help to understand how to design larger mesh networks that use network coding.

The authors in [27] proposed CANCAR, a distributed dynamic congestion-avoidance routing procedure for wireless mesh networks that makes use of network coding awareness. Scholars in [27] implemented CANCAR as a proactive link-state per-hop routing procedure, which could be based on any conventional routing and network coding methods. CANCAR can detect the most highly-loaded node, and prevented the node from saturation by diverting some of the least coded traffic flows to alternative routes. Therefore, CANCAR achieved higher coding gain by the remaining well-coded traffic flows on the node. The simulation work in [27] showed that the CANCAR combined well with COPE, and achieving higher throughput.

2.1.6 Improvement of Distributed MAC Protocol using (XOR) Network Coding

The work in [28], the authors introduced an extension to distributed MAC protocols that enhance the efficiency of coding decisions and allows calculation of packets before they are transmitted to their destination. They provided an algorithmic network coding at MAC that manages the stored packets intelligently in the MAC queue in each node. The algorithm improves the knowledge of the node for available correct coding opportunities by using opportunistic acknowledgements. They demonstrated that their protocol shows significant throughput improvement compared to standard network coding. However, this work is related to Xor-based network coding, which works on the neighbourhood knowledge of received packets. Similar work was done in the case of unicast traffic [29], showing a 20% to 30% throughput increase using their RLNC-proposed algorithm, called Multipath Code Casting (MC2). Another paper shows almost two-fold throughput increases, authors in [30] compared this with traditional routing when their RLNC-based algorithm is applied, named Optimised Multipath Network Coding (OMNC). However, the work in [31] has shown that if an appropriate link scheduling for NC is not performed in TDMA based in wireless mesh networks, the end-to-end delay may be increased (although a network coding gain can be obtained). Therefore, the authors in [31] proposed a novel joint network coding and scheduling scheme to increase network throughput with end-to-end delays guaranteed. The fundamental concept of their proposal is flow-based, and employs a new concept, "duplicated allocation followed by resource release: DARR" [31]. Thus, the slots scheduled on a path are sequentially arranged within a frame even when network coding is used.

In the multi-rate case, the researchers in [32] proposed a multi-rate IEEE 802.11b MAC protocol with network coding, but their work was analytical and based on a framework formulated for many stations, and a two-dimensional discrete-time Markov chain model state transition diagram. Furthermore, the authors used the Markov chain for the backoff phase and the value of the backoff counter. As a result, nested summations of saturated throughputs formulated the saturated throughputs. They claimed there was an advantage of the proposed multi-rate MAC protocol with network coding over a basic IEEE 802.11b using RTS/CTS mode.

The study in [33] takes a different approach, focusing on the throughput and fairness in single-relay multi-user wireless networks. An interesting claim was based on the fairness of wireless medium access among stations (STAs), the access point (AP) and the relay station (RS) results in asymmetric bidirectional flows via the RS; therefore the wireless throughput decreases substantially. They solved this problem by introducing optimisation of minimum contention window size, which was implemented at CSMA/CA with network coding. It assigned appropriate transmission opportunities to both the AP and RS. Their solution can help to reduce bottleneck links in networks and achieve fairness between uplink and downlink flows. They used numerical analysis and simulations to evaluate the performances of CSMA/CA and network coding in single-relay multi-user wireless networks. However, their study was based on the scenario of central management as the access point.

2.1.7 Xor Network Coding on MAC Layer

A recent paper [34] provided a new strategy for MAC in wireless meshed networks by identifying overload scenarios in order to provide additional channel access priority to the relay. The critical point in this study is that the relay needs to adjust its back-off window size according to the incoming and outgoing packet ratio. They claimed that the throughput in cross topology in their novel MAC reached 66% for network coding and 150% for classical forwarding in theory, which is believed to be 33%. They claim that their result showed an even more substantial gain for network coding, up to 65% over forwarding, as it copes better with channel losses under high-load scenarios. In their implementation, the algorithm works only when the system reaches high loads in order to increase the priority of a relay in an intersessional coding region.

The work in [35] claimed that there was a short period unfairness when nodes were competing for wireless channels, which decreased the opportunity of intersessional network coding in the wireless mesh network. Based on that, a coding-aware cross-layer optimisation mechanism was proposed. Again the key element of this study was the priority of coding flow calculated in relay nodes according to the packet number of the previous flow in the forward queue, and the values of the priorities were sent to the corresponding upstream nodes to adjust their MAC parameters. Their experiments and simulation showed the proposed mechanism could balance the traffic of coding flows in the relay nodes in a short period and create more coding opportunities.

In another study [36], researchers were considering multiple unicast scenarios in a random access network with exponential back-off. This comprised of an access point and wireless nodes, with authors in [36] proposing a new network coding-based packet-forwarding algorithm. Authors considered the pitfalls of multiple unicast scenarios and discussed the proposed scheme, which converts it to a combination of multiple anycast and a multicast scenario. Besides, they discussed the critical advantage of their proposed scheme when nodes attempt to access the channel based on slotted Aloha with exponential back-off. Their forwarding scheme was discussed for two cases when $K = 2$, or 3. By having such a scheme of multiclass open queuing network in [37], [38], [39] as previously studied, they could reach the maximum stable download throughput in the network.

Authors in[40] presented potential throughput gains of up to 630% when multi-packet reception and the network were mixed together. They argued that NC and MPR gain at saturation is not maximised in 802.11 networks. Also, the current 802.11MAC is fair to nodes, but it is unfair to flows of information in Multi-Hop networks. Their proposed approach showed a significant increase in the achievable throughput when neither NC nor MPR are used in similar networks.

Authors in [41] presented an algorithm of MAC-based network coding specific to butterfly networks called MBNC (MAC Based Network Coding). According to the authors' observation, the differences in the numbers of buffered packets for upstream flows in the coding node's FIFO output queue, could increase coding opportunities by dynamically adjusting the contention windows of the MAC layers of upstream nodes, thus improving network performance.

Authors in [42] formulated the status of a router in a wireless mesh network as a directed graph. Based on the directed graph, they proposed an algorithm for finding coding opportunities, which formed cycles in a graph to find opportunities. The computational complexity in [42] was $O(n^2)$, where n represents the length of the packet queue in a router. The proposed XNC demonstrated that the algorithm improved the average coding gain by 8.7%, compared to the Xor algorithm in [12].

A study in [43] proposed a new encoding strategy, Xor-Top, which depended on local topology information. XOR-Top tries to decode coded packets stored in its buffer. If we compare it with COPE [12], XOR-Top could precisely identify coding opportunities according to local topology, and uses less bandwidth for signalling. Xor-Top boosted the throughput by up to 240% over traditional unicast and up to 150% over COPE. In addition, average end-to-end packet latency was reduced.

CAPO was proposed in [44]. CAPO attached a module which adhered non-encoded packets together opportunistically. CAPO conveyed more packets in a single transmission than COPE [44] using the proposed module. Moreover, CAPO had a lazy decoding process which endeavoured to decode stored coded packets, to increase the decoding probability. As a result, there was a reduction in the number of retransmissions using the minimum number of decodings. Authors in [44] compared the lazy decoding process with the XOR-Top in [43]. Their work showed that the number of decodings done via the lazy decoding process was much lower than with XOR-Top.

In optical core networks, the work in [45] proposed a framework to redesign the dedicated path protection considering both network-side and client-side implementation by incorporating photonic XOR network coding. The authors in [45] exhibited numerical results on a realistic network highlighted the efficiency of their proposal with a saving of up to 25% wavelength resources and indicated client-side dedicated path protection with NC as the most capacity-efficient design.

Another study in [46] came up with a novel network-coding aided medium access control (MAC) protocol with mobility support (NCMOB-MAC). NCMOB-MAC was compatible with the Automatic Repeat reQuest (ARQ) scheme. NCMOB-MAC can be implemented in vehicular ad-hoc networks (VANETs). NCMOB-MAC offered a new way of coordinating between a sender and destination with a set of fixed relay nodes. The relay nodes adjusted contention window sizes based on priorities and the network coding opportunities. Altering the contention window size, it depends on the position of a moving node in order to reduce packet loss. As a consequence, throughput soared to 45% and energy efficiency improved by up to 68%, compared to the network coding based cooperative MAC protocol.

Authors in [47] investigated network coding in Multi-Hop wireless networks for unicast. They built a simple network where network coding reduced the overall throughput. This resulted when network coding schemes were greedy which reflects the fact that all opportunities to encode and broadcast packets were exploited. Authors in [47] concluded that if network coding and scheduling are designed separately, then the expected throughput in network coding might not be obtained. Authors suggested the use of adaptive schemes that use network coding opportunities, and the need to link scheduling choices with network coding decisions as a necessity. Therefore, authors in [47] proposed Xor-Sym, a scheduling scheme that used COPE[12]. Xor-Sym showed a lower complexity than COPE [12], with similar performance.

Moreover, the work in [48] addressed the Distributed Coordination Function (DCF), which had proven that critical inefficiencies could arise in the case of Multi-Hop packet forwarding. As a result, a new MAC scheme was proposed based on the virtualisation of the Point Coordination Function, which works on chain topologies with bidirectional traffic flows. The scheme used a token-like access mechanism in tandem with network coding. However, the solution required periodical handshake packets to coordinate. Analytic and simulation results proved that their scheme could provide an aggregated uplink and downlink throughput comparable with bidirectional point-to-point transmissions.

A paper in [49] proposed FlexONC (flexible and opportunistic network coding) as a solution to fully utilise the broadcast nature of wireless using coding opportunities. Authors assumed a fixed route between the source and destination that every packet should travel through could cause buffer overflow in some specific relay nodes, as in case of eight hops. Their proposed solution is comparable to BEND in [50] and COPE in [12]. However, it is based on the specific topology of eight nodes. Authors in [49] considered throughput when bit error rates (BER) were increased. This is another angle of looking into the impact of the link layer on throughput in wireless mesh networks.

Recent work in [51] focused on an analytical study on throughput and end-to-end delays in network coding of interflow in Multi-Hop wireless networks in terms of the IEEE 802.11 DCF Function. They applied queuing theory to propose an analytical framework for bidirectional unicast flows in Multi-Hop networks. In their analytical model, they considered the exponential back-off time with clock freezing and virtual carrier sensing. Authors in [51] looked into several parameters, for instance, the probability of successful transmission concerning BER and collision probability, waiting time and retransmission functionality. They concluded that the great capacity of network coding regarding the maximum stable throughput showed up when the physical and MAC layer were taken into account, and that it can work well in smaller topologies. However, with an increased number of relay nodes in the network, the maximum stable throughput is similar to a traditional forwarding scheme.

Much more recent Authors in [52] used two-way exclusive OR (XOR) relay to enable hidden nodes to exchange data with low delays and high data rate, while keeping signal processing simple. Researchers in [52] analysed practical two-way XOR relaying systems, where finite relay buffer, non-negligible signalling overhead, and lossy wireless channels are all captured. A two-layer model was developed to characterise practical two-way relay systems, which was later reformulated into a Markov process after combining inter-layer state transitions of the two-layer model. Using Markov techniques, they evaluated the steady-state probabilities of the Markov process and. They focused on throughput, delay, and packet loss by using Monte Carlo simulations. Authors in [52] claimed that their model could be used as an online tool to configure the buffer resources, adapting to wireless channel conditions and signalling requirements.

2.1.7.1 Xor Coding on MAC-layer retransmission schemes

In this work [53], a unique MAC layer retransmission algorithm referred to as Xor rescue (XORR) was proposed by authors in [53]. It estimated the reception status without overheads and came up with a new coding metric, which accommodated the effects of frame size and channel condition. XORR applied NC-aware fair opportunistic scheduling, which is theoretically proven to be fair. Their results showed that XORR outperforms the non-coding fair opportunistic scheduling and 802.11 by 25% and 40% respectively.

The study in [54] claimed that the earlier network coding-based retransmission [55] was not as effective as their solution (BENEFIT). The authors in [54] proposed BENEFIT for better latency and bandwidth usage. BENEFIT was based on the fundamentals of network coding for retransmitting lost packets in single-hop wireless multicast network. Authors in [54] suggested that using their algorithm would help network architects to adjust throughput-delay requirements of the network based on rules extracted from BENEFIT.

A novel dual Xor hybrid automatic retransmission request model ($XOR^2 - HARQ$) was introduced by [56] for wireless broadcasting. This is different from the traditional network coding based HARQ (NC-HARQ). By performing additional Xor operation, a receiver could recover two lost packets concurrently with one retransmission. Work in [56] was analytical. However, $XOR^2 - HARQ$ was factually less of required retransmissions than that of NC-HARQ.

Another work in [57] addressed retransmissions at the MAC layer in a lossy environment.

Authors in [57] proposed a neighbour-assisted coding-aware opportunistic retransmission scheme to improve network throughput. The fundamental idea in [57] was described: "If a node fails to receive a packet due to link loss, its neighbouring node(s) receiving the packet can assist the retransmission of the packet, possibly encoded with other packets (s) via localised network coding. If such retransmission is expected to be beneficial". Scholars in [57] claimed that the proposed algorithm could reduce the overall number of packet retransmissions at the MAC layer.

Moreover, the work in [58] proposed XBC, a Xor coding to optimise the retransmission buffers on relay nodes that support in-network recovery. XBC was based on two parameters, the coding degree and coding distance for random and burst losses. It was evaluated by using Snoop as the baseline. XBC showed that TCP throughput could be improved by up to 20%.

Finally, authors in [59] introduced a low-overhead retransmission called blind Xor (BXOR). This Xored multiple packets into a single retransmission. BXOR could recover an increased number of lost packets per retransmission. Additionally, it can keep overheads low by using blindly Xored packets through the retransmitter only. The authors claimed that BXOR could outperform existing retransmission schemes if the conditional reception probability (CRP) of the combined packets is higher than 50%. Their simulation experiments showed that within a 110 m radius of the original packet transmitter, BXOR reduced the packet reception failure rate to 60% when compared with other retransmission schemes.

2.1.8 Xor Network Coding on TCP Protocol

Authors in [60] introduced a resilient and robust TCP-aware network coding with opportunistic scheduling in ad-hoc wireless networks. Furthermore, authors in [60] considered a TCP parameter such as a congestion window, and wireless channel conditions to enhance TCP throughput. Their simulation results showed that using traditional network coding combined with opportunistic scheduling could improve the performance of TCP up to 35% in wireless mobile ad-hoc networks. Authors in [60] likened the congestion window to channel condition, for example, TCP sender responded upon realising any changes in link conditions. If the TCP sending rate is very high, it will choose a scheduler with the highest channel rate (QAM64), and if link quality is low, it chooses (QAM16). This study does not consider RLP or related delays. It reduces the congestion window based on link quality.

Meanwhile, authors in [61] investigated the benefit of network coding for TCP traffic in a wireless mesh network. They implemented their network coding in the 802.11a standard and authors in [61] focused on measuring TCP throughput. Unlike previous implementations of network coding, authors in [61] used off-the-shelf hardware and software. Moreover, there was not any modification in TCP or the underlying MAC protocol. Scholars in [61] showed that network coding not only reduces the number of transmissions by transmitting many packets through a single transmission, it results in a smaller loss probability due to reduced contention on the medium. Moreover, due to asynchronous packet transmissions, there was a small opportunity for encoding, and it resulted in marginal throughput benefits. Coding opportunities could be increased by inducing small delays at intermediate nodes. However, this extra delay at intermediate nodes resulted in longer round-trip times that adversely affect TCP throughput. They argued that a delay in the range of 1 ms to 2 ms could help to maximise TCP throughput. For the topologies considered in this paper, network coding improves TCP throughput by between 10% and 85%.

Work in [62] demonstrated how the use of network coding might bring about synchronisation between TCP flows. Authors suggested that, under particular conditions of random packet loss, the aggregate throughput might be low when network coding is used, and the use of opportunistic listening to perform network doing may expose TCP flows to additional losses, and this results in a lower network coding gain and, in some cases, TCP might be better without network coding. Another finding that with high random loss rates, the loss synchronisation effect became evident and contributes to the degradation of TCP performance over wireless networks. Their study and observation were based on the random loss which it is explained to be non-congestion losses that may often link to wireless nature links and characteristic of lower layers (MAC). Their observation that when the link is subject to random losses at both middle node and source, there is no network coding gain or improvement in good-put for TCP. Moreover, when a high loss rate is in middle link, the per-flow good-put is better when random losses happened at the middle link. Authors explained that as a result of high loss synchronisation. Lastly, the strong correlation between synchronisation rate and coding rate, the more the flows are, the more the coding ratio (coding opportunities). That might explain the marginal higher improvement in good-put. However, in presence in high loss rate, this observation becomes invalid. On the other hand, the authors argued that opportunistic listening would lead to few coding opportunities in TCP, it would expose TCP flows to higher loss rates. The results in [62] are only valid under some assumptions on butterfly topology, and traffic pattern and loss models, these assumptions could not work in different topologies such as grid or chain.

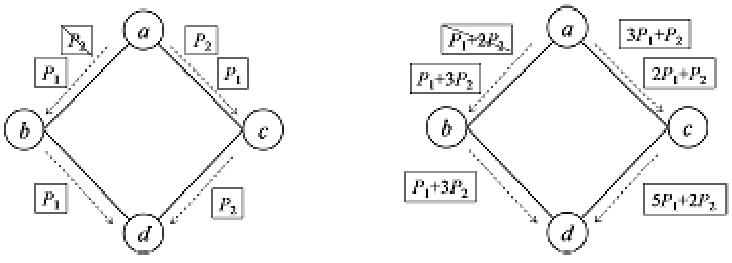
Authors in [63], looked into the interaction between TCP's congestion control and the network coding that was done by wireless relays. The primary focus was on the influence of the coding buffer used by relays. They suggested if timeout for the coding buffer is longer than 30 ms and the buffer is more than four packets. It did not lead to a higher percentage of coding opportunities, it might affect the overall performance, since keeping a packet in the buffer for a long time may increase the Round Trip Time (RTT), and as a result, it will decrease throughput. Another finding stated that when the Frame Error Rate (FER) is higher than 10%, the throughput will be higher. If errors affect the middle link in X topology, Network coding can outperform the traditional operation for FERs lower than 0.3. It can be explained as synchronisation between the two flows, which yields into a high number of coding opportunities. Moreover, if losses are just over the side links, Network coding will be worse due to failing to decode packets, when FER is higher than 7%. Authors in [63] linked their finding with previous work in [62] that TCP performance could be enhanced by network coding when wireless channel quality is above a certain level.

Authors in [64] presented and characterised the performance of CORE, a protocol that brings together the efficiency in spectrum usage of inter-session network coding schemes and the robustness against packet losses of intrasession network coding. Using mathematical analysis and of the gains of CORE followed by protocol design and implementation details of CORE's. Authors in [64] provided extensive measurements with off-the-shelf wireless nodes under the various channel and system conditions comparing CORE to other state-of-the-art approaches, namely, forwarding (no coding) and COPE (inter-session network coding). These measurements supported their theoretical findings, showing that CORE not only outperforms COPE and forwarding in general, but that order of magnitude gains are possible for cases with high packet losses. Specifically, CORE has a throughput gain of more than ten times over a COPE-like scheme and seven times over forwarding when the error ratio is 50 % on all links. Beyond these gains over other protocols, CORE implementation can achieve close to optimal performance with a gap of less than 0.43 dB.

A new study in [65] focused on 802.11 standard families, and authors in [65] argued that these standards incur significant overheads to transmit messages in multiple sources - multiple destination (MSMD) schemes. Authors in [65] contemplated the results of the emulation with COPE-like network coding and various Transmission Control Protocol (TCP) congestion control algorithms in wireless networks. In the implemented scheme, two nodes stream data to each other through the bottleneck relay node. The Bottleneck Bandwidth and RTT TCP protocol (BBT) demonstrated the best results in this relaying scheme with COPE-like network coding.

2.2 Random Linear Network Coding

RLNC is usually used with the probabilistic approach [66], [67]. After the introduction of network coding theory in [2], many scholars became aware of the importance of network coding and route selection in wireless networks. Opportunistic routing takes advantage of the wireless nature: it is a broadcast medium. Consequently, opportunistic routing enhances transmission reliability and boosts the network throughput over traditional routing protocols. For example, if several nodes can overhear a transmitting of the data packet from a neighbouring node, any neighbour next to the sender could forward the data packet close to the destination. However, there are specific criteria that need to be met while meeting certain conditions [14]. Opportunistic routing fundamentally has two distinctive characteristics: one, forwarder-set selection (this would be an end-to-end forwarder set selection which is chosen), and two, prioritisation (preventing duplicate packet forwarding). This is established based on a metric such as geo-distance, hop-count, expected any-path transmission (EAX) or expected transmission count (ETX) [14]. ExOR and MORE are classified as forms of opportunistic routing, and we are going to discuss both ExOR and MORE in the next section.



(a) ExOR (b) MORE

FIGURE 2.5. A Transmission Scenario using ExOR and MORE Protocols. Figure taken from [68].

2.2.1 ExOR

The traditional routing protocols in unicast traffic depend on forwarding a packet through the network on a hop-by-hop basis. Authors in [68] introduced extremely opportunistic routing (ExOR) as a unicast routing technique to increase the throughput in wireless Multi-Hop networks. It is vital to recall that ExOR does not apply network coding, but it can improve the network performance, when ExOR is used, the source node divides data into batches (we can call these generations). Later it broadcasts the packets while assigning a priority among the next hop neighbors who are responsible for delivering data packets. If the highest priority node (forwarder) receives a packet, it waits some time according to its ranking then it can send an ACK. Otherwise, the lowest priority forwarder is responsible for acknowledging that packet. As a result, a forwarder would broadcast only packets that are not ACKed by higher priority forwarder.

2.2.2 MORE

It is clear that ExOR increases the throughput and avoids simultaneous transmissions using local scheduling. However, it depends on the MAC layer, which can prevent spatial reuse. Besides, it is not suitable for multicast traffic. A better a opportunistic routing model named MORE was proposed in [69], which introduced network coding to deal with ExOR disadvantages. When a forwarder node receives a packet, it first checks if it is an innovative packet, and if not, it drops it. A combination is considered innovative if it is linearly independent of other previously received, and based on that the forwarder nodes creates a random linear combination of all the received packets that belong to the current batch or generation and broadcasts it. When the medium is not busy, MORE exploits waiting time to win the medium to combine packets that were received previously. When the destination node has enough combinations to decode, it sends an ACK back that tells the following batch it was correctly received, and that will help the intermediate nodes to flush their queue and memories. It is clear the difference between ExOR and MORE is that ExOR needs to coordinate with other nodes in the network and that it heavily depends on the MAC layer for scheduling to build a specific ranking among other nodes in the network figure 2.5

shows the difference between MORE and ExOR as previously explained.

2.2.3 Improvement on MORE protocol

Researchers in [70] claimed that the network coding opportunistic routing (NCOR) approach in wireless networks can enjoy high throughput and excellent reliability. However, NCOR suffers problems of better deciphering latency and frequent packet collisions. Additionally, NCOR has the deficiency in helping QoS factors. Therefore, Authors in [70] proposed a delay-restricted opportunistic routing protocol (DCOR) primarily based on broadcasting the MAC protocol to aid low-latency service transmission in mesh networks with NCOR routing strategies. Their simulation proved that the optimised NCOR routing approach and DCOR routing should support fair transmission of many flows with varied service types, that decreased end-to-end delays may be achieved within the low-load network conditions and congestion might be averted in high-load network conditions. However, in lossy environments like wireless networks, a redundancy control is needed to mitigate losses. As authors in [71] claimed, a few implementations propose adding a set amount of redundant packets in line with batch, but this leads to the hazard of losing bandwidth by means of over-redundancy or unsuccessful deciphering due to packet insufficiency, and dramatically degrades the overall performance. Therefore, adaptive redundancy control for network coding (ARC) adds redundant packets dynamically to mitigate losses. ARC is primarily based on the use of MAC acknowledgements as a response to link quality, and as opposed to determining the ideal variety of redundant packets, ARC finds the best time to transmit redundant packets to cover losses. Authors in [72] came up with the RLNC broadcast protocol called ARLNCCF, and claimed the critical factor of RLNC is the definition of generations: small generation sizes may help to reduce decoding complexity and packet latency, and allowing packets from specific sources to be combined can enhance PDR, lessen packet latency, and in addition lessen the wide variety of packet transmissions on the MAC layer. The analytical model confirmed that cross-source coding reduces the desired number of MAC transmissions by between 8% and 20%.

In routing, a new study suggested by the author [73] proposed PlayNCool which was considered to be an opportunistic protocol with local optimisation based on network coding to increase the throughput of a wireless mesh network. The protocol depends on a helper that will work when only after it hears enough packets to be truly useful. They showed that PlayNCool provides three-fold gains in individual links, which translates into a significant end-to-end throughput improvement.

In more generic approach, the work in [74] focused on the problem of multiple source broadcasting in mobile ad hoc networks. The author's observation was that RLNC provided increased resilience to packet losses compared with XOR-based coding. In their study, they developed an analytical model. Authors in [74] revealed that combining RLNC with probabilistic forwarding, which was the approach authors applied, had significantly impact RLNC's performance. Thus, the authors introduced a novel approach to combine RLNC with a deterministic broadcasting algorithm in order to prune transmissions. They proposed a connected dominating set-based algorithm that worked in synergy with RLNC on the "packet generation level." Since managing packet generations is a critical issue in RLNC. Their proposed scheme is suitable for mobile environments and did not compromise the coding efficiency.

The latest development in MORE protocols comes from the authors in [75], who proposed I^2MIX, that can benefit from integrating inter- and intra-flow wireless network coding. It was shown through trace-based evaluations that I^2MIX could reduce the total number of transmissions by 21-30%, While the MORE protocol uses an intraflow coding with opportunistic routing, but interflow coding requires best-path routing, which means I^2MIX and MORE cannot be coupled together. As a result, the authors presented a routing algorithm called OSPR, which can find the paths that expand the coding gain of IMIX.

2.2.4 Random Linear Network Coding on MAC Layer

Authors in [76] investigated random network coding performance on wireless time division multiple access channels (TDMA). They confirmed that in a network coding scheme over TDMA, MAC should approach theoretical network potential in lossy networks. Network coding can provide reliability against packet loss and mobility, and authors in [76] emphasised the significance of the forwarding rate to ensure reliability and overall performance of in the network, while intrasessional network coding is being used.

A study in [77] addressed the problem of assisting broadcast traffic in ad-hoc networks by applying network coding. The fundamental concept of this work was primarily based on letting intermediate nodes encode a couple of packets into a single packet. Their solution was supported by evaluation of the impact of RLNC, in a realistic MAC layer in an ad-hoc network environment through the network simulator ns-2.

Authors in [78] presented the Rainbow protocol, a content distribution protocol for Multi-Hop ad-hoc networks. The proposed protocol uses a content-directed MAC protocol, through which transmission priority is given to those nodes most able to hand over useful content material to their neighbours. Rainbow applies a MAC-priority technique, wherein the concern of packet transmission from a node relies upon on the rank of the coefficient matrix related to the coded content the node holds. Simulation results in [78] indicate that Rainbow achieved improvements of 1.3 to 1.9 times in the time which is needed to disseminate content over other flooding techniques.

In [79] the author introduced a new timing strategy for the combination of data packets in random network coding. They analysed the impact of a realistic MAC layer, transmission schedule and packet combination strategy on random network coding. This was based on the observation that the impact of the MAC layer was not as substantial as expected, and that therefore, the use of a mechanism to avoid the collision problem only resulted in limited improvements in the packet delivery ratio performance and affected the latency. In their study, they pointed out that packet combination strategy plays a fundamental role. Again, authors in [79] stated that all the schemes and protocols studied in their investigation did not achieve the theoretical performance in [80], and thus authors in [80] provided the motivation for investigating a suitable combination strategy.

Authors in [81] concentrated on the design of a practical broadcasting strategy based on network coding in wireless ad-hoc networks. Theirs was a practical dissemination algorithm exploiting network coding for data broadcasting in ad-hoc wireless networks. They stated that deadlock situations might occur where the delivery process is interrupted, and some of the nodes never gathered the required packets. In order to solve it, they developed an original proactive network coding (ProNC) strategy. ProNC conformed its transmission schedule according to the decoding status of neighbouring nodes. Therefore, ProNC could detect when nodes needed additional packets in order to decode and act accordingly. Later, authors focused on the behaviour of network coding schemes in multi-rate ad-hoc environments. The authors in [81] came up with a lightweight rate adaptation heuristic. ProNC applies an estimation technique based on the node's density to reduce the collision probability, and ProNC deliberately swamps the network with a certain amount of innovative packets.

The author of [82] proposed a novel MAC protocol based on the Aloha protocol, which could perform network coding processes during one packet per slot at each transmitting terminal, with the network coding coefficient vectors used by different transmitting terminals orthogonal with each other. However, the traditional collision assumption was changed to: if there are more than K packets transmitted synchronously, the receiver cannot decode any packets, which is called 'collision'. This resulted in more than one transmitting terminal sending packets in the same time slot. The study in [82], was based on a collision definition which was not practical or realistic.

Meanwhile, the authors in [83] proposed the inter-session network coding mechanism that could be implemented in resource-limited sensor motes. Their solution decreased the overall traffic in the network, and thus the energy consumption was reduced. The proposed solution took into account deep header compressions of the native 6LoWPAN packets and the hop-by-hop changes of the header structure. Implemented solution decreased signalling traffic that was typically happening in network coding deployments. The authors in [83] validated their scheme considering end-to-end packet delay, packet loss ratio and energy consumption. The results showed that the proposed technique could work well in delay-tolerant sensor networks.

Recently, authors in [84] demonstrated a feasible way to design network coding as a virtual network functionality offered to the communication service designer. They suggested the integration of network coding and Network Function Virtualization (NFV) structural designs. The combination was depicted as a toolbox for service designers can use for flow engineering to accommodate network coding technique. Authors in [84] argued that the proposed framework could be .be tailored and optimised depending on the service and underlying virtualised system or network. Again authors in [84] used case where they design geo-network coding, an application of network coding for which coding rate was optimised using databases of geo-location information towards an energy-efficient use of resources. Their numerical results showed the advantages of both the proposed network coding design framework and the specific application.

Regarding future technologies, authors in [85] highlighted the advantages of benefits and applications of network coding for 5G Network and device-to-device communication. Scholars in [85] presented the state-of-art research, theoretical benefits, and detail how network coding could improve 5G Networks and D2D communication. Their results demonstrated that RLNC network coding could almost double the network throughput while increasing network robustness and decreasing the overall time to disseminate messages.

2.2.5 Random Linear Network Coding on TCP Protocol

TCP used to be designed for reliable service delivery in wired networks. In wired networks, packet losses show up due to congestion relief. Nowadays, TCP is additionally utilised in wi-fi networks where congestion and numerous other reasons for packet losses exist. It results in decreased throughput and accelerated transmission round-trip time when the state of the wi-fi channel deteriorates.

Authors in [86] proposed TCP/NC and argued that rateless codes and batch-based coding are not compatible with TCP's retransmission and sliding-window mechanisms. TCP/NC implemented network coding into TCP with only minor changes. The primary goal in [86] is to mask losses in TCP by applying random linear coding. TCP/NC uses TCP sources and buffers as an encoding buffer which represents the role of the coding window. The sender generates and sends random linear combos of the packets in the coding window. The coefficients used in the linear mixture are also included in the header. The decoder acknowledges the closing packet by requesting the byte sequence number of the first byte of the first unseen packet, via the use of an ordinary TCP-ACK. Any ACKs generated by the receiver TCP are now not dispatched to the sender. Alternatively, ACKs are used to update the receive window field generated by the decoder [86].

An analytical study in [87] compared the performance of TCP and TCP/NC in [86]. Authors in [87] implemented a simple framework introduced by Padhye et al. Their analysis described the throughput of TCP and TCP/NC as an act of erasure rate, round-trip time, maximum window size and the duration of TCP connection. They concluded that network coding was sturdy against erasures and failures, and could avoid performance degradation of TCP in the lossy scenario. Furthermore, TCP/NC proved to be better than standard TCP, also TCP/NC could maintain larger window sizes and faster rapid growths. They recommended using TCP/NC in a lossy wireless environment.

In the first empirical study, authors in [88] proposed a reliable multipath protocol called Multi-Path TCP with Network Coding (MPTCP/NC). Authors in [88] developed a model for calculating achievable throughput. A comparison between MPTCP and MPTCP/NC was presented using both the empirical data and mean-field approximation. Authors in [88] suggested the use of MPTCP/NC to overcome the challenges of packet scheduling in the presence of a lossy wireless network, and claimed that MPTCP/NC can improve the quality of service for mobile users.

The work in [89] introduced a numerical analysis model, which it calculated throughput of network coded TCP against lossy links and error correcting coding. The work in [89] focused on factors that affect the network coded TCP such as maximum window size, end-to-end erasure rate and redundancy factor. Authors in [89] claimed that their model could precisely predict the average maximum window and average throughput under a wide range of erasure rate, maximum window size and redundancy parameter. They calculated throughput according to study in [90].

The study in [91] tried to address the inability of TCP protocol to differentiate between losses due to congestion and random packet losses in the noisy channel. Researchers in [91] proposed an algorithm to fine-tune the redundancy factor R of the TPC/NC protocol proposed by Sundararajan et al. in [86]. By adding some extra functionalities to the network coding layer. They developed a loss differentiation scheme to fine-tune R, it was hinged on the TCP Vegas loss predictor and the cumulative feedback information of ACKs and duplicates ACKs, These ACKS were indicators of the network condition. TPC/NC was full-duplex, and it handles multiple TCP connections simultaneously. They simulated their implementation on OPNET, their algorithm used in TCP/NC [86], it showed enhanced TCP throughput than the standard TCP/NC, TCP-Reno, TPC New Reno, and TCP Reno with SACKS.

TCP with network coding (TCP/NC) in [91] causes the packet loss, which was a result of the wireless transmission error, and it did not affect congestion control. Authors in [92] used delay-based congestion control of TCP Vegas to manage with the congestion issue of TCP/NC in [91]. Authors in [92] claimed that TCP/NC [91] was simplified and it led to unfairness. Therefore, congestion exposure algorithm (CEE-TCP/NC) was proposed, it proposed to ensure TCP/NC would be friendly to TCP protocols in the case of congestion. CEE-TCP/NC changed TCP's loss-based congestion indicator with considering gaps in the ACK that received by TCP sender. Authors in [92] claimed that CEE-TCP/NC detected packet loss due to congestion without depending on the variation of RTT or duplicate ACK. Again, authors of CEE-TCP/NC used theoretic analysis and simulation and stated that CEE-TCP/NC was seen to be more sensitive to congestion and friendlier to TCP flows when it was running in congested wired bottleneck compared to TCP/NC. Moreover, CEE-TCP/NC could reach high throughput as other TCP/NC schemes in a lossy environment, and it can stop damaging the performance of other TCP flows as it happened in TCP/NC in [91].

In lossy wireless networks, random losses are affecting TCP as an indicator of congestion. Therefore, TCP downs the sending rate, which leads to low performance. However, TCP/NC [91] could resolve this issue. TCP/NC disguised the random losses by allowing the receivers to acknowledge every degree of freedom although original data is not decoded fully. Thereupon, TCP masks the random losses without affecting the performance. However, authors in [93] claimed that TCP/NC sends redundant traffic for a pre-set interval which does not recover random losses in time. They proposed Dynamic Coding (DynCod), which was an end-to-end adaptive redundancy control based on upgrading the coding schema in TCP/NC. The main idea was the destination can acknowledge the source if the latest data sent from the sender was decodable or not and how many packets were lost through Ack packets. Simulation results in [93] showed that DynCod overtook TCP/NC [91] and TCP Reno in case of throughput and packet delivery time.

Another study in [94] scrutinised the impact on the stability and equilibrium of the TCP Reno congestion control mechanism when a network coding layer is applied in the TCP/IP stack. The proposed TCP-NC protocol linked with random early detection (RED) as active queue management mechanism. Authors in [94] claimed that stability proved in the absence of forwarding delay. Authors in [94] introduced TCP-NC-RED , which explained that TCP-NC-RED became unstable when delay or capacity increases, as TCP Reno did when the redundancy factor soared.

With the advancement in network coding in wired and wireless networks, the congestion avoidance issues in TCP are being investigated from the perspective of a network coding paradigm. There are some challenges in integrating network coding along with TCP. It will bring some complexity in the implementation stage, such as extra redundancy, and added latency as a result of o packet processing. There were very few works focussed on the impact of redundancy and the corresponding impact of the number of packets combined, which is known as generation size as authors in [95] claimed. Authors in [95] introduced a simple mathematical framework for optimising the sending rate for TCP with network coding and the generation size at the sources for multiple flows. They developed a modified NUM issue by proposing accumulated utility function that considered both the sender rate for every flow and the generation size. Authors in [95] noted that there was a need to include inter-session network coding in case of multiple flows, along with avoiding Ack compression by preventing the queuing of ACK packets. They anticipated doing such idea would improve the throughput and reduced number of transmission in case of inter-session network coding.

Work in [96] came up with a new network layer, that fitted below the transport layer and it masked the non-congestion occurs in packet losses in TCP. The network coding was equipped with the well-known class of Maximum Distance Separable (MDS) codes. It was done by encoding a set of k TCP segments into more bigger set n of encoded network coding segments and consequently broadcasting set. The destination would be able to reassemble all n segments by receiving the subset of k by using the MDS property. Results proved that the throughput was better compared to TCP if the packet erasure probability was abundantly substantial. Authors in [96] suggested to a product of symbol length and dimension must meet the average TCP segment size to evade padding. Secondly, the scheme should be adjunct to meet channel conditions for instance if the code rate was high (code rate $R = 1$ means no encoding) are applied at low packet erasure, otherwise smaller code rates are applied in high erasure channel.

TCP-HONC was proposed by authors in [97] to improve throughput and optimise end-to-end delay for TCP with network coding in Multi-Radio Multi-Channel Wireless Mesh Networks (MRMC WMN). It was implemented by using linear network coding and, it applied hop oriented network coding for TCP flows given that the local information of each node. TCP-HONC modified the number of available encoding blocks and scheduled packets on each hop. TCP-HONC used distributed rate control and path selection algorithm based on back-pressure approach, which utilised network improved and end-to-end delay. TCP-HONC overtook TCP/NC [86] end-to-end by 236% and TCP/NC hop-by-hop by 143% in term of throughput and deducted the average end-to-end delay up to 27% and 90% in single path scenario. Moreover, TCP-HONC gained a total throughput of 347% in a multi-path scenario, when the packet loss rate on every link was 35%. Delay and jitters of TCP-HONC were small in the presence of high loss rate. TCP-HONC showed good fairness and balance load from congested paths to non-congested paths, and the queue length on the bottleneck link was stable. Recoding data at relay nodes drove these benefits and enhanced delay reduction from one hop Ack and retransmission scheme.

The work in [98] introduced a network coding model based on the use of triangular coefficient matrices for packet generations which would get rid of the coding delay. Authors in [98] scheme could reach a high throughput even if the sender knows the decoding rate at the receiver. The proposed scheme was compared with square matrix coding [99] and online coding in [100] , it outplayed both square matrix and online coding. Furthermore, the proposed scheme reached a higher throughput in TCP than online coding in the multi-hops scenario in an elusive environment.

In this work [101], authors concerned with the poor behaviour of TCP in lossy bursty behaviour in case of indoor scenarios. They used the Kodo network coding implementation and integrated it with the TCP. Authors in [101] found that network coding could hide the impairments of the wireless channels. Although, retransmission scheme in IEEE 802.11 MAC could do the same result. However, network coding can achieve better performance. In case if losses were bursty, MAC layer retransmissions were not sufficient. The proposed TCP-NC was better and much stable. For example, the gain throughput of TCP-NC was up to 350% in low-quality links (FER equals 0.3).

Authors in [102] argued that Transmission Control Protocol with Network Coding (TCP/NC) used additional sub-layer called Network Coding layer below TCP layer to handle packet losses without sensed by TCP layer. The authors introduced some variants of TCP/NC such as TCP/NCwLRLBE (TCP/NC with Loss Rate and Loss Burstiness Estimation) which can improve the retransmission and adapt to the changing of the channel.However, most versions of TCP/NC did not consider two problems, that are the bi-directional packet loss and the unordering packet receiving, which affect the good-put performance seriously. Therefore, Authors in [102] examined the goodput performance degradation by the bi-directional packet loss and the unordering packet receiving, and proposed a solution. The result of their simulation on ns-3 confirmed that the proposed scheme could work well when loss happens in both directions as well as in unordering packet receiving environment compared to the TCP New Reno and their proposed protocol, TCP/NcwLRLBE.

Again, authors in [103] studied the characteristics of the TCP/NC tunnel on heterogeneous networks, eventually pointing at the TCP/NC tunnel-based (intelligent gateway) system distributed over IoT environments. Scholars in [103] proved that the TCP/NC tunnel could efficiently utilise the bottleneck bandwidth even with congestion and achieved a significantly high goodput of E2E-TCP sessions in a wide range of link loss degree mainly when the tunnel link bandwidth was sufficient.

Authors in [104] presented TCP-VON which stands for TCP Vegas with online network coding. It can be decoded progressively instead of generation to generation. Using online network coding, it could enhance the throughput and reliability of the end-to-end communication, and online network coding can lower decoding delay. In TCP-VON, the sender transmits redundant encoded packets when the sender senses packet losses from ACK messages. Alternatively, it transmits innovative encoded packets. In conclusion, authors [104] compared the delay and throughput of TCP-VON and automatic repeat request network coding based TCP (TCP ARQNC). Their results proved that TCP VON outran TCP ARQNC regarding the average decoding delay and network throughput.

The work in [105] implemented CoMP, a network coding multipath forwarding scheme which enhanced the reliability and the performance of TCP flows in wireless mesh networks. CoMP applied both congestion control and rate control at which linear combinations were transmitted. CoMP anticipated the losses and transmitted redundant linear combinations. It helps to make decoding delay at the minimum level, and it forestalls TCP timeouts and retransmissions. Furthermore, CoMP accomplished a higher throughput. It maintains TCP flows feasible for wireless mesh networks under the heavy losses scenario.

Authors in [106] took on the challenge that TCP faces because of random losses and ACK message interference. They applied the use of interflow coding between data and ACK packets to reduce the number of transmissions among nodes. Authors in [106] took advantage of 'pipeline' random linear coding approaches with adaptive redundancy to mask the high probability of lost packets on weak links. As a result, ComboCoding is introduced, which merges intraflow and interflow coding to TCP transmission reliability against error-prone wireless networks. ComboCoding achieved 2 Mbps goodput with 30% per link RLP; meanwhile TCP New Reno with no coding achieved only 200 Kbps. Compared to the original pipeline coding, ComboCoding reduced transmission overhead by 30% under perfect link conditions and by 10% overhead in most tested hops.

The work in [107] focused on TCP over mobile ad-hoc networks. Authors proposed a network-coded multipath scheme for traditional TCP-CodeMP that adapted frequent link changes in ad-hoc networks and there was no need for explicit control messages. TCP-CodeMP has three components: first, the random linear coding process works with adjustable redundancy; second, multipath routing; and third, ACK Piggy coding. Their results were presented in a three-hop static scenario. The TCP-CodeMP improved TCP flow throughput to 70%. Moreover, CodeMP recorded at least 700 Kbps aggregated TCP throughput, also it recorded 0.99 degrees on Jain's fairness index. Researchers in [107] used an extreme scenario which consisted of two TCP flows, and nodes that were moving at 25 m/s with a packet error rate that reached 40%.

In order to achieve higher TCP throughput in bidirectional two-hop wireless networks, authors [108] proposed a new acknowledged packet transmission model in IEEE802.11a. The proposed model took advantage of the sub-carriers which transmit zero padding bits added to the MAC layer protocol data unit (MPDU). The relay node is in charge to identify which node transmits ACK packet (native) or the Xored packet. Authors in [108] claimed that they could improve throughput by 3.4 Mbps.

Recently, authors in [109] manipulated the capabilities of Software-Defined Networking (SDN) to measure packet losses. Subsequently, the proposed solution coded UDP and TCP traffic using RLNC with adaptive redundancy, to guarantee a specific QoS. The authors in [109] introduced a new approach to calculate the redundancy accurately for a specific decoding (or delivery) probability per packet. Furthermore, authors in [109] proposed to prediction interval to evaluate past measured loss ratios and estimate future ones. Their resutls showed that packet losses could be measured with a certain probability, with a maximum deviation of 3%. Besides, they demonstrated that the adaptive redundancy allowed to achieve a delivery probability not deviating more than 1.5% from the desired one. lastly, their solution could enable TCP to attain a stable throughput.

2.3 Cross-layer Xor Network Coding on TCP Protocol

The work in [110] focused on unicast flows with intersession network coding, as in COPE [12]. Their NCAQM was queue management that mimicked the model of the optimal solution as the authors claimed. NCAQM was implemented at relay nodes. The study in [110] showed that TCP over NCAQM performed significantly better than TCP over COPE. The gut reaction was to eliminate the rate mismatch between flows which were encoded together through a synergy of rate control and queue management. They developed the NUM (Network Utility Maximisation) problem and derived a distributed solution. Authors in [110] proposed congestion control for unicast flows by applying a network utility maximisation problem. Their results were compared to flows with network coding. For instance, in two hops, the optimal solution was 33%, TCP+NCAQM was 27% and TCP+COPE was 12%. In Cross Topology, optimal was 60%, TCP+NCAQM was 31% and TCP+COPE was 16%. Lastly, in X topology, optimal was 33%, TCP+NCAQM was 22% and TCP+COPE was 10%.

In this work [111], authors were interested in improving the performance of network coding in lossy wireless environments using COPE in [12]. They proposed I2NC which combines intersession and intrasession network coding and has two sharp points. The error-correcting capabilities of intrasession network coding makes the scheme resilient to losses. Lastly, redundancy allows relay nodes to operate without knowledge of the decoding buffers of their neighbouring nodes. Again, authors in [111] applied NUM formulation of the problem. Moreover, authors in [111] proposed two practical schemes: I2NC-state (intrasession for error correction) and I2NC-stateless (intersession for redundancy), which mimic the structure of the NUM-optimal solution. They addressed the interaction of their approach with the transport layer. They showed the benefits through simulation in GloMoSim.

Authors in [112] presented flexible network coding. They focused on both interflow and intraflow coding approaches, where both the sender and relay nodes encode packets linearly from the same flow. Authors in [112] used TCP traffic in interflow with Xor network coding, and UDP is used with intraflow by applying RLNC. Intraflow illustrated substantial performance improvement by capitalising on combined UDP with the linear coding approach. Authors in [112] claimed that they reached enhancements of 73% compared to TCP by allowing intermediate nodes to recode the packets. In the interflow scheme, TCP segments are combined from different flows at relay nodes, and as a result, the overall throughput increased by 22%.

Recent study concerning the poor performance of TCP in multi-hop wireless. Authors in [113] investigated to what extent network coding could help to improve the throughput in a chain topology multi-hop wireless network. Their work focused on IEEE 802.11's DCF, to perform distributed packet scheduling. Moreover, the reverse flowing TCP ACKs were xored with forward flowing TCP data packets. Without any modification to the MAC protocol, the gain from network coding was negligible. The inherent coordination problem of carrier sensing based random access in multi-hop wireless networks dominated the performance. They provided a theoretical analysis that yielded a throughput bound with network coding. Lastly, scholars in [113] introduced a modification to the IEEE 802.11 DCF, based on tuning the back-off mechanism using a feedback approach. As a result, the performance of TCP sessions was improved by more than 100%.

2.4 Cross-layer Random Linear Network Coding on TCP Protocol

The work in [114] was concerned with finding the best optimisation decision in case of full network or empty packet queues at the network, where cross-layer interaction solutions were implemented between MAC and network layers. Authors in [114] focused on the MAC layer nature and network layer where network coding processes exist. They analysed performance metrics such as maximum throughput, number of transmission and coding packets and energy costs. They concluded that the resulting optimisation trade-offs depend on cross-layer interactions between MAC and network layers, and the optimal network operation depends on the cooperation of nodes in the network.

PiggyCode was introduced by [115], a network-coding based on enhancing TCP performance in 802.11 Multi-Hop wireless network. The core of this study was a network coding scheme operated between the network and the MAC layer. The coding process was similar to piggybacking the TCP-ACK packet within the TCP-DATA packet – the only change was that network coding processes did not change packet size. Moreover, PiggyCode encoded TCP-ACK and TCP-DATA packets together, and as a result it decreased the overall utilisation of the network and improved TCP performance in the wireless networks. In grid topology, PiggyCode achieved a gain of up to 16% in throughput and reduced the average time needed to deliver TCP-ACK packets by up to a factor of ten.

2.5 Summary

Since its emergence, network coding has caught the attention of many scholars. In the past, the fundamental theory of network coding has been progressing significantly. COPE [12] was implemented to explore the possibilities of network coding. The state-of-the-art network coding within wireless Multi-Hop networks has been explored. The capability of network coding in the wireless environment has been exploited to take advantage of the broadcast nature of the wireless medium. COPE and its variants in terms of Xor and RLNC evidence this. However, studies have tried to investigate the possibility of including the functionalities of network coding in the OSI network stack. Related works on network coding in the MAC and transport layer have been presented. The challenges of incorporating the functionalities of network coding in the MAC layer are still present. The role of relay nodes adds some complexity which would restrict the ability of network coding to fulfil its potential.

In the cases of Xor network coding at the transport layer, the network coding struggles to improve network throughput. The role of relay nodes has not been acknowledged yet by TCP protocol. Thus, the need to look into the MAC and transport layer has become a necessity. Researchers in [7] has mentioned that the network throughput can reach more than two-fold. This was widely believed and the maximum throughput gain has been taken as two-fold as in [12]. This thesis studies the challenges and possibilities of Xor network coding in Multi-Hop wireless networks, and also investigates how this can be applied in harmony with MAC and transport layer. The thesis focuses on Xor network coding. It is a particular case of random network coding using Galois Files of size 2. In the Xor network coding, packets are encoded by applying a bitwise Xor operation. In thesis, Xor network coding is chosen, and COPE [12] is installed between the MAC layer and the network layer. COPE is a suitable match for our investigation at the MAC and transport layer. Besides, COPE can use both UDP and TCP traffic. Moreover, COPE allows two layers to collaborate in the process of Xor network coding. Such collaboration can improve the overall performance of Xor network coding in the multiple TCP traffic cases.

As mentioned above, the MAC layer functionality could affect the role of relay nodes and adds some complexity which would restrict the full potential of Xor network coding. Besides the applied techniques in the literature did not consider the drawback of such approach, namely MAC priority techniques. Chapter 3 will investigate such technique to determines to what degree Xor network coding using MAC priority approach could maximise the throughput in Multi-Hop networks. At transport layer,the role of relay nodes has not been acknowledged yet by TCP protocol. Furthermore, TCP throughput has not remarkably improved. To find why and how throughput and reliability could be enhanced, this thesis investigates different TCP versions such as Tahoe, New Reno and Westwood. Chapter 4 is dedicated to explore ways to excel TCP while Xor network coding is applied.

If the MAC and Transport layer presents a challenge to Xor network coding, could cross-layer design provide ideal solutions to reach ultimate performance? Chapter 5 investigates the possibilities and challenges of Xor network coding in Multi-Hop wireless networks and how it can work with cross-layer design, that is cross-layer between MAC and transport layers.

CHAPTER 3

IMPROVING NETWORK CODING THROUGHPUT VIA MAC LAYER

As discussed in chapter 2, scholars in [7] indicated that the maximum throughput gain that was obtained using network coding (in wireless scenario) was only two-fold. This chapter aims to achieve higher (than the two-fold) throughput with better delay and jitters performances. Wireless Multi-Hop networks often rely on the use of IEEE 802.11 technology. However, the Distributed Coordination Function (DCF) of IEEE 802.11 is not optimal in various networks. For example, the DCF performance drops when a network coding is applied. It is, therefore, needed to improve the performance so that network coding technique can be used. The DCF is fairer to nodes than performing network coding process. This is because the relay nodes in Multi-Hop scenario have urgent requirements to gain more access and regulated access, which accommodate the network coding process.

3.1 Introduction

S.Katti *et al* in [12] have implemented COPE, a new architecture for wireless mesh networks. The work was implemented between network and MAC layers, but it is mainly focused on the practicability of network coding in the network layer, where a decision is taken for the destination path. The work in [12] has put the theoretical studies on network coding into practice, and provided a significant performance improvement, and as such it has drawn the attention of many researchers. However, the MAC layer was not a major concern in the implementation of this work.

COPE with MAC-layer priority was introduced in [24]. It uses contention window (CW) in the MAC layer and considered as an improvement of COPE in [12]. This could provide greater access to the relay (middle) node that is responsible for encoding and broadcasting of coded packets.

The work in [34] used the same concept of MAC priority, but was based on the ratio of incoming and outgoing packets queued at the middle node. Therefore, the middle node could adjust its and other nodes' CW according to the table of ratios and the corresponding CW values. Other works in [1], [41], [116] and [50] followed the same concept from different angles and reached different solutions. For example, authors in [116] considered how long-term and short-term channel asymmetries could be used to reach the highest throughput. Moreover, scholars in [50] proposed the BEND algorithm to create more chances of mixing packets in a queue in a specific topology, while work in [24] allocated bandwidth on the MAC layer in a two-hop network through the use of the virtual queue.

The proposed algorithm adopts the role of the relay nodes in Multi-Hop scenarios, hence it will be the main player of encoding and transmitting coded packets. Therefore the DCF mechanism is altered to give nodes a fair chance to the medium. Moreover, it pays greater attention to relay nodes as it is used as means of coding decision centres and improving overall throughput. Hence, the algorithm is speculatively assumed to improve the throughput of COPE in Multi-Hop scenarios, and it is capable of outlining the importance of algorithms to substitute the default contention window (CW) calculation in the MAC layer in 802.11 standards. Correctly, this proposed solution is assumed to give nodes a fair share of medium according to their importance in contributing in the network coding process. As a result, the overall total throughput is improved: our simulation recorded a significant increase in throughput in various Multi-Hops scenarios.

Overall, fundamental concerns about the proposed algorithm and its assumptions that make the method work are as follows: COPE: has very low throughput, it does not reach the maximum throughput and COPE does not use medium efficiently. COPE with MAC priority, which is the improved version of COPE, can reach a good level of throughput but the number of coded packets is lower, or none at all. Therefore, the proposed algorithm gives middle nodes a greater share of the medium to improve the total of throughput. COPE with MAC-layer priority performs similarly.

3.2 Problem Statement

The default CW calculation causes long waiting times, and consequently, low throughput are achieved for contenting nodes. As shown in figure 3.1, every time the station fails, it cumulatively doubles the CW values and causes more delays. The accumulated throughput is needs to be maximised. In order to meet this requirement, this work examines how the medium behaves when there is no priority to the relay nodes – for instance nodes in the two-hop topology. Therefore, great scrutiny on the whole simulation is needed to find out why the relay nodes could not get more access to the medium, and why there were gaps of silence during the whole process which was calculated to be more than 33.35%, as figure 3.1 depicts.

The relay node has gained 38.88% over the two edge nodes. It is clear there is an unexplained

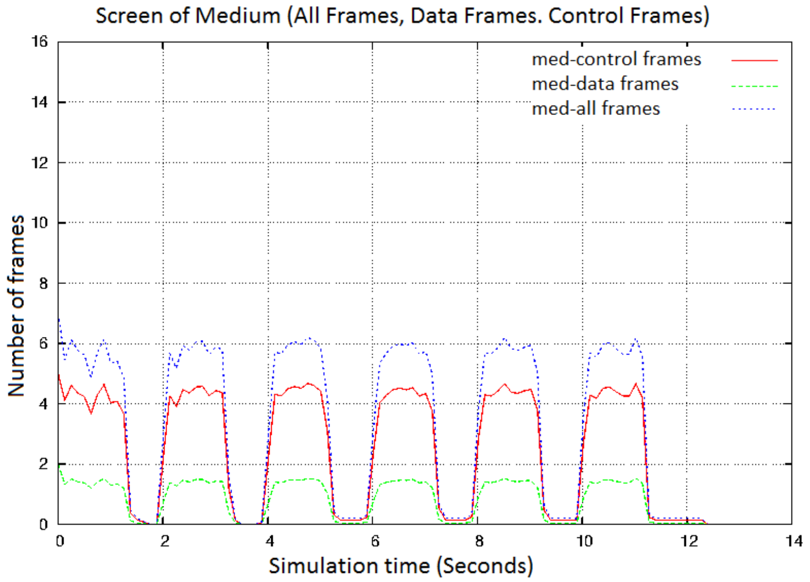


Figure 3.1: Screen on the medium in the case of two hops (all frames, data frames, control frames).

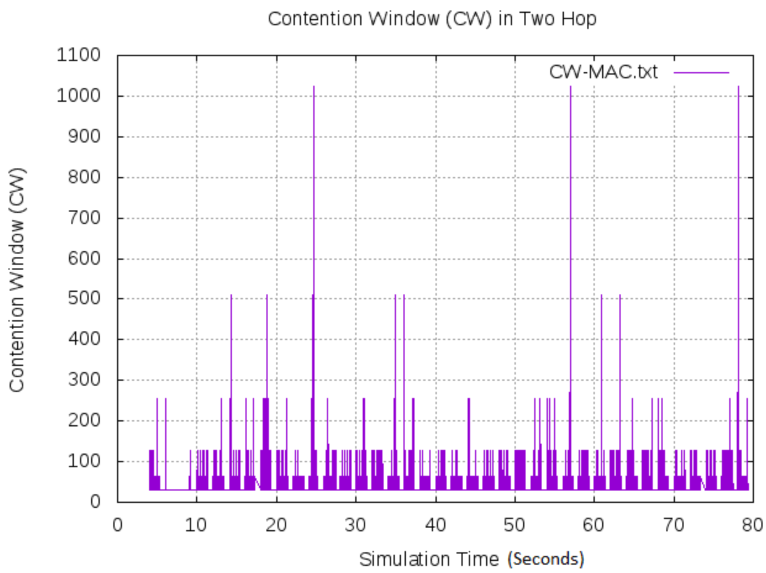


Figure 3.2: Screen on Contention Window in a case of two hops.

issue with the number of encoded packets, whether full utilisation of medium or higher throughput can be achieved. From this point, there was no priority to nodes to gain more access than edge nodes. The results are similar to the first implementation to COPE in [12]. Moreover, the relay node had equal chances to medium. The CW for the relay node had higher values and longer waiting times, which affected total throughput. On the other hand, the edge nodes had similar CW values, which were large. Therefore, the CW had a negative impact on edge nodes: it

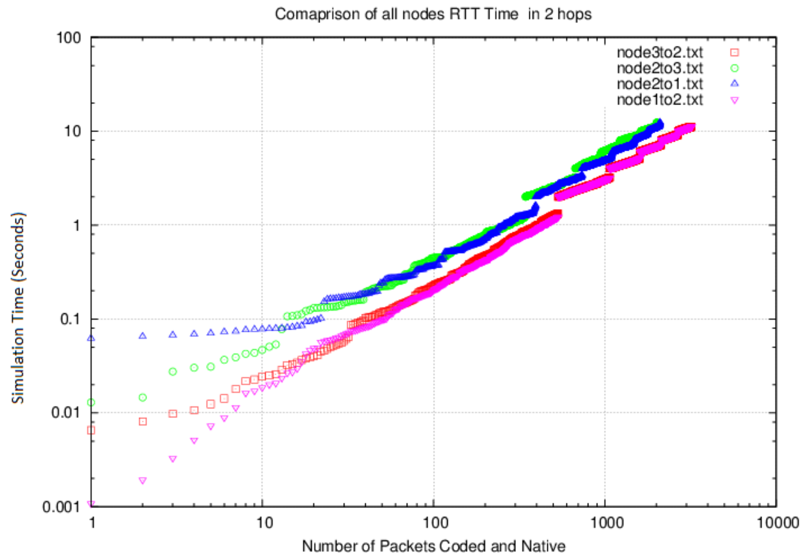


Figure 3.3: Comparison of round-trip time in a two-hop case.

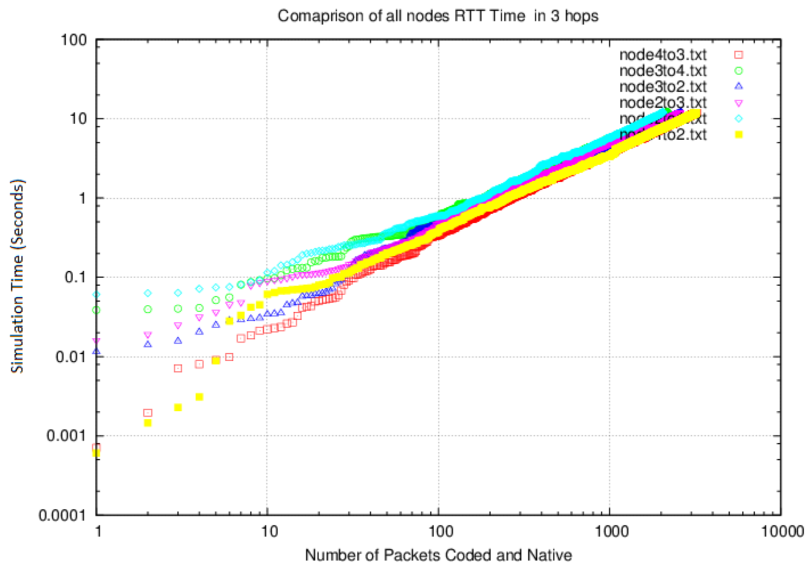


Figure 3.4: Comparison of round-trip time in a three-hop case.

constrained their access to the medium. In fact, the round-trip time increased rapidly without MAC priority towards the relay node, as in figure 3.3 and 3.4, which depict the RTT for two and three hops respectively. Meanwhile, Figure 3.2 shows the CW behaviour in a two-hop network, and it is in concordance with figure 3.1. This proves the inefficacy of the current CW in DCF in the nodes of our simulation. Figure 3.2 depicts the gap of silence in the network medium. It is due to higher CW values that push away contented nodes fairly in the network. However, what if the relay node ceases this gap of no transmission and sends its already-encoded packets? The RTTs in the second and third hops have increased linearly, especially the CW calculation. In conclusion, CW doubles intensively at relay nodes and among other nodes in networks, causing drains in overall throughput and long delays.

We observed that CW was fast built-up, it was due to when the waiting times uniformly spread the backlogged traffic over a larger time frame. In the case of Multi-hop scenario, the rapid build-up of waiting time along with an increasing number of collisions on the medium, as a result, the throughput is degraded, and overall performance is jeopardised as the nodes are waiting for the duration of waiting time which varied exponentially with the binary base. The need to address this issue in Multi-hop networks where Xor network coding is applied, become a necessity and a pressing issue. Therefore, we are analysing the current contention window for better outcome favouring Xor operation in network coding in the multi-hop scenario.

3.3 Proposed Solution

The proposed algorithm consists of three phases. Figure 3.9 explains the logic steps of the algorithm, and figure 3.10 illustrates where the proposed algorithm is positioned against the default CW of DCF function in the MAC layer:

- Phase 1: The contending stations have to choose initial values based on the proposed algorithm's calculation of minimum and maximum CW value related to the number of competing stations (i). For example, if the number of hops is more than two, the initial value can be calculated as:

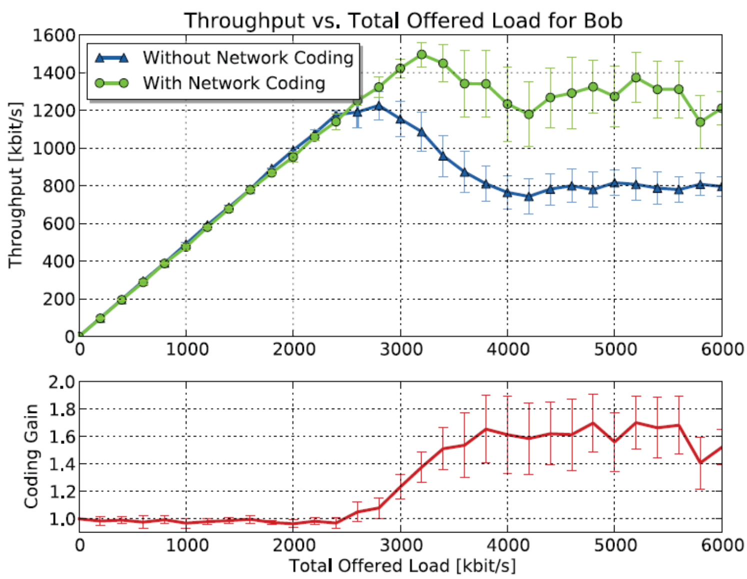


Figure 3.5: COPE, COPE-MAC-priorit (two-hop) [1].

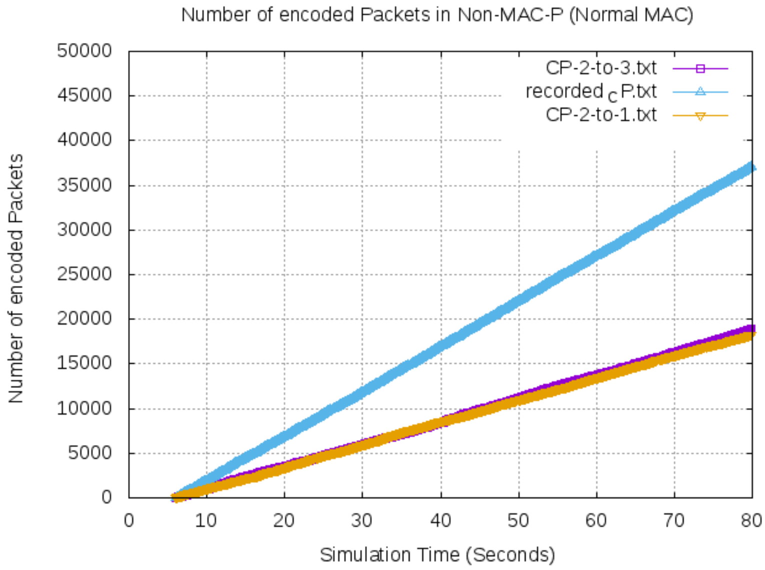


Figure 3.6: Coded packets over time with few chances in network (COPE with Normal MAC).

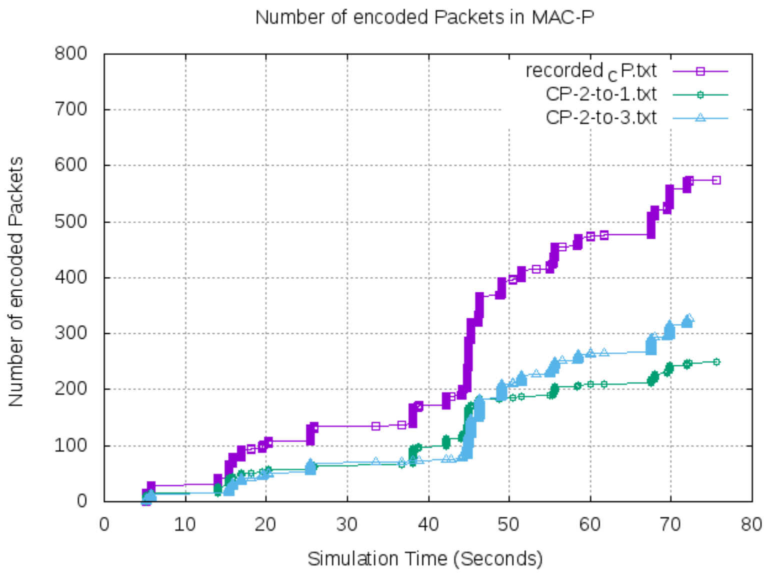


Figure 3.7: Coded packets over time with few chances in network (COPE with MAC priority).

$$CW = 0.618 * (MIN\_CW + Node_i). \tag{3.1}$$

In the above equation, 0.618 was obtained after dividing the success CW value by the sum of fail and success CW value.To illustrate this, suppose we take all values and divide on success value. There is one unique observation, at which the ratio of the larger values of

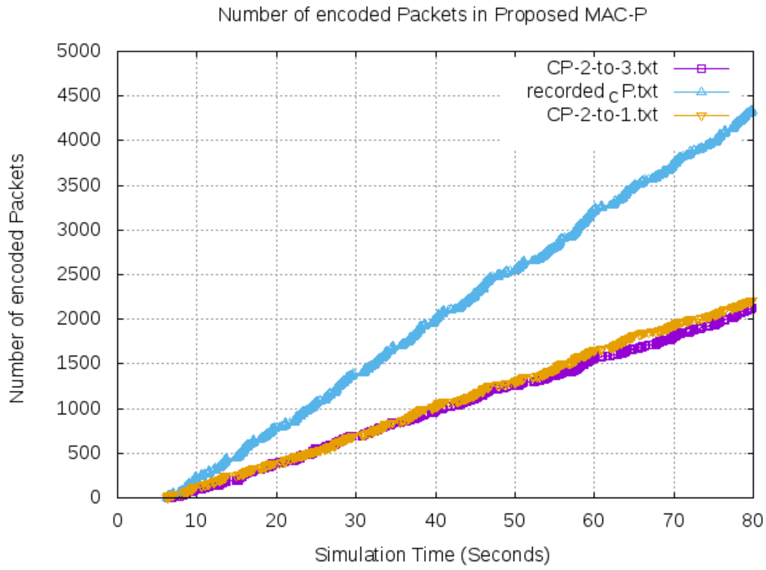


Figure 3.8: Coded packets over time with more chances in network (proposed algorithm).

CW to the recently succeeded CW values was exactly the same as the ratio of the whole values of failed and success. The value can be seen as a result of the Fibonacci sequence. The series is quite simple,it starts with 1 and adds 1 to get 2. It then repeats the process of adding every two numbers in the series to determine the next one, for example 1, 2, 3, 5, 8, 13, 21, 34, 55, 894, 55, 89 etc.

- Phase 2: If the contending station fails to gain access, the proposed algorithm scheme determines the best CW value. This is done according to the following:

  1. The probability of success will be calculated based on the following equation:

     $$Ps = 1 - f/(f + S) \quad [117], \tag{3.2}$$

     where f and S are the number of failed attempts on MAC to access a medium and the number of successful transmissions on MAC, respectively.

  2. The average waiting time is calculated by dividing the current CW by two and multiplying by the number of backoff slots.

     $$AverageWaitingTime = \frac{CW\_Normal\_Calculation}{2} \times BackoffSlots \quad [117]. \tag{3.3}$$

  3. The new CW is recalculated by multiplying the last known CW with the ratio of 0.618 to grow gradually. The reason to choose the aforementioned value of 0.618, is driven by the dividing the previous contention window value (usually bigger value) with recent failed value.

$$NCW = 0.618 \times Last\_CW\_Calc. \tag{3.4}$$

4. The average updated CW value is taken for further assignment.

$$AverageCW = \left(\frac{CW + Last\_CW\_Calc}{2}\right). \tag{3.5}$$

5. The difference of the current calculated CW in DCF and the proposed new CW is calculated as:

$$Diff = CW\_Normal\_Calculation - NCW. \tag{3.6}$$

6. The optimal CW (OptimalCW) value is 3.

- Phase 3: When the probability of success is less than 90%, a new assignment to the failed CW is calculated as follows:

$$CW = 0.618 \times (Diff + OptimalCW). \tag{3.7}$$

  If it is less than 90%, the CW is calculated differently as follows:

$$CW = 0.618 \times (AverageCW + OptimalCW). \tag{3.8}$$

  The results show that the best throughput is achieved when the probability of success is over 90%.

3.3.1 IEEE 802.11 Protocols

3.3.1.1 IEEE802.11a

802.11a is one of the first two extension to the original 802.11 standard. It supports transmission speed up to 54 Mbps using signals in the regulated frequency spectrum. It has higher operating frequency compared to 802.11b, which has a limiting impact on the signal's ability of penetrating walls and other obstructions, and range coverage [118]. Additionally, 802.11a is incompatible with 802.11b.

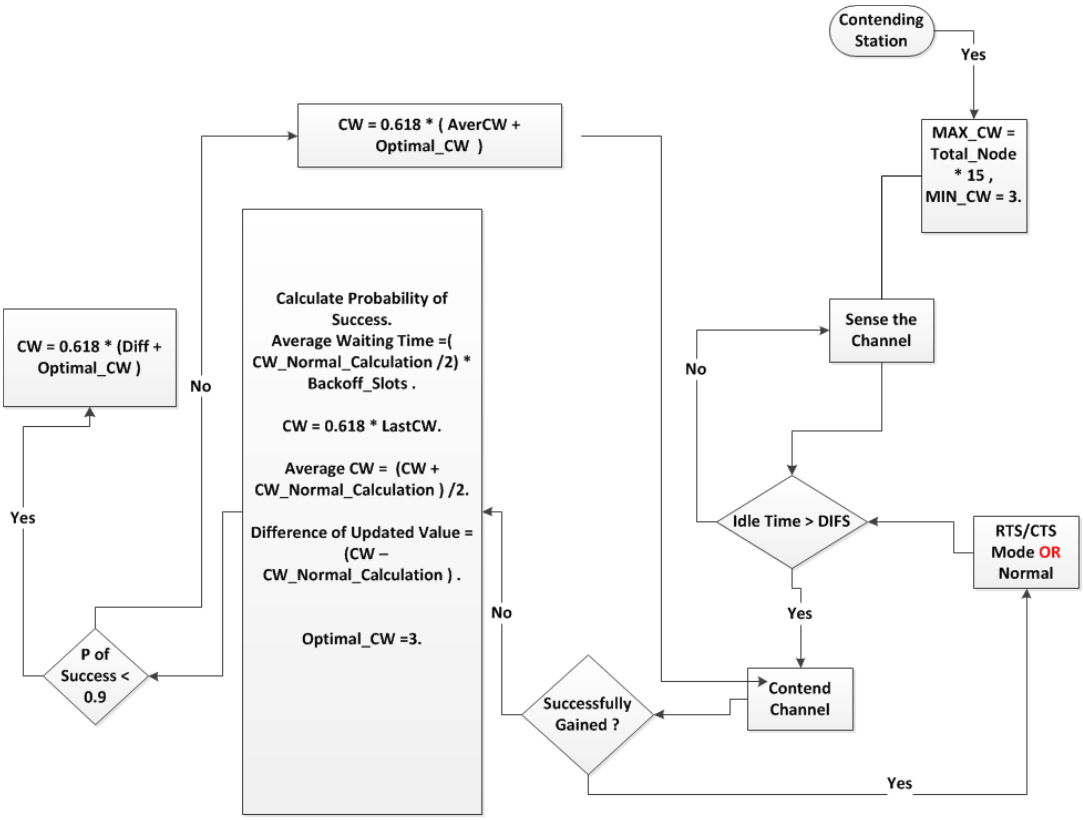


Figure 3.9: The proposed contention window algorithm at MAC layer.

3.3.1.2 IEEE802.11b

IEEE expanded on the original 802.11 standard in July 1999, creating the 802.11b specification. 802.11b supports a theoretical speed up to 11 Mbps. A more realistic bandwidth of 5.9 Mbps (TCP) and 7.1 Mbps (UDP) should be expected. 802.11b uses the same unregulated radio signalling frequency (2.4 GHz) as the original 802.11 standard. Vendors often prefer using these frequencies to lower their production costs. Being unregulated, 802.11b gear can incur interference from microwave ovens, cordless phones, and other appliances using the same 2.4 GHz range [118]. However, by installing 802.11b gear a reasonable distance from other appliances, interference can easily be avoided [118].

3.3.1.3 IEEE 802.11g and IEEE 802.11n

802.11g is the WLAN standard currently widely supported by wireless devices and network equipment. Advantages of 802.11g include:

- Transmission speed up to 54 Mbps.
- Uses 2.4 GHz unlicensed frequency bands for greater range.

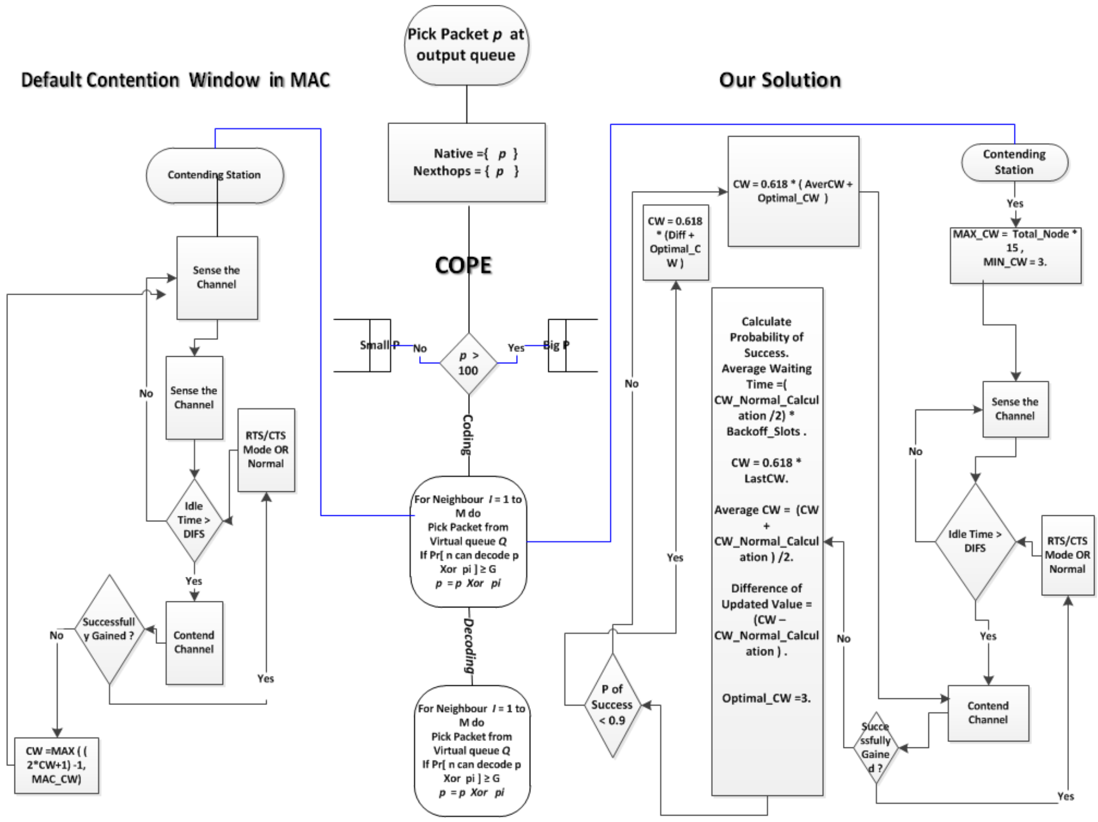


Figure 3.10: Position of the proposed contention window algorithm.

- Combines the best features of 802.11a and 802.11b.
- Backward compatibility with 802.11b which allows to work with 802.11b wireless network adapters.

802.11g was upgraded to 802.11n to enhance the transmission speed. This was achieved using MIMO technology that was implemented using two or more antennas. In 2009, up to 300 Mbps transmission speed was reported using 802.11n. The IEEE 802.11n has a better quality of service (QoS) and backward compatibility with all the legacy (IEEE 802.11a/b/g) devices. In this simulation, 802.11g and 802.11n have been used. We used ErpOfdm Physical layer mode [119] with different data rates range from 2 to 24 Mbps. We also influence the quality of service (QoS) to implement our solution.

3.4 Performance Metric and Simulation Settings

In order to evaluate the proposed solution performance, the following metrics must be considered:

NS3 settings	Value
Number of Packets	1000000
Distance between nodes	250 m
Running Time	300 Seconds
Number of Runs	10 times
MAC layer Queue Capacity	10 frames
Test Protocol	UDP

Table 3.1: Simulation settings.

1. Throughput: this is the number of packets successfully received by the destination with respect to time (bits/s)

$$T = \frac{Number\ of Received\ packets \times PacketSize \times 8}{Simulation\ time\ (Seconds)} \tag{3.9}$$

2. RLP: this is the ratio of the number of packets lost and the total number of packets sent on the medium.

$$RLP = \frac{The\ number\ of\ packets\ lost}{The\ total\ number\ of\ packets\ sent} \tag{3.10}$$

3. Delay: this refers to the amount of time it takes a packet to be transmitted from source to destination.

4. Jitters: these are defined to be a variation in the delay of received packets.

3.4.1 Simulation Settings

The simulation was conducted using the Network Simulator 3 package (version ns.3.27). COPE was implemented and added to NS3 source files, and the IEEE 802.11 DCF with CSMA/ CA was adopted as the MAC layer protocol. All nodes have their own radio transceiver. The MAC-layer FIFO transmission queue of each radio transceiver is set to 64 packets. Various Physical layer modes were used during the simulation.

Fixed Received Signal Strength (RSS) model [120] was applied during the simulation. In this model type, the received power is fixed to a predefined value, regardless of the distance. The transmit power was set at -80 dBm. The number of packets transmitted for each flow was over 1 million. The packets size was varied from 100 to 1500 bytes, and a 20 Mbps data rate was used. For each simulation scenario, 3 to 14 nodes were placed in the chain topology. CBR traffic flows were injected into the networks from the sources. Friis propagation loss model [121] was used to estimate the signal power received by the receiver. The path loss exponent, which is a parameter of the shadowing model, was set to 4. The Wifi channel in ns3 was set to constant propagation delay model. The MAC layer was set to default without the quality of services. Moreover, the MAC type was on Ad hoc mode. All nodes use Constant Position Mobility Mode. Hence, all nodes are set in stationary positions on the edge of each other. Static and dynamic routing tables were used at the network layer, this includes Ad-hoc On-Demand Distance Vector (AODV) routing [122] which is a routing protocol for mobile ad-hoc networks.

Every node handles a set of independent ns3::DcfState, each represents a single DCF within a MAC stack. In addition, each DcfState has a priority implicitly associated with it. The priority is determined when the ns3::DcfState is added to the DcfManager. The first added DcfState gets the highest priority and the second gets the second highest priority. This helps to handle the "internal" collisions. When two local DcfState try to get access to the medium at the same time, the one with highest priority local DcfState accesses the medium, and the other faces an "internal" collision. Moreover, nodes are assigned different CW. Each node is assigned a maximum and a minimum value for its own CW using DCA. DCA class implements the packet fragmentation and re-transmission policy. It uses the ns3::MacLow and ns3::DcfManager helper classes to respectively send packets and to decide when to send them. Packets are stored in ns3::WifiMacQueue until relay nodes can send them in the topology.

For every simulation run, the seeds are configured by changing global seed. These values are stored in two $ns3::GlobalValue$ instances: $g\_rngSeed$ and g_rngRun in deep ns class of randomness function. The class $ns3::RngSeedManager$ provides an API to control the seeding and run number behavior. This seeding and substream state setting must be called before any random variables are created; for example: $RngSeedManager::SetSeed(15)$; // changes seed from default of 1 to 15, and $RngSeedManager::SetRun(200)$; // changes run number from default of 1 to 200.

In the simulation, UDP traffic was generated using ns3 application layer. We used $ns3::OnOffApplicationClass$, which generate traffic to a single destination. This traffic generator follows an On/Off pattern: after Application::StartApplication is called, "On" and "Off" states alternate. The duration of each of these states is determined with the onTime and the offTime random variables. During the "Off" state, no traffic is generated. During the "On" state, cbr traffic is generated. The cbr traffic is characterised by the specified "data rate" and "packet size". The topology used consisted of multi-hop with two flows in reverse direction, figure 1.3 in chapter 1 depicts the used topology. Other settings are summarised in table 3.1.

The simulation results were collected using ns3 flow statistic tool. The above settings have been applied to all scenarios. The simulation results were recorded in a spreadsheet, and these data were plotted in graphs using QiPlot and MATLAB. The graphs show throughput, packet loss ratio, delay and jitters values, and packets numbers (received and transmitted).

3.5 Results and Discussion

3.5.1 Results

3.5.2 Two Hops

The proposed solution has a substantial impact on medium sharing. It uses a new contention window calculation in DCF. The proposed solution favours relay nodes and gives them small values of CW to accommodate the middle node to send encoded packets in the waiting queue. A collision event was observed only when a contending node fails to gain access to the medium and backs off. When this happens, a new calculation, like in the stepwise criteria, is performed to find the next best CW value as in section 3.3. The proposed algorithm is tested using increased loads (data rates, packet size, flows) on different modes of MAC layers, and the results are compared with 'COPE with MAC priority' found in [24], [1], [34] , [41], [116], [50] and COPE in [12]. The results are far better than COPE in [12], which affords no priority to relay nodes. Moreover, the proposed algorithm reached the same highest throughput level as in COPE with MAC priority found in [24], [1], [34], [41], [116], [50]. Specifically, figure 3.11 showed that the proposed algorithm produced higher and steady throughput with increased offered load (this refers to the total traffic load which includes packet size, number of flows, and data rates). Figure 3.5 shows the results compared with the work in the [1], as can be seen the results match in the topology of two hops. Figure 3.11 shows the box-plots of throughput results compared with MAC priority and normal MAC using COPE in [12]. When the offered load is increased, the proposed solution could reach almost 7 Mbps throughput with an average throughput of 6.8 Mbps against MAC priority techniques. The average of normal MAC using COPE [12] was around 2.3 Mbps of throughput. The box-plots in figure 3.11 looks comparatively short, which indicates insignificant differences. However box-plots of normal COPE are little wide and throughput values slightly skewed but with acceptable range. The highest throughput value was recorded in the case of our proposed MAC-P at 7 Mbps, whereas in the case of MAC-P the throughput was around 6 Mbps. The reason why the proposed MAC-P had better performance was because of its a new calculation for failed CW.

The authors in [1] did not mention any changes in DCF's CW. Moreover, when we evaluate the concept of the MAC-priority solution, it showed few or no coded packets. Unlike the works in [24], [1], [34], [41], [116] and [50], the proposed solution in this work functions well with more than two hops. On the other hand, figure 3.7 represents the cost of using COPE with MAC priority and how it affects the process of coding since the relay nodes between two destinations will not get any, or only a few, coded packets. This is because there are a few chances for the middle node to find a matching pair of packets to encode from different flows. Figure 3.6 showed that the number of coded packets sent to the correspondent nodes is much higher than the MAC propriety: it was more than 35,000 packets in total, and did not contribute to higher throughput than MAC priority, this is shown in figure 3.11. The total number of coded packets in MAC priority was less than 600 packets. Meanwhile, the proposed MAC-P was nearly 4,500 packets. The results in figure 3.8 also show that the average throughput when compared with MAC priority was 29 Mbps. MAC with priority has the gap of silence that was between 20 to 40 and 60 to 70 second of simulation time (figure 3.7). Hence, the relay node forwards any packets to the next hop if there is no matching pair to be coded. It meets the requirement of the original COPE implementation, as first introduced in [12]: the use of packet reception reports indicates coding opportunities at each node. These reception reports act as a message to tell each node what packets they hold.

Delay and jitters performances are shown in figure 3.13 and figure 3.14. The highest delay is recorded for MAC-P techniques in 802.11g. The MAC priority in [24], [1], [34], [41], [116] and [50] have shown an average of 2.5 seconds. The delay while using the proposed MAC-P was around 2.09 seconds. Normal MAC was lowest with 1.68 seconds. The proposed MAC-P was worse for a short period when the offered load was between 4 to 13 Mbps. The box-plots in figure 3.13 shows that normal COPE had the lowest delay with dense values and no abnormalities. The same can be said for proposed MAC-P and MAC-P. It seems that MAC propriety technique has trad-off. It can give opportunities to relay nodes to boost throughput, but it causes more damage in term of delays as depicted in figure 3.13. When there is MAC priority, there is accumulation of packets at centre nodes while normal COPE did not induce any extra delay while finding and coding packets at the queue of intermediate nodes. Therefore, normal COPE had the least value of delay. But, it is small drawback compare to significant improvement in throughput as in our proposed MAC-P and MAC priority techniques.

In term of jitters, the proposed scheme was the lowest and least fluctuated box-plots 3.14 with an average of 0.0072 seconds. MAC with priority was second (black) with average jitters of 0.0113 seconds. Furthermore, normal COPE was the highest with 0.0283 seconds. Furthermore, the box-plots of normal COPE were much wider than MAC-P, and values of normal COPE had skewed from the mean vigorously. Although these values are still bigger than both MAC-P and proposed MAC-P. Despite that delays in both MAC propriety techniques and our proposed MAC-P were noticeable, jitters were much small because of indirect effect. The effect of allowing senders nodes to transmit packets and subsequently reduce the variation of packets delay if these packet delivered correctly.

The RLP results are shown in figure 3.12. It shows the results for all test implementations (proposed MAC-P, MAC-P, and normal [COPE]). The normal MAC was highest in lost packets with an average of 72%, while MAC priority was much better than previous normal MAC. It had an average of 44% RLP, while the proposed solution has the lowest RLP, which was around 38%. Figure 3.12 shows that box-plots of normal COPE are wider than MAC-P and proposed MAC-P, and the values skewed with upper and lower whiskers at high offered. On the contrary, the proposed MAC-P and MAC-P had dense values with few skewed values in case of MAC-P. The reason why normal COPE had the highest RLP rateis because of its big CW value as compared to the MAC-P and proposed MAC-P. Therefore, the RLP was higher for COPE using normal MAC.

On another hand, our proposed MAC-P techniques had enjoyed much more frequent access to medium as a result of small CW values assigned to intermediate nodes. As a result, the number of lost packet was relatively lower than MAC-P techniques. MAC-P technique had more bigger CW values than our proposed MAC-P.

In conclusion, the proposed algorithm challenges the weaknesses which are shown in the first COPE and COPE with MAC priority. The number of coded packets in the proposed algorithm is higher than COPE with MAC priority: it is doubled while maintaining higher throughput. During the simulation time, the packets were encoded and transmitted continuously, and congestion was not an issue. Moreover, the queue at the relay node has not recorded any dropped packet notifications during the simulation. The proposed algorithm can work to ease the congestion in the network. It can save sent packets from being dropped significantly. On another hand, it seems that MAC propriety technique has trad-off. It can give opportunities to relay nodes, and it also enhanced jitters and RLP. Nevertheless, it causes more damage in term of delay.

On other hand, the proposed solution guarantees the least waiting time for nodes which perform Xor network coding in a more controlled manner. For example, the proposed solution allows the relay nodes to transmit the coded packets, which is crucial to boost the throughput in the network. Meanwhile the two edge nodes were also found to enjoy the benefits of no doubling times when they had failed to reach the medium, and that is because of the proposed scheme allocates DCF channel in a better manner, favouring relay nodes that perform coding by using the best CW value for relay node without constraint from other nodes or edge nodes in a wireless network. It maximises the chances for more throughput in two and more hops. As a result, the proposed contention window scheme at MAC layer has improved the throughput when compared with other conventional schemes. Furthermore, the proposed solution has guaranteed the relay nodes a least possible waiting time and quick access to the medium. Contrary to the 802.11 standards, it helped the relay nodes to empty their coded packet queue without imposing medium access fairness. Therefore, the proposed solution is suited for Xor network coding.

3.5.3 Multi-Hops Topologies

Using the same stepwise criteria for the proposed algorithm, figure 3.15 shows many implementations and drops in throughput when the number of hops increases in the MAC layer basic mode (disabled RTS/CTS mode). It is clear that COPE [12] fails to reach reasonable throughput at ten hops; however, COPE with MAC priority provided better throughput [24], [1], [34] , [41], [116], [50] over on different number of hops. The proposed solution has much higher throughput than COPE with MAC priority and normal COPE on all hops, and the proposed MAC-P scored the highest throughput on all hops. Because of better CW values assigned to nodes when nodes failed to win the medium. These values are based on criteria previously mentioned in section 3.3. The reason both normal COPE and COPE with MAC priority deteriorated was because the two were using the default CW calculation.

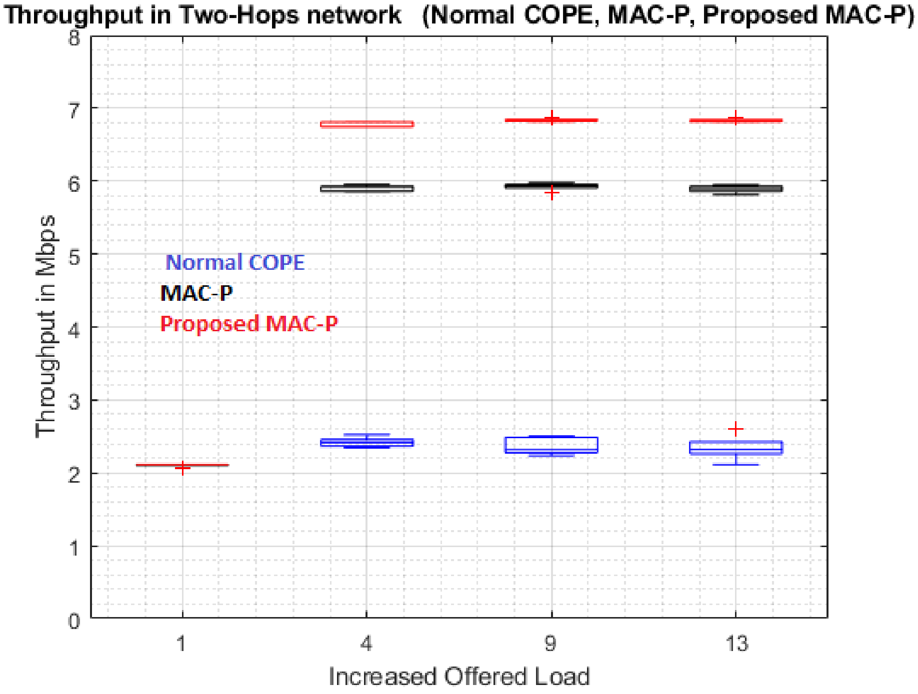


Figure 3.11: Throughput in two hops, increased offered load.

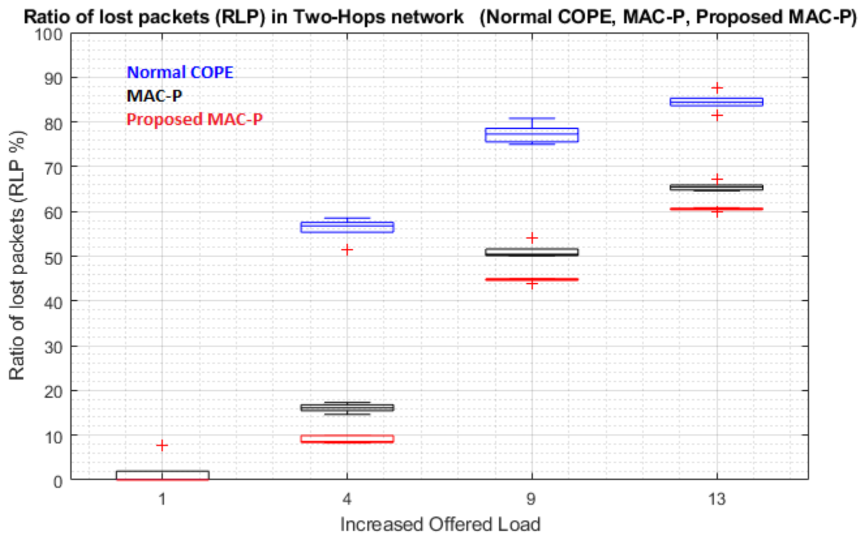


Figure 3.12: Ratio of lost packets in two hops.

The average throughput of the proposed MAC-P is 2.76 Mbps. Figure 3.15 shows the throughput for proposed MAC-P with the maximum value of 6.76 Mbps. In the box-plot of proposed MAC-P, the skewed values are noticeable in hop 7 to hop 9, and there was one occasion when the outlier is far away from other data values in hop 6. Between hops 7 to 9, the box-plots of proposed

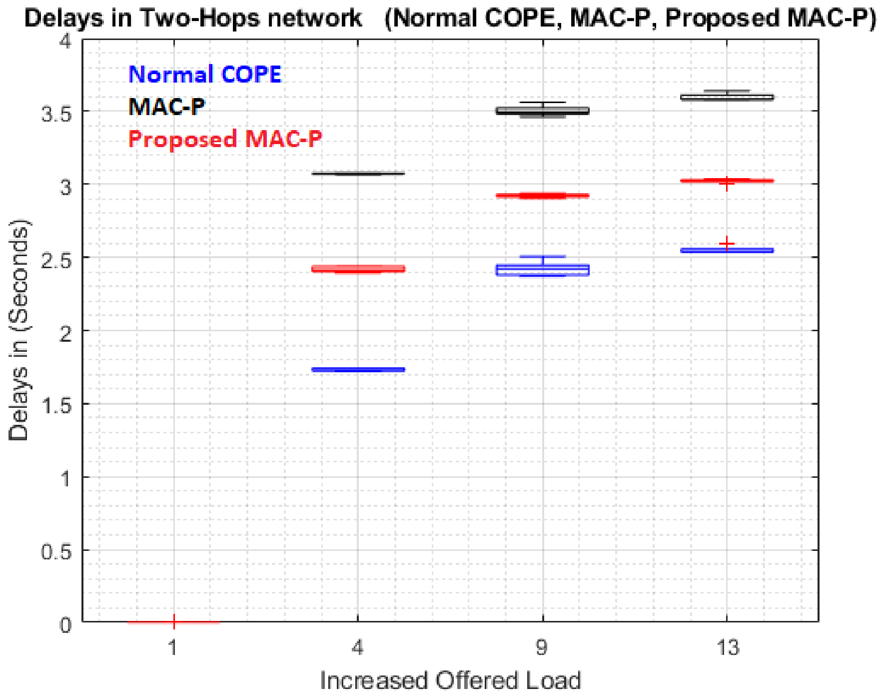


Figure 3.13: Delays in two hops.

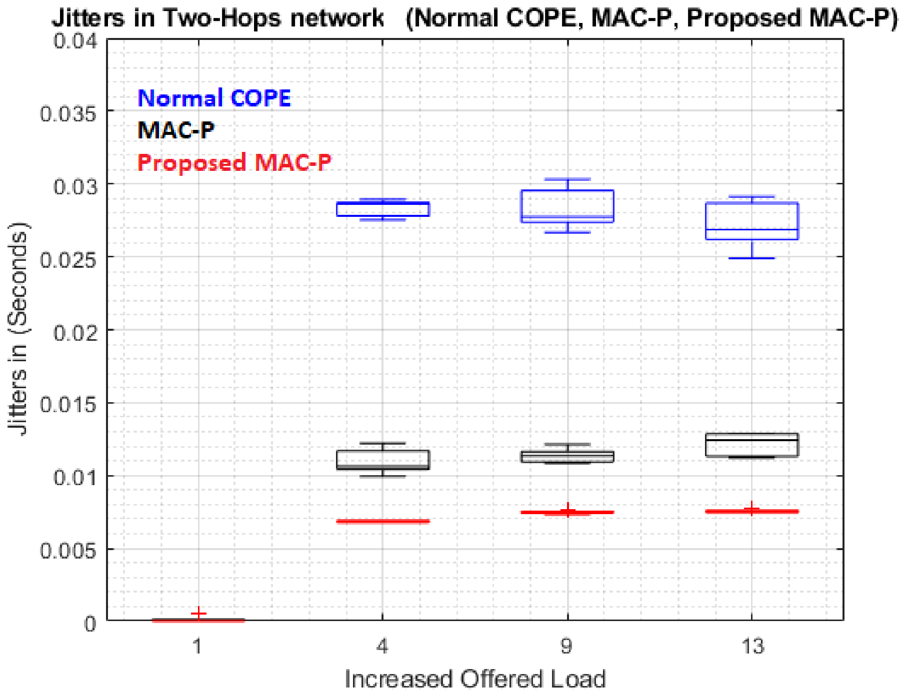


Figure 3.14: Jitters in two-hops.

MAC-P were muddled with MAC-P technique. The box-plots of proposed MAC-P were skewed widely, which showed abnormality in values in these hops. By and large, the proposed MAC-P was not overtaken by normal COPE, and abnormality of values was determined by randomness of simulation chosen seeds. It is clear that proposed MAC-P may suffers performance degradation between hop 7 and hop 9. Although, the overall average throughput is higher than MAC priority and normal MAC using COPE. In case of MAC priority, the box-plots at hop 6 to hop 9 shows sign of abnormality, where outlier is once up and below the expected value. However it is still within the edge of fewer than one Mbps. Normal MAC is edgy and out of the normal distribution value for some hops. For instance, hop 3 and 4 have both one value was far from other throughput values. The ratio of lost packets in figure 3.16 shows the marginal difference between values for all three implementations. The box-plot in figure 3.16 is small due to the small variance of all RLP vales. The Normal MAC COPE and the proposed MAC priority have widespread box plots in hop 4 only. The proposed MAC priority has slightly much better RLP, and MAC-P came in second place. The RLP of the proposed solution was the lowest when it is compared against COPE using normal MAC, and COPE using MAC-layer priority. However, RLP is only significant until 5th hop. For example, at 2nd hop RLP improved 61% and continues to improve less than 10% until 5th hops. In case of 7th hops, the proposed MAC-P was weirdly skewed in values as shown in figure 3.16 box-plot. It was extended to overtake both normal COPE and COPE with MAC priority. It presented skewed RLP values as sign of abnormality. Meanwhile, normal COPE has the highest RLP. The RLP ratio grew up gradually reaching the highest value (98%). The RLP of MAC-P techniques was positioned between other two implementations. The RLP of MAC-P techniques reached minimum of 60% at two hops and maximum of 96%. When we looked into COPE using normal MAC, it did not enjoy frequent access to the medium, and that because it had a bigger CW value than MAC-P and proposed MAC-P. Therefore, the RLP was the higher for COPE using normal MAC. On contrary, MAC-P techniques had enjoyed much more frequent access to medium as a result of small CW values assigned to intermediate nodes.

On the other hand, the delays of all implementations are braided when they reached the eighth hop. Figure 3.17 show the delay of all three implementations. Before reaching the tenth hop, the proposed MAC-P was slightly small compare to MAC-P with an average of (8.96) seconds. Meanwhile, the average delay of MAC-P was around 11.72 seconds. However, at 10th and 11th hops the proposed MAC-P was highest, the box-plot of the proposed MAC-P indicates some abnormality at the last hop. The overall delay of the proposed MAC-P went up as the number of hops increased, the same as MAC-P and Normal MAC COPE. Besides, the skewed values of proposed MAC-P is evident from the 6th hop to 11th hop for all implementation, which is consistent with the fact that the delay increases as path to destination increases. Nevertheless, the proposed solution was the highest at 12 hops. It saw an average of 18.55 seconds. Furthermore, the delay has improved by 19% when compared to the MAC-layer priority, but the delay was not improved in the case of normal MAC. The box plots of MAC-P values were far or below

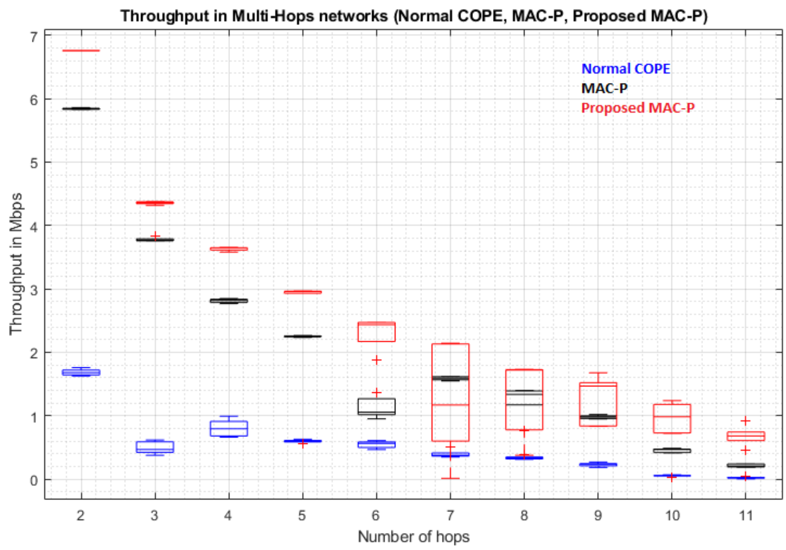


Figure 3.15: Throughput in Multi-Hop networks.

the range of normal distribution, especially in 6th and 10th hops. The same applies to the proposed MAC priority and normal MAC COPE. Normal MAC COPE had the lowest delay for all implementations. Because of no MAC priority, there was not any accumulation of packets at centre nodes while MAC-P proposed MAC-P added extra delay while finding and coding packets at the queue of intermediate nodes. However, it is a small delay compared with the benefit of higher throughput. Figure 3.18 shows the jitters for all three implementations.

The jitters of the proposed solution was the lowest improvement against MAC-P, at 61%. Normal MAC was highest in term of jitters, with an average of 0.26 seconds. While MAC-P and the proposed MAC-P were quite close until the 7th hop. The proposed solution has an average of jitters of 0.0565 second, while the average of MAC-P was 0.096 second. The box-plot of proposed MAC-P has some values abnormal skewed between hops 7th and 8th. In the case of MAC-P, it looks more settled until 10th hop. Whereas for normal MAC using COPE is more variable than others. For example, some values were out of range in the 10th and 11th hop. The overall both MAC-P and proposed MAC-P are expected to have small jitters because of having more small CW values, and that helps sending nodes to gain more access to pass any packets in their queues, and subsequently reduce the variation of packets delay. To summaries, Table 3.2 depicts the calculated average jitters for every hop in our simulation. Furthermore, Table 3.3 presents overall improvements in throughput when compared with MAC priority. What contributes to this improvement is the choice of CWs on the MAC layer that works in harmony with Xor network coding.

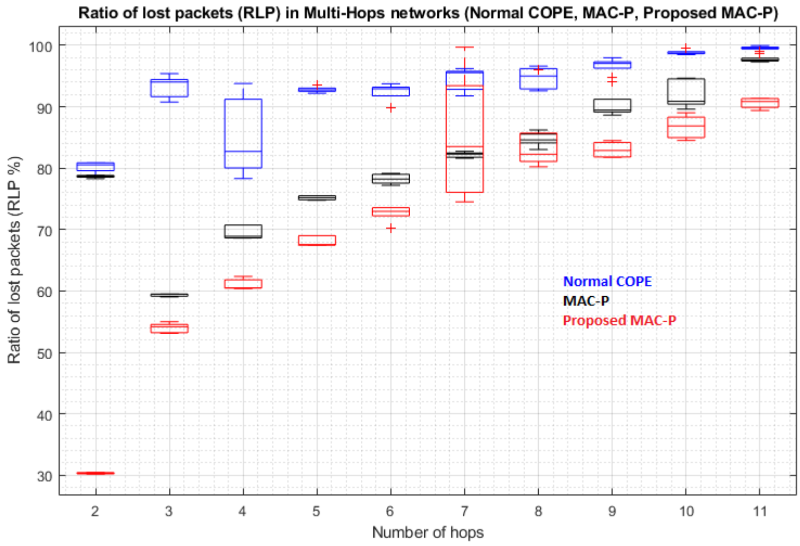


Figure 3.16: Ratio of lost packets in Multi-Hop networks.

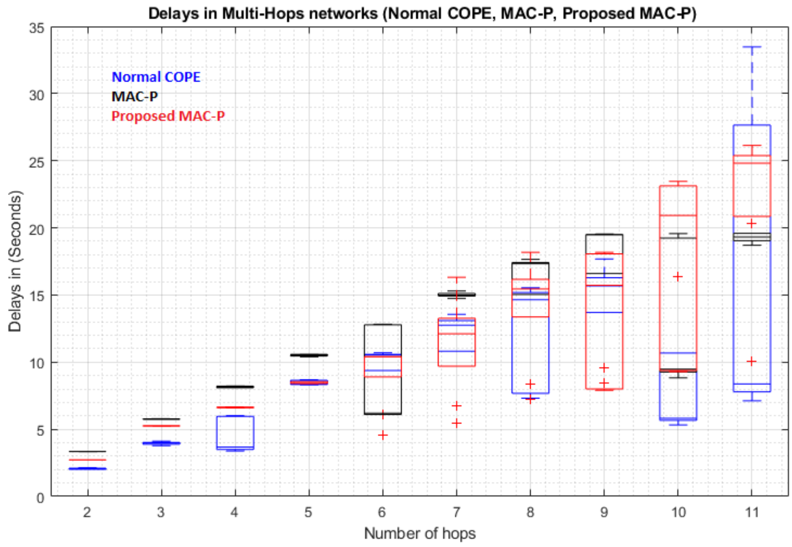


Figure 3.17: Delays in Multi-Hop networks.

3.6 Discussion

The inherited overhead that exists in state of the art MAC 802.11, COPE using current MAC and COPE with MAC-layer priority, achieved low throughput due to this fact. The inherited overhead lowers throughput of the 802.11 DCF to around 20% or 30% of bit-rate depending on variant [117]. As a result there is wasted throughput in a traditional MAC-layer scheme, and COPE inherits the weakness caused by overheads. Still, it is not the main problem we face; Duda [117]

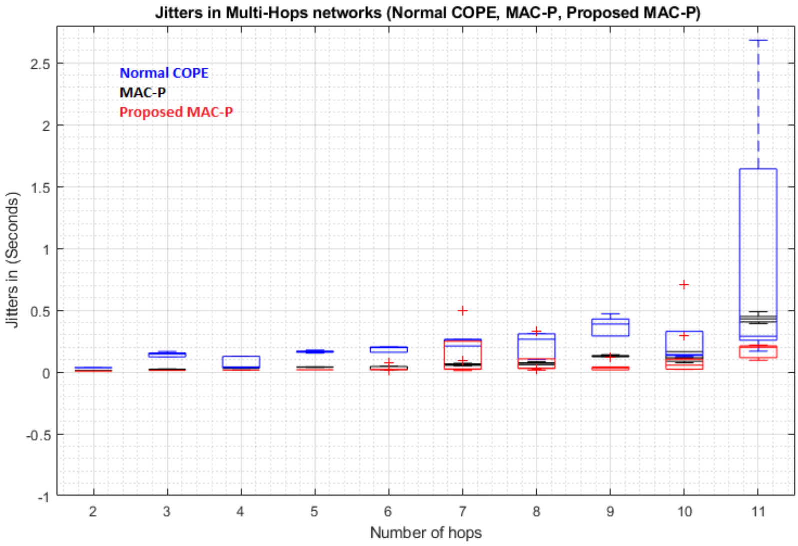


Figure 3.18: Jitters in Multi-Hop networks.

argued that the random back-off algorithm increases overheads (by up to 35% for 802.11b and 43% for 802.11g) and provides 7.15 Mbps to 30.68 Mbps throughput [117]. That is true when the CW's calculation comes into play and doubles the CW's value, which creates a real challenge to network coding relay nodes to gain access in cases of more than two hops. Also, the contention between stations increases the probability of collisions and decreases available throughput. When a station senses the channel is busy, it waits for a random back-off time, which further lowers its useful throughput (the random back-off even applies to a single station). The average waiting time of a station is given as:

$$AverageWaitingTime = \frac{CW}{2} \times SLOT \quad [117], \ mentioned \ in \ section \ 3.3 \tag{3.11}$$

According to the above equation, for a single station efficiency decreases to $U = 0.65$ for 802.11b (throughput of 7.15 Mb/s) and $U = 0.57$ for 802.11g (throughput of 30.68 Mb/s) [117]. Indeed we could experience this in the case of COPE. However, COPE itself cannot overcome inherited wasted throughput. On the contrary, when altering the CW of the relay node to accommodate its function in the network, it could reach higher throughput than COPE. Although, COPE with MAC-layer priority results in few or no coded packets, due to the realisation of no matching pairs to be coded together.

For a small number of stations, 802.11 benefits from good short-term fairness. When the number of stations are increased, the short-term fairness of 802.11 becomes worse. This is due to the exponential back-off (CW). In order to improve short-term fairness, an optimal DCF-like access method needs to use the equal sizes of CWs for all contending stations [117]. The Authors in [35] had the same argument that network coding would suffer from short-term fairness in cases

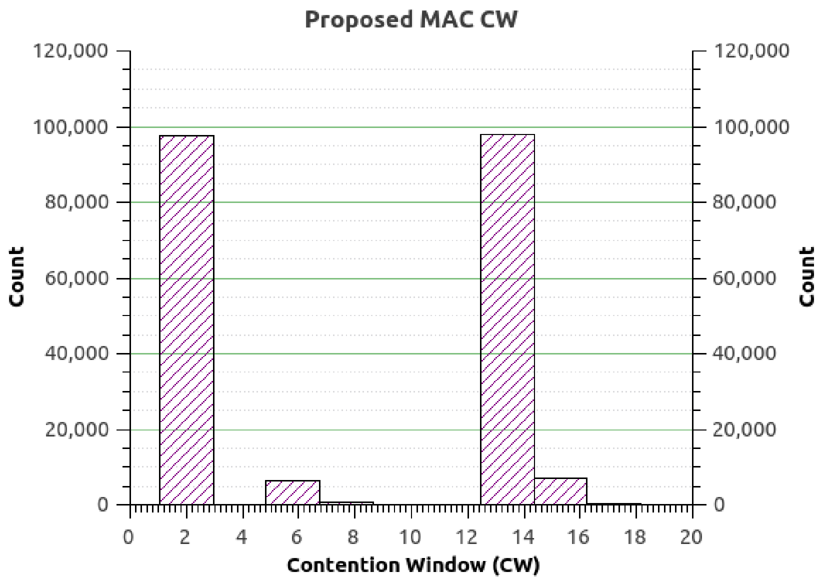


Figure 3.19: Histogram of proposed MAC-CW.

of intersessional coding. However, their solution can help with balance but leads to starvation of rich traffic. As the experience with three hops and random topology cases show, it does not provide real improvement to the overall throughput. It is a short period that can be solved using different initial CWs for every station. Another possible remedy can be performed by managing the queue in relay nodes to balance the traffic as in [35].

Looking at the contention window at the MAC layer for the proposed MAC-CW can prove the superiority of the scheme in providing the minimum value for relay nodes in the wireless topology. Figure 3.19 depicts the histogram of the proposed MAC-CW, where the relay nodes have been assigned CW values between 2 and 3 at the beginning of the simulation initiation stage, where edge nodes are being assigned a default setting 15. Interestingly, as the simulation progressed, relay nodes obtained values between 13 and 17. However, the most common values were between 12 and 15 for relay nodes, as figure 3.19 shows. Equally the edge nodes will assign much lower values, between 2 to 3 or 5 to 8. The most counted values were for CW of size 2 – around 100,000 counts in histogram figure 3.19.

3.6.1 Implications of MAC Priority on Non-prioritised Flow

The effect of MAC priority on other traffic which doe not enjoy the benefit of MAC priority is overestimated. In principles, if the flows are not assisted by MAC layer interaction, the flows are subject to fair share principle that stated in 802.11 standards. However, network coding is about coding at the intermediate node, the fairness of the MAC protocol might implicate such a role. To understand the effect of MAC priority on non-prioritised traffic, let us consider the topology in figure 3.20. The topology consists of seven nodes, five in cross and three in liner arrangement.

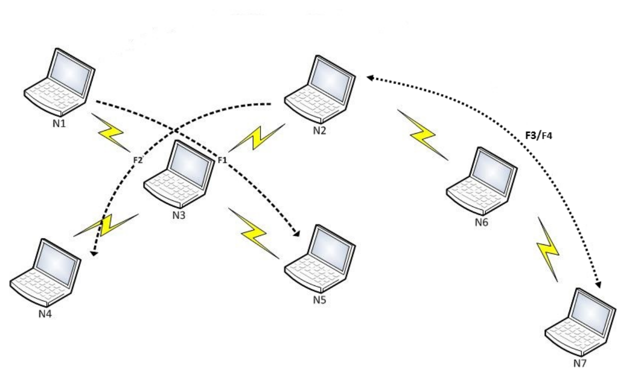


Figure 3.20: Seven Nodes Topology.

Traffic generated from node 1 to 5 and node 2 to node 4 in cross topology. In linear, the traffics are bi-directional traffic between node 2 and node 7. In cross topology node 1 and node 2 send traffic and the middle node encodes for both node 4 and node5. They have MAC priority feather turned on. The linear topology has no MAC priority benefit. Node 2 and node 7 use the normal MAC standard 802.11, and CW is fairly contented between nodes. After running the simulation for 50 times, it is clear the throughput in flow 1 and two 2.

Flow 3 and 4 have experienced less throughput, for example, flow 3 could reach a maximum of 0.6 Mbps, and a minimum of 0.26 Mbps. On the other hand, flow 4 reach a maximum of 2.14 Mbps, but the overall average is 0.9 Mbps. The reason for such an outcome comes from the nature of competing on the medium. Every node try to gain access, and the one that wins the medium will win the access. The impact of other MAC priority (flow 1 and 2) is limited as all nodes are put on the edge of each other, and without MAC priority the flow is not supported to gain frequent access, which can be coded at the intermediate node. Because of that, the throughput of non-MAC priority is low.

In the box-plot of figure 3.21, in the case of flow 3, 50 % of values of throughput would not reach the 0.5 Mbps. Moreover, in flow 4 box-plot the value skewed when the offered load is 9. Half of the values could go up to 1.5 Mbps. Although some values are more variable than others, as figure 3.21 indicates. The ratio of the lost packets seems similar at first glance, but the advantage is for MAC priority flows. It is still a small marginal difference. This marginal improvement of RLP was a result of the nature of UDP traffic which is subjected to loss. Figure 3.22 showed that skewed in values for RLP is small, except for low offered load in case of flow 4 and flow 1. Flow 4 has a median value of 30. However, this was in case of low offered load only. In the same category, the 75 % values of flow 1 would be under 40% of RLP when the offered load of 1.

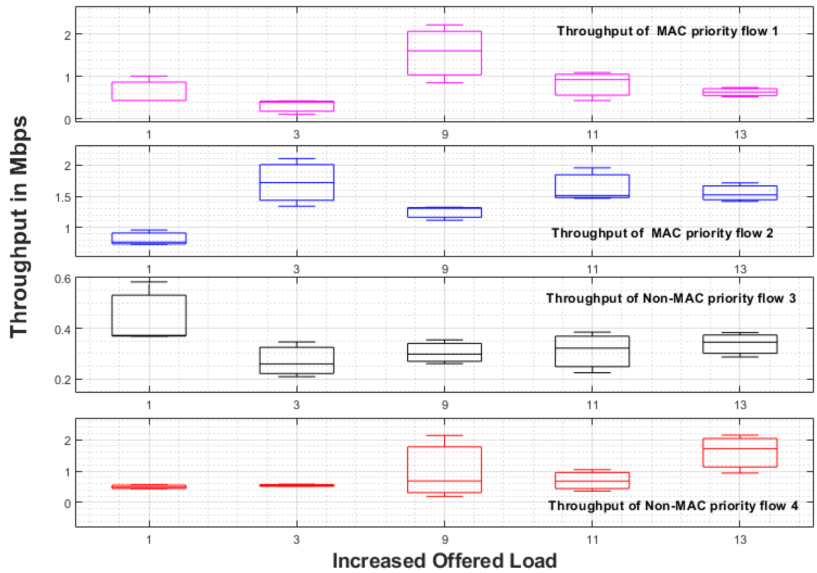


Figure 3.21: Throughput of All flows.

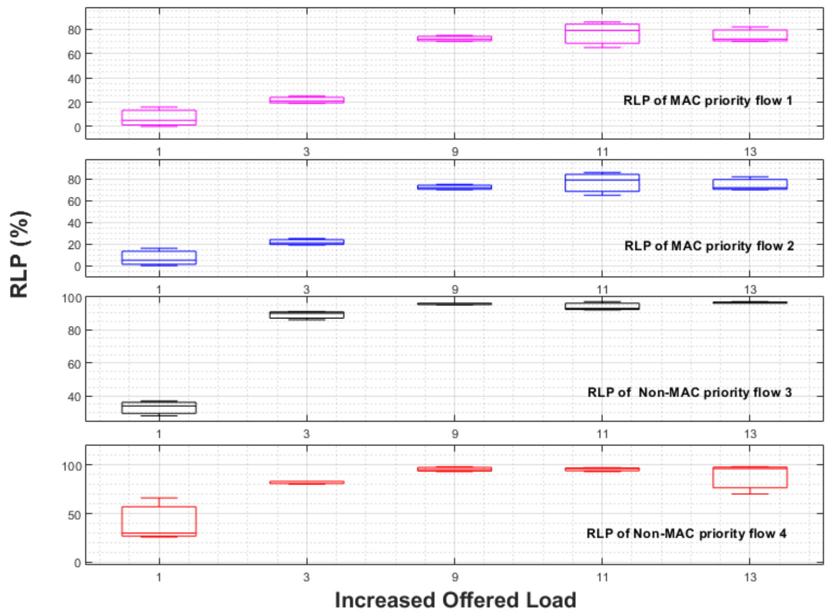


Figure 3.22: Ratio of Lost Packets of All flows.

The rest of box-plots in figure 3.22 are nearly between 89 and 93 per cent. Delay and jitters of all flows are more variable than others. Figure 3.24 and figure 3.25 show that such varied values in case of delays and jitters of all flows. Flow 3 suffered a heavy burden of delay - a maximum of 6 seconds delay. Moreover, the average value of the overall delay was 3.04 seconds, with a median of 2.97 seconds. When the offered load increased, the variation of delay values is small.

The same can be said for flow 4, except in the case of low offered load in the network. There is a very strong similarity between flow 2 (MAC-priority) and flow 4 (without MAC priority), both have the same maximum value and their values follow normal distribution even when they are apart away. The median of flow 4 is slightly smaller around 1.0633 seconds, while the median of flow 2 is 1.12031. Both have a similar median of 0.9 seconds. The overall delay is recorded in case of non-prioritised flows. Jitters of prioritised flows (f1 and f2) have smaller value due to MAC priority feature and frequent access to a medium through CW. However, the non-prioritised flows (f3 and f4) have higher values. Figure 3.25 depicted such observation, for example, flow 1 and flow 2 the least jitters of 0.00752 and 0.00492 second respectively. The highest was seen in flow 3 and flow 4 0.05082 and 0.007732 second, respectively. The box-plot of flow 1 and flow 2 seemed more fluctuated than flow 3 and flow 4. The reason for the small value of jitters is MAC priority through assigning small value of CW in the intermediate node, so it can assist the flow packets to pass down the medium. However flow 3 and flow 4 have higher jitters values, which is due to the significant CW value. Figure 3.25 shows that flow 3 and flow 4 box plot are more smooths and values are commonly distributed. Flow 1 box-plot is much skewed, but the values of jitters still small compare to non-prioritised flows. Jain's fair index equation in equation (3.12) has been applied to study the implication of MAC priority on non-prioritised flows in the network in figure 3.20.

$$\frac{(\sum x_i)^2}{n \sum x_i^2} \tag{3.12}$$

In the case of non-priority flows, the fair index was fair when offered load is low. It is only fair at very low load around 92% fairness for all MAC-priority and non-priority flows. Moreover, as the load increases the index leveled up to around 75% fairness, which means around 25% of flows are starved (non-priority flows). Meanwhile, Jain's fair index was 0.9998 for two flow using the proposed MAC-P. Therefore, MAC-priority technique is only fair to nodes who are applying its principle, and non-priority flows are subject to struggle for gaining the medium. In fact, that is very reasonable as MAC-priority technique is about giving more access to designated nodes rather than competing fairly. Figure 3.23 reflects the struggle of non-priority flows. There was only abnormality in figure 3.23 when the offered load of 3, the fairness index could skew down to 0.6 for all flows.

3.7 Summary

It is important to understand the vital roles of the CW (back-off) calculation when network coding is applied, and the role of relay nodes in the network. Our proposed solution can guarantee the lowest waiting time for stations in a more controlled manner. It also gives relay nodes between two ends the advantage of using the medium when there is a coding opportunity. The two edge nodes can also enjoy the benefits of no doubling times when it fails to reach the medium. In

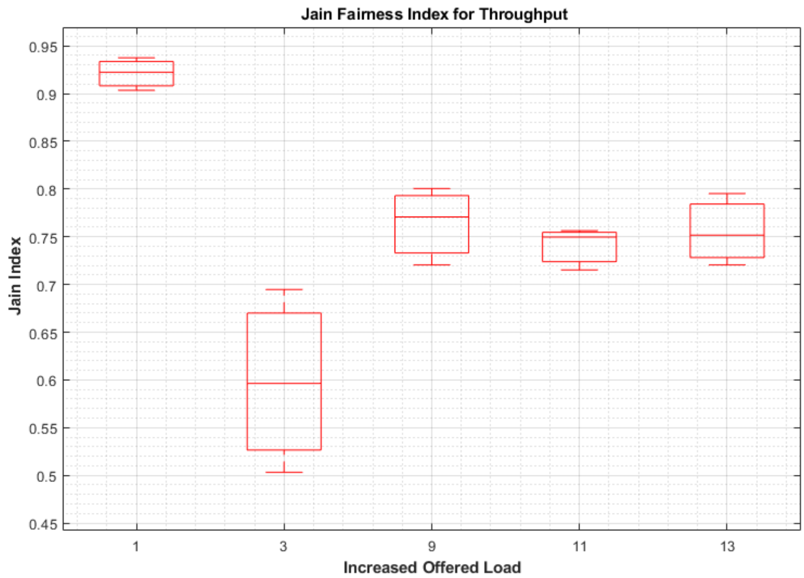


Figure 3.23: Jain Fairness Index for Throughput of all flows.

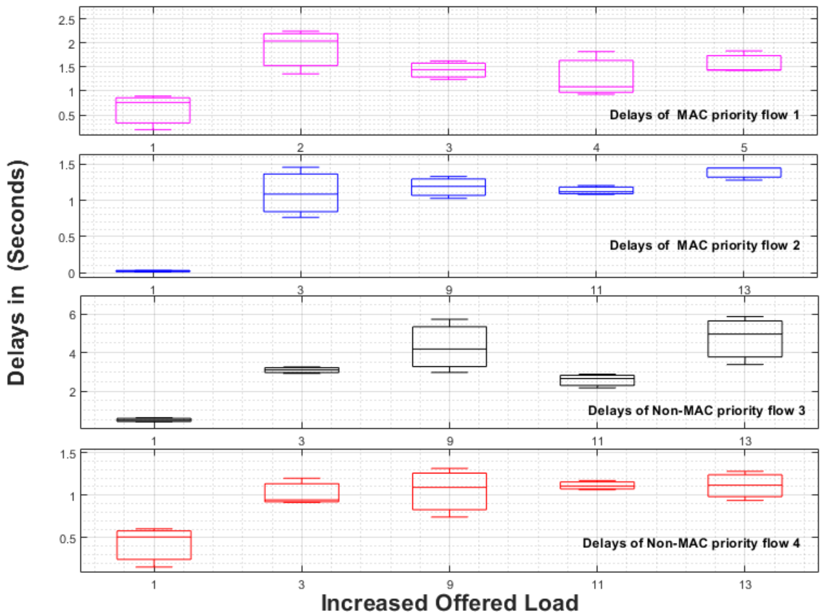


Figure 3.24: Delays of All flows.

addition, the proposed scheme allocates the DCF channel in a better manner favouring relay nodes that perform coding by using the best value for relay node without constraint from other nodes in the network. In addition, the proposed scheme performed well against COPE and COPE with MAC-layer priority.

The finding of this chapter can be summarised as follow:

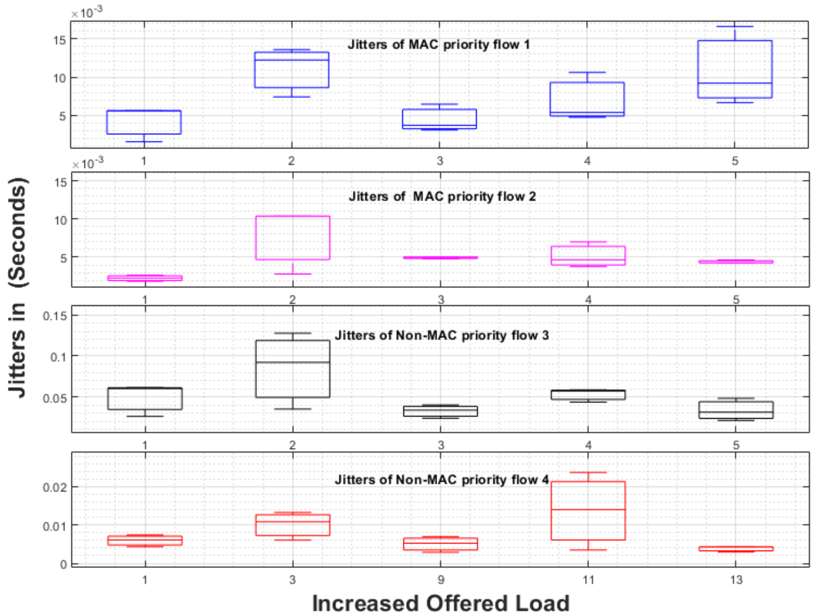


Figure 3.25: Jitters of All flows.

- COPE has very low throughput; it did not reach the maximum utilisation or throughput, and COPE does not use mediums efficiently. On the other hand, COPE with MAC priority can reach a reasonable level of throughput, but the number of coded packets is lower. These packets contribute slightly to the overall throughput. COPE with MAC-layer priority results in few or no coded packets due to the realisation of no matching pair to be coded together at a time.

- The proposed algorithm gives middle nodes a higher share of the medium to improve total throughput. It maximises the chances for more throughput in two and more hops. Furthermore, it does not restrict the bidirectional flows generated by edge nodes from accessing the medium when they have packets to send.

- The proposed scheme has improved the throughput when compared with other implementations (figure 3.15). The improvement ranges from 24% to 97% over 11th hops. For example, the proposed scheme has improved the delay by 24% when compared with COPE with MAC priority.

- Jitters and delay have proved to be the lowest against COPE using normal MAC layer and COPE with MAC- layer priority. It has showed 96% improvements in terms of jitters. Furthermore, the delay improvements in the proposed scheme was around 7% against MAC-layer priority, where in cases of standard MAC it was not any improvement.

- The RLP was the lowest when put against COPE using normal MAC and COPE using MAC-layer priority. But it was significant until the 5th hop. For example, at the 2nd hop

RLP improved by 61% and continues to improve less than 10% until the 5th hops. On the contrary, COPE using normal MAC did not enjoy access to the medium frequently when it has a packet in the queue. And that is because of assigning big CW value in traditional 802.11 DCF. Therefore, the RLP was higher for COPE using normal MAC.

- COPE using current MAC and COPE with MAC-layer priority inherited overheads that exist in state-of-the-art MAC 802.11. As a result, they achieved low throughput. Moreover, the random back-off algorithm in DCF increases the overhead and provides low throughput. Unconventionally, when the CW's calculation comes into play and doubles the CW's value of the relay node, it creates a real challenge to network coding relay nodes to gain access to the medium. Further, contention between nodes increases the probability of collisions and decreases the available throughput.

- A small number of nodes in the network can benefit from good short-term fairness, and this does not contribute to significant throughput. However, it is a short period that can be solved using different initial CW for every node, or another possible remedy can be done through managing the queue in relay nodes to balance the traffic as in [35].

- The impact of MAC priority on non-prioritised flows is limited, and without MAC priority the flow is not supported to gain frequent access, which can be coded at the intermediate node. Because of that, the throughput of non-MAC priority is low. It has the same throughput in case of COPE using normal MAC.

- The non-prioritised flows suffer prolonged jitters and delays. The reason is the lack of MAC assisting the flow, for example, MAC priority has the small value of jitters and delays due to assigning a small value of CW in the intermediate node, so it can assist the flow packets to pass down the medium. Furthermore, COPE using normal MAC has a similar attitude; it has no MAC priority feature. Therefore, it has higher delays and jitters values.

- In case of non-priority flows, the fair index was fair when offered load is low. MAC-priority technique is only fair to nodes which are applying its principle, and non-priority flows are subject to struggle for gaining the medium. In fact that is very reasonable as MAC-priority technique is about giving more access to designated nodes rather than competing fairly.

Jitters in seconds			
Number of Hops	Non-MAC-P (Normal MAC)	MAC-P	Proposed MAC-P
2	0.03501228	0.01212816	0.00725624
3	0.1410698	0.02190086	0.01229746
4	0.07283218	0.02991768	0.01425892
5	0.1648278	0.04014696	0.0175881
6	0.1736427	0.02834286	0.019019472
7	0.22641808	0.06027966	0.14503756
8	0.2189188	0.0641713	0.08811296
9	0.348149	0.1293306	0.02976436
10	0.2627564	0.13924354	0.06142308
11	0.9447328	0.4309664	0.1687412

Table 3.2: Average Jitters in Multi-Hops.

Improvement in Muli-Hop (Mbps)	
Number of Hops	Improvement in %
2	75
3	89
4	78
5	80
6	76
7	68
8	74
9	82
10	94
11	97

Table 3.3: Improvement in Multi-Hop.

CHAPTER 4

TCP Reliability and Latency in Coded Networks

In this chapter, the thesis investigates the challenges to improve the throughput of TCP in the case of Xor network coding in a Multi-Hop wireless network coding. COPE has not shown any real improvements in terms of throughput in TCP, and it is still a real challenge in literature. Xor network coding does not benefit from TCP's high reliability and throughput performance. TCP takes the nature of networks into consideration, but it does not acknowledge the network coding process which allows to ease the congestion by performing coding at intermediates nodes such as in COPE. Meanwhile, the perspective of network coding has a more rigid approach that depends on the relay nodes to perform coding rather than being dependant on the message exchange between the sender and the receiver. This delegates network condition on current transmission. This chapter looks into TCP variant implementations such as Tahoe, new Reno, Westwood. The chapter presents WeNC-TCP which can accommodate the network coding nature functionality in the Transport layer.

4.1 Introduction

The main functions of TCP include controlling congestion and managing overflow traffic in the network, which improves throughput and overall performance. The traffic control or management is carried out by increasing and decreasing the value of the Cwnd according to network conditions [123]. The Cwnd limits the amount of sent data to ensure smooth transit that avoids congestion and eases any traffic overflow in the network. At the receiver, the destination Cwnd is advertised to the sender, and tells the sender about the destination's ability and availability to handle one or more packets. TCP is recalculated to the maximum number of bytes of unacknowledged data allowed to be sent down on the medium, and the minimum is taken from both the Cwnd and the advertised window as follows [123]:

$$MaxWindow = min(cwnd, AdvertisedWindow) \quad [123]. \tag{4.1}$$

When the ACK is received within the retransmission timeout (RTO) timer, the Cwnd will be doubled in value and it enters the case of the slow-start. Again, when the Cwnd comes closer to the slow-start threshold, it will build up linearly by adding one maximum segment size (MSS) to the Cwnd for every ACK it receives [124]. When there is an indication of packet loss, the congestion avoidance mechanism comes into play, and it halves the slow start threshold to the current Cwnd and deducts the Cwnd size to one MSS [124]. We can describe TCP's congestion window mechanism in steps for simplicity as follows [125]:

- When the Cwnd is below the threshold value, the sender is in the state of slow-start, and therefore, the Cwnd increases exponentially quickly.
- When the Cwnd is above the threshold line, the sender is in the state of congestion avoidance, and as a result, the Cwnd increases linearly.
- When a triple-duplicated ACK-loss incident develops, the threshold is assigned to be one half of the current Cwnd, and the Cwnd is assigned to the threshold value.
- When a timeout loss incident develops, the threshold is assigned to be one half of the current Cwnd, and the Cwnd is set to be 1 MSS.

The vital question is: why does TCP behave differently after a timeout event than after a triple-duplicate ACK incident, and is it seen as an evolutionary process? The first TCP version (Tahoe) unconditionally halves its CW to 1 MSS, and goes into slow-start mode. However, the new version of TCP, TCP Reno, writes off the slow-start phase after a triple-duplicate ACK. The philosophy behind it is that, although a packet is lost, the arrival of a triple duplicate ACK indicates some segments have been received, and they are being acknowledged. It is unlike the case of a timeout – it proves that the network can deliver some segments (possibly with some lost or out of order). Therefore, omitting the slow-start phase after a triple-duplicate ACK can be replaced with so-called fast recovery [125].

4.2 Problem Statement

The impact of network coding on TCP was studied by [86], [60], [61] and [62], for instance, the solutions in [86] is based on xoring the Acknowledgement packets to improve TCP reliability and throughput. Where as in [60] likened the congestion window to channel condition, for example, TCP sender responded upon realising any changes in link conditions. If the TCP sending rate is very high, it will choose a scheduler with the highest channel rate (QAM64), and if the link quality is low, it chooses (QAM16). Authors in [62] recommended the use of the synchronisation of TCP flows which will yield to contribute more coding rates, however, it marginally enhanced the TCP good-put.

Interestingly, authors in [62] [61] argued that the TCP function of the AIMD (additive increase/multiplicative decrease), had a significant impact on the benefits of network coding. Usually TCP's congestion control mechanism continuously adapts the TCP sending rate to network conditions according to the current available capacity of a network. Specifically, the AIMD mechanism is incredibly sensitive to packet losses and translates them as signs of congestion. Furthermore, upon detecting a loss, TCP halves its sending rate by reducing its CW size by half. In the early simulation test, it was clear, and there was AIMD impact on Xor coding network throughput as in COPE [12]. The TCP's AIMD has not been designed to accept the interaction of intermediate nodes role in the coded network. AIMD was suitable for the classical transmission paradigm. Thus, it can do well in wired networks.

In wireless networks, slashing TCP upon detection of congestion or lost packets can restrict the utilisation of medium and holding back any attempts to utilise unused resources. For example, COPE can reduce traffic congestion on intermediate nodes by finding the right packet to encode and broadcast, and as a result bandwidth and network resources can be utilised for another transmission opportunity. That is not all. The wireless network is situated on the principle of contention on MAC layer as in IEEE 802.11, and there are many packet losses due to wireless channel errors and contention on the medium. Again, TCP reacts to all such packet losses by reducing its transmission rate by half, and that results in low throughput and average utilisation. Therefore, TCP throughput is fundamentally limited by the end-to-end loss probability rather than the available end-to-end capacity [61]. Can the performance of Xor network coding improve if the TCP was encouraged not to react to every incident as a sign of congestion? This is the starting point of our investigation.

When compared to UDP, TCP does not gain any significant benefits from the increased capacity due to the network coding process. Author in [61] concluded that TCP only benefits indirectly from network coding, the network coding reduces the number of transmissions which results in lower contention on the wireless medium than the non-coding approach. Furthermore, network coding was seen by author in [61] as a packet-damage factor, or incurring delay factor in the overall end-to-end delay. This work and other works blamed solely congestion mechanism, but are there other TCP functionalities affecting or implicating the performance of Xor network coding in wireless multi-hop networks ?

The need to revisit the design of TCP functionality could benefit the TCP to enjoy the network coding ability and reduce contention and the number of transmissions on the medium, and ultimately improving TCP throughput and reliability.

4.3 System Model

This section describes the applied system model. The model has been implemented and tested. The implementation consists of three parts: TCP Tahoe, TCP New Reno and TCP Westwood. The proposed WeNC-TCP solution, which has the base structure of TCP Westwood, is also described in this section.

4.3.1 TCP Tahoe

The first simple TCP implementation was Tahoe [126], and was based on a go-back-n model with accumulative positive acknowledgements and the expiration of the RTO, which was required before the flow was able to transmit any lost packets. TCP Tahoe facilitated slow-start, congestion avoidance and fast recovery algorithms [60]. When a packet is lost, fast transmit is triggered to reduce the waiting time of transmissions instead of waiting for the retransmission timer to expire. If TCP triple-duplicate ACKs are received, the sender infers a packet loss and retransmits the lost packet [127].

The sender now sets its ssthresh to half the current value of Cwnd (maintained in bytes) and begins again in slow-start mode with an initial window of size 1. The slow-start phase lasts until the Cwnd reaches ssthresh and then congestion avoidance takes over. In this phase, the sender increases its Cwnd linearly for every new ACK it receives by following:

$$cwnd = \frac{MSS \times MSS}{cwnd} \qquad [127]. \tag{4.2}$$

Note that with TCP Tahoe the sender might retransmit packets which have been received correctly due to go-back-n classical model. Timeouts are used as the means of last resort to recover lost packets [127]. Mostly, the incident of three DupAcks considers the main reason behind low throughput, in our simulation, we came across continuous DupAcks which causes considerable delays, and slows the process of smooth transmission of TCP flows. Slow-start and congestion avoidance mechanism restrain the overall throughput. The slow-start curve plunges sharply into congestion avoidance mode, where it increments by one $1MSS$ per Round Trip Time (RTT). As a result, it causes more delays and degradation in throughput. Table 4.3 depicts the statistical information collected from simulations in NS3. It explains TCP Tahoe behaviour when Xor network coding is used. The implementation of Xor network coding is based on COPE [12]. In Tahoe, the slow-start function is counted to be 10% of overall called components of TCP Tahoe. However, congestion avoidance in Tahoe was 88%, it indicates the problem with TCP Tahoe in Xor network coding networks. Therefore, we can conclude that the main issue with Tahoe is the lack of adaption mechanism to accept interaction of intermediate nodes to perform Xor network coding. It is caused by misinterpretation of the role of Xor network coding in the network. Meanwhile, DupAck was only around 2% as Figure 4.1 depicts. It shows how the Dupack of two Acks were entirely consistent in our simulation which it meets the beliefs about the inefficiency of Tahoe in literature.

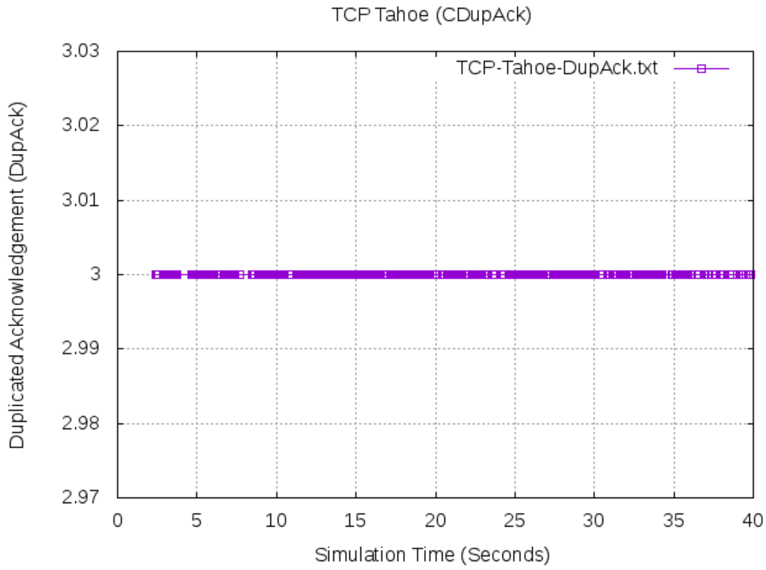


Figure 4.1: DupAck in TCP Tahoe.

4.3.2 TCP New Reno

New Reno is improved version of Reno that was released in 1990 BSD Reno. The Reno TCP sender only retransmits a packet after a retransmit timeout has occurred, or after three duplicate acknowledgements have arrived which triggers the fast retransmit algorithm. A single retransmit timeout might result in the retransmission of several data packets, but each process of the fast retransmit leads to the retransmission of only a single data packet. In other words, two problems arise with Reno TCP when multiple packet losses occur in a single window. First, Reno will often take a timeout. Second, even if a retransmission timeout is avoided, multiple fast-retransmits and window reductions can occur. When multiple packet losses occur, if the SACK option (Selective Acknowledgement) is on, the TCP sender has the information to make intelligent decisions about which packets to retransmit and which packets not to retransmit during fast recovery [128].

However, TCP New Reno improvements considered both the fast retransmit and fast recovery algorithms. The TCP sender can conjure from the arrival of duplicate acknowledgements, whether multiple losses in the same window of data would have occurred, and it can avoid taking a retransmit timeout or making multiple congestion window reductions due to such an event. The New Reno improvements apply to the fast recovery procedure that begins when three duplicate Acks are received, and it ends when either a retransmission timeout occurs, or an ACK arrives that acknowledges all of the data which were previously sent, besides it includes the data that was outstanding when the fast recovery procedure began [128].

Surprisingly in New Reno, neither full ACKs nor Partial ACK affects or contributes to the overall throughput. Furthermore, full ACK and partial ACK did not incur any delays to packets

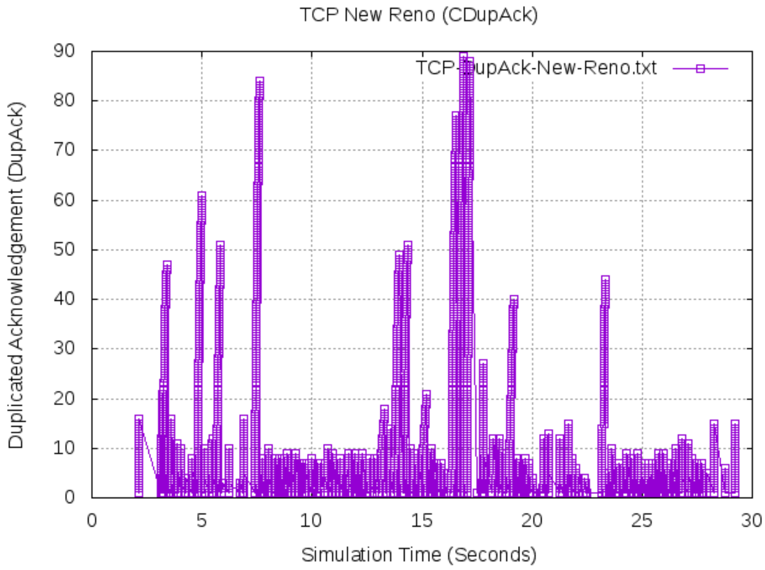


Figure 4.2: DupAck in TCP New Reno.

or affect the slow-start. The slow-start could be increased to lift the restrains of permitted data on the medium. However, it can result in duplicate ACK. Only congestion avoidance and DupAck are evidenced to cause a disturbance within New Reno. They decrease total throughput in two or more hops as Table 4.4 depicts. Table 4.4 shows the statistical information of all new Reno components in our simulation. Congestion avoidance was counted to be the first calibre of 77% in New Reno, it was the most called function in simulation time. The second called function was duplicated ACK in fast recovery mode, it is unlike Tahoe where duplicated Ack itself was the least called function where it was only counted for 2% of the whole simulation. In New Reno, duplicated ACK goes to the quick remedy of fast retransmission to save time for waiting for the retransmission timer to expire. Duplicated ACK in new Reno was accounted to be 16% of simulation time, Table 4.4 shows the statistics in real simulation time. In Table 4.4, New Reno has 77% of simulation time is being used in congestion avoidance mode, while the slow-start was only 2% of all simulation time. It is clear that the inefficacy of the congestion avoidance mechanism crippled most of New Reno operations. Moreover, 16% of new Reno simulation time was spent on Duplicated Acknowledgement (DupAck) within fast recovery mode, which represents the last known sequence number of Ack which, it might be lost, or it comes out of order. DupAck is triggering the fast recovery mode to send a packet which it was not acknowledged yet, or it comes out of order between other sequential ACKs. Moreover, figure 4.2 explains the incidents why there was 16% of DupAck incidents in simulation time. It describes the problem with throughput drains and the high ratio of RLP. Figure 4.2 shows the increased number of DupAck packet that lost, or it was out of order, which it triggers fast recovery mode. The DupAck was regularly occurring until 30 seconds of simulation time before it stopped.

4.3.3 TCP Westwood and TCP Westwood+

TCP Westwood [129] is a sender-side modification based on the TCP Reno protocol stack, which improved the performance of congestion control (Cwnd) in TCP for both wired and wireless networks. Furthermore, TCP Westwood is designated on an end-to-end bandwidth estimation to set the Cwnd and slow-start threshold after a congestion occurrence, or after three duplicate ACKs, or a timeout. The authors in [129] noticed that low-pass filtering estimated the bandwidth for the rate of returning acknowledgement packets. Unlike TCP Reno, which hastily halves the Cwnd after three duplicate ACKs, TCP Westwood strategically configures a slow-start threshold and a CW by taking into account the bandwidth used at the time of previous congestion experience. Authors in [129] claimed that TCP Westwood significantly increased throughput over wireless links and fairness compared to TCP Reno, and New Reno in wired networks [129]. TCP Westwood+ is the upgraded version of TCP Westwood, which is an end-to-end bandwidth estimation for setting control windows after congestion. The novelty of Westwood+ is its sole propriety of the available end-to-end bandwidth estimation algorithm. TCP Westwood bandwidth estimation algorithm did not work well in the presence of reverse traffic due to ACK compression [129]. In our work, we do not use any ACK compression for reverse traffic. Therefore, we use Westwood solely. Table 4.5 Westwood shows two great percentages, first is the congestion avoidance mechanism which it was accounted to be 62% whereas DupAck was around 30%. However, slow-start was only 5% in Westwood, and it is the lowest value when it compares to Tahoe in table 4.3 and New Reno in table 4.4.

4.3.4 WeNC-TCP

The proposed WeNC-TCP (Westwood-based COPE Network Coding-TCP) is depicted in figure 4.3. It has the framework of TCP Westwood. However, it is designed to accommodate Xor network coding such as COPE [12]. WeNC-TCP starts when the sender begins sending its data packets to potential recipients, and upon receiving corresponding acknowledgements of data packets. If the new Ack is in fast recovery mode, the Cwnd is increased by 1.5 times and is based on the theoretical throughput gain of network coding in the classical two-hop topology. If the received Ack is not in fast recovery mode, it will be challenged with comparison against the threshold value. If the threshold exceeds the current Cwnd (slow-start), it will enter congestion avoidance mode. As a result, Cwnd will assign one segment size (MMS). This helps to rectify any congestion, and it reduces the injection of a new segment into the network. Otherwise, it will be calculated differently as in equation 4.3. This equation will provide a gradual increase by 75%, more than 75% will cause the Cwnd to fall back into the congestion state. Therefore, WeNC-TCP is configured with the equation 4.3, to ensure a gradual increase of 75%.

$$Cwnd = ssThresh \times \frac{1+\sqrt{5}}{2}. \tag{4.3}$$

If the threshold exceeds the current Cwnd (slow-start), it will enter congestion avoidance mode. As a result, Cwnd will assign one segment size (MMS). This helps to rectify any congestion and it reduces the injection of a new segment into the network.

In cases where triple-duplicated acknowledgement (DupAck) received at the sender side, WeNC-TCP has to check if the counter of DupAck scored three incidents of duplicated Ack, and WeNC-TCP is not in fast recovery mode. If so, the WeNC-TCP will then check if the Cwnd exceeds the threshold value. If Cwnd reached the threshold value then Cwnd is calculated based on the last value of threshold multiplied by the value of (1.618) which is numerical value of $\frac{1+\sqrt{5}}{2}$ as in 4.4. As mentioned previously, this value is providing with gradual increase 75% rather than doubling the Cwnd. The reason behinds previous equation is to make sure Cwnd does not fall back drastically into congestion, and also it will assist the Cwnd to grow gradually rather than exponentially. If Cwnd does not threshold value, simply WeNC-TCP will end the DupAck process.

$$Cwnd = ssThresh \times \frac{1+\sqrt{5}}{2} \tag{4.4}$$

However, in the case of triple-duplicated acknowledgements in fast recovery mode, the Cwnd is multiplied by two segment size as follows:

$$Cwnd = SegmentSize \times 2 \tag{4.5}$$

It is vital to notice that WeNC-TCP deals with fast recovery in the first of its action, and this ensures smooth (no disturbance) Ack reception. Additionally, the feature mentioned above reduces three duplicated Acks incidents. As a result, WeNC-TCP will multiply Cwnd with two-segment size, to retransmit the lost packets quickly and exits the process of DupACK.

In case of retransmission, estimation of the current bandwidth has to take place first, and the bandwidth will be estimated based on Ack packets and TCP packet's header information fields. Afterwards, the threshold has to assign the maximum value of either two-segment size or current bandwidth estimation multiplied by minimum Round Trip Time known as in equation 4.6 .

$$ssThresh = \max(SegmentSize \times 2,\ BW \times minRTT) \qquad [129] \tag{4.6}$$

This equation is the foundation of Westwood. It is the cornerstone of Westwood ability to estimate what best value of threshold established by the estimation of bandwidth and related lowest RTT known as in original Westwood [129].

Upon transmission time-out timer (RTO) expires, the Cwnd is adjusted with one segment size and the threshold is assigned the value of estimated bandwidth (BW). The estimation of bandwidth is calculated by equation 4.7 in Westwood. It is the product of segment size multiplied by the number of acknowledged segments divided by round trip time (RTT).

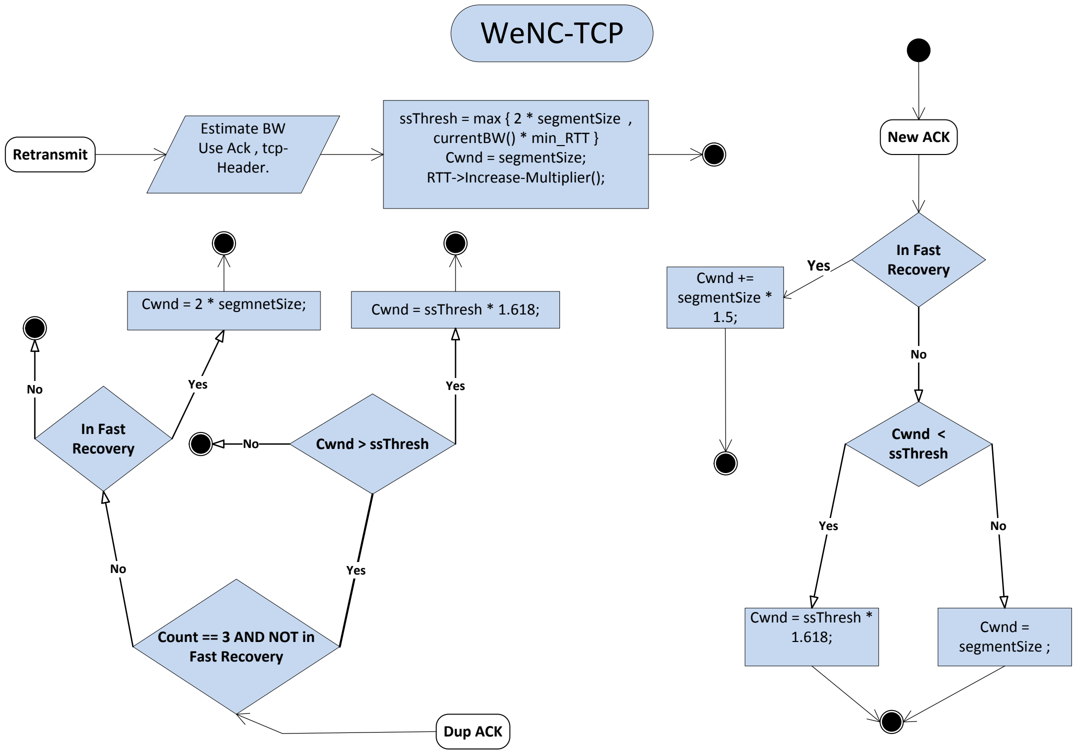


Figure 4.3: Proposed WeCN-TCP model.

$$BW = \frac{SegmentSize \times ackedSegments}{RTT} \quad (4.7) \quad [129]$$

Finally, RTT is doubled by the function of a multiplier, and that because of setting a new time for any possible RTO incident. The multiplier will double the RTT time for RTO to be checked and determine if it expires, which means there is a lost packet and it needs to retransmit again to the receiver.

In our simulation, we noticed that DupAck is more likely to have happened when Xor network coding is being used: it frequently happens in Tahoe, New Reno and Westwood. DupAck results in degradation of throughput and causes the RTO timer to expire. Therefore, more emphasis is put on the process of three duplicated Acks condition. Moreover, it causes TCP to enter retransmission mode. As a result, RLP ratio will be higher in Tahoe, New Reno and Westwood.

4.4 Performance Metric and Simulation Settings

In order to evaluate the algorithm's impact on TCP performance, the evaluation metrics are similar to chapter 3 section 3.4. The metrics are stated briefly as:

NS3 settings	Value
Number of Packets	5000000
Distance between nodes	250 m
Running Time	180 Seconds
Number of Runs	10 times
MAC layer Queue Capacity	10 frames
Test Protocol	TCP

Table 4.1: Simulation settings.

1. Throughput: this is the number of packets successfully received by the destination with respect to time (bits/s)

$$T = \frac{Number\ of Received\ packets \times PacketSize \times 8}{Simulation\ time\ (Seconds)} \tag{4.8}$$

2. RLP: this is the ratio of the number of packets lost and the total number of packets sent on the medium.

$$RLP = \frac{The\ number\ of\ packets\ lost}{The\ total\ number\ of\ packets\ sent} \tag{4.9}$$

3. Delay: this refers to the amount of time it takes a packet to be transmitted from source to destination.

4. Jitters: these are defined to be a variation in the delay of received packets.

4.4.1 Simulation Settings

The simulation was conducted using the Network Simulator 3 package (version ns.3.27). COPE was implemented and added to NS3 source files. The simulation environments are described as follows. IEEE 802.11 DCF with CSMA/ CA was adopted as the MAC layer protocol. Each node was equipped with a radio transceiver and a MAC-layer FIFO transmission queue of size 64 packets. We used ErpOfdm Physical layer mode [119] and with different data rates range from 10 to 24 Mbps. Fixed RSS Loss Model (RSS) [120] is applied to the received power which is constant independent of the transmit power, and it is set to -80 dBm. Packets size are varied from 100 to 1500 bytes. Furthermore, the number of packets is more than 1 million for each flow. In general, we use a data rate of 20 Mbs to test TCP. For each simulation scenario, 3 to 14 nodes are placed in a chain topology. CBR traffic flows were injected into the networks from the sources. Each flow uses a data rate of 20 Mbps unless specified otherwise. The topology used consisted of multi-hop with two flows in reverse direction, figure 1.3 in chapter 1 depicts the used topology.

The Friis Propagation Loss Model model [121] was applied to estimate the signal power received by the receiver. The path loss exponent, which is a parameter of the shadowing model, was set to 4. Wifi channel in ns3 was set to constant Propagation delay model. In the MAC layer setting is set to default without the quality of services. Moreover, MAC type is on Adhoc mode. All nodes use Constant Position Mobility Mode. Therefore, all nodes are set in stationary positions on the edge of each other. Again, at the network layer, we used static routing tables such in chain topology, to boost coding to follow COPE operations

The TCP protocol is configured by using ns3::TcpL4Protocal class in ns3. It started with specifying the TCP socket buffer in both sender and receiver. It has the capacity of 524288000 bytes. The Slow Start Threshold is set to the default value, and the initial cwnd is set to 3., and the TCP socket type is determined by choosing from different varieties from Tahoe, new Reno, and Westwood for every TCP test. Moreover, TCP sender is assigned different ports for each flow such as 80 or 85, later assigned destination address and activating TCP socket factory in TCP sender side by creating a socket on the sender for application. TCP Congestion Window is tracked all time using OutputStreamWrapper class and pointer in ns3, and all changes in Cwnd is print in files for further analysis.

In our simulation, we tested (Tahoe, New Reno and Westwood) TCP variants and WeNC-TCP in a saturated environment with an increased number of nodes and TCP flows. Traffic is generated at constant rates using NS3 features of TCP socket factory, with at least two bidirectional flows. Other settings are summarised in table 4.1.

For every simulation run, the seeds are configured by using changing global seed and re-run the simulation. These values are stored in two ($ns3::GlobalValue$) instances: $g\_rngSeed$ and g_rngRun. We used the class ($ns3::RngSeedManager$) provides an API to control the seeding and run number behavior. This seeding and substream state setting must be called before any random variables are created; e.g:

- $RngSeedManager::SetSeed(7)$; // Changes seed from default of 1 to 7.
- $RngSeedManager::SetRun(100)$; // Changes run number from default of 1 to 100.

The simulation results were collected using ns3 flow statistic tool. The above settings have been applied to all scenarios. The simulation results were recorded in a spreadsheet, and the data has been plotted in graphs using QiPlot and MATLAB. Thus all information is gathered and presented in figures. for example, each The graphs show throughput, packet loss ratio, delay and jitters values, and packets numbers (received and transmitted).

4.5 Results and Discussion

4.5.1 Results

We tested three implementations of TCP Tahoe, New Reno and Westwood in using COPE [12] as along as our WeNC-TCP. Throughput has not improved significantly in WeNC-TCP , it was marginal around 3-5 %. In figure 4.4, it is clear the throughput values for all TCPs and proposed WeNC-TCP are close in values, and skewers in values are extremely small. Therefore, Box-plots of all TCP are small due to this fact. The variance of value is relatively small , and unnoticeable.

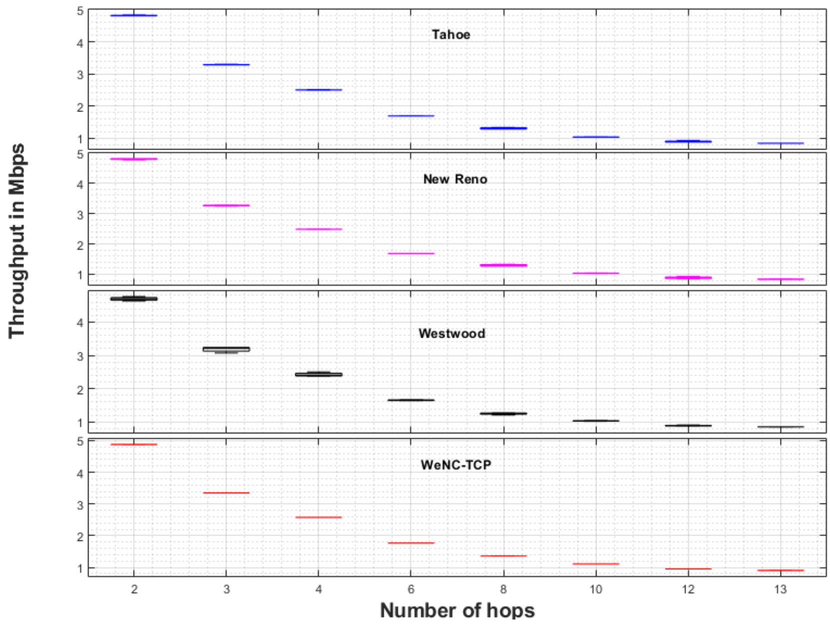


Figure 4.4: Number of hops vs WeNC-TCP throughput.

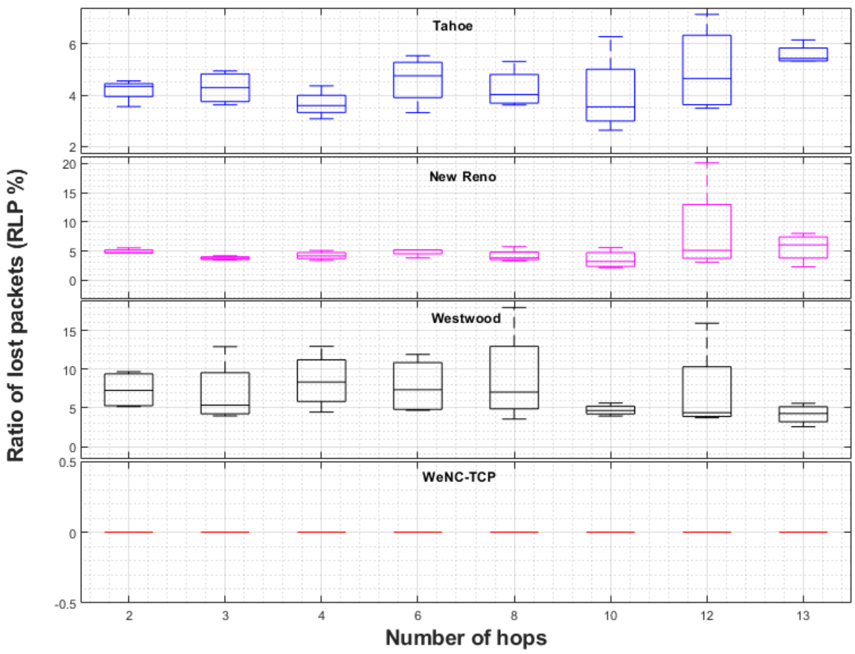


Figure 4.5: Number of hops vs WeNC-TCP RLP.

However, The maximum value was recorded for WeNC-TCP at 4.9 Mbps. As number of hops increased the throughput curved down for all TCP versions, only marginal advantage for WeNC-TCP shown in figure 4.4. Although, Cwnd of WeNC-TCP did not reach maximum limit of 65000 bytes, which means throughput could be improved further by injecting more packets into the

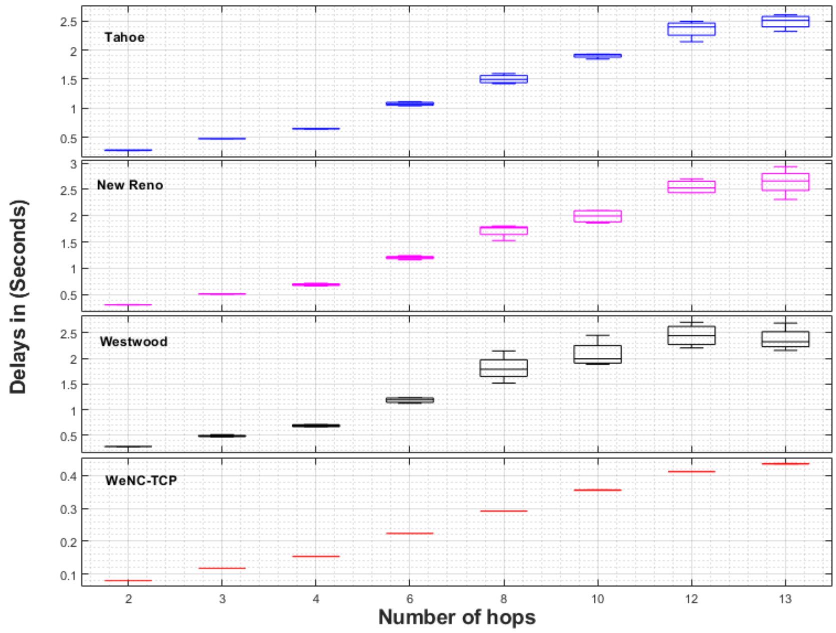


Figure 4.6: Delays of WeNC-TCP vs number of hops.

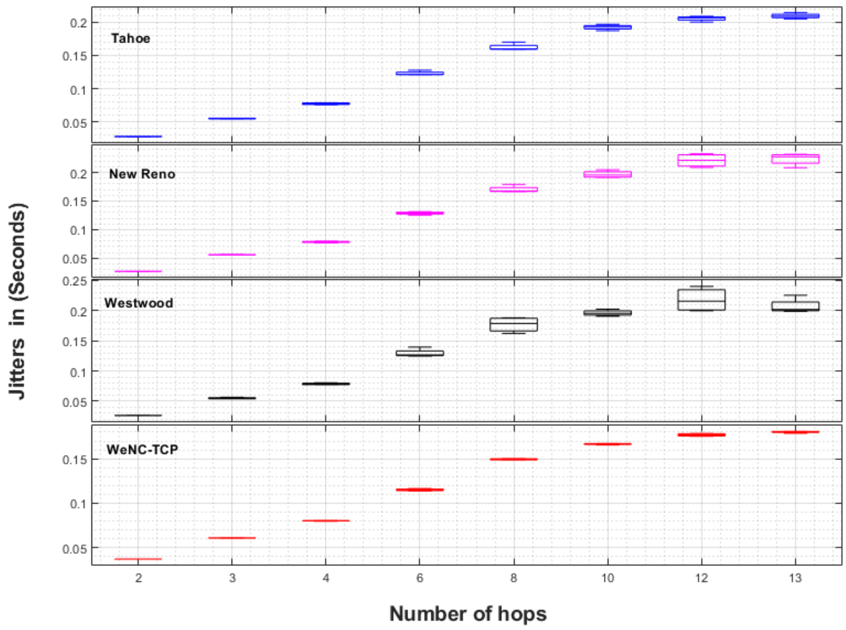


Figure 4.7: Jitters of WeNC-TCP vs number of hops.

network. However, due to misinterpretation of medium condition at lower layers, the probability to use the available resources are remote possibilities. Moreover, the principle of piggybacking makes it difficult to reach high throughput.

The best of WeNC-TCP can be seen in case of ratio of lost packets(RLP) shown in figure 4.5, it has the best reliability of zero lost packets. The zero RLP ratio was because of no DupAck or Retransmission incidents were recorded. And that explained The RLP is being closed to zero. Threshold of 500 to 2000 bytes in TCP WeNC-TCP contributes to reduce the DupACK incidents, and that resulted in, delaying enter congestion avoidance state in WeNC-TCP, and it controls the increase before entering a critical point of congestion. Such behaviour will help to minimise the loss packets ratio.

TCP Westwood has taken most of heavy burden of RLP drains, while TCP New Reno and Tahoe have average lost of only 5%, TCP Westwood has average RLP of all hops around 7 %. In box-plots of Tahoe, Westwood and New Reno were skewed up as a sign of potential increase of RLP after 8th hop, depicted in figure 4.5. It is understandable that as path increases the more packet loss is likely to be happened. Tahoe is more likely to have more packets loss because it lacks of selective packet retransmission and fast recovery mechanism, which is not based on the model of go-back-n principle (prematurely). But Westwood has proved to be worse than others TCPs, although Westwood has mechanism to adapt the condition of medium and readjust its sending rate based on available bandwidth, it did not prevent or reduce the RLP. Furthermore, It was not superior as author in [129] claimed to be better than Reno in term of throughput and fairness, when compared to TCP Reno over wireless links. The author claimed that TCP Westwood would choose better CW values based on bandwidth estimation after congestion occurred. It is undoubtable that Westwood does not fit well when Xor network coding is applied. Despite New Reno hiccups at 12th hops, TCP New Reno has slightly better RLP compares to Tahoe and Westwood. The incident of single value of TCP New Reno's RLP is resided out of scope, it does not impact having good RLP as it can be viewed in figure 4.5. On other hand, Delays for all TCPs are varied and were within normal distribution shown in figure 4.6. For example, TCP Tahoe , New Reno and Westwood box-plots were nicely centred and increased as number of hops increased. Except one incident of TCP westwood at 10th hop, where some values skewed up and above average and median value of 2 seconds. It suggested that delay could went up. In case of WeNC-TCP, the skewed and variance of values were very small in values. Although, the maximum delay it could reach 0.4 second. And it has improvement of 100% compares to other TCPs. Delay of WeNC-TCP increased gradually as number of hops increased from less than 0.1 seconds in at 2nd hop to the maximum of 0.2 at 13th hop. Improvement of WeNC-TCP was around 80-82% when it puts against other TCPs. Delay was reduced because of regulating the level of critical congestion and smooth injection of TCP packets, and that can be observed in whole simulation time, for example slow-start and congestion avoidance were calculated to be 50% each equally shared.

The jitters of all TCPs, the box-plots of all TCPs were hardly skewed, and were small in size as shown in figure 4.7. A lone incident at 12th hop of Westwood, and it was pointed to skewed upwards, signalling as sign of higher jitters value. The behaviour of all TCP remained consistent when the jitters and the number of hop have increased, this can be examined in Figure 4.7. WeNC-

TCP jitters have soared up as number of hop increased, with only small or indistinguishable difference in values. And that was believed to be the same of all TCPs. The maximum value of jitters wound not exceed the 0.2 seconds of all implementations, and the highest value was recorded on TCP New Reno at 0.22 second. Where the minimum was seen in Westwood at 0.026. WeNC-TCP has the lowest average jitters of 0.121 second, where the maximum was 0.25 in Westwood. WeNC-TCP performance at first two hops was lacking behind other TCPs, It was only improved at 4th hop. It was again marginal at maximum if 12% improvement. WeNC-TCP enhanced jitters around 8% in Tahoe, and 11% in Westwood, and finally 12% regarding New Reno. To summarise, all WeNC-TCP attained improvements are in table 4.2, it represents the number of hops and improvements in case of RLP, delay, jitters, throughput. Some values in jitters column replaced with (-) which represents no improvement in perspective column. Finally , WeNC-TCP performance excellency was because of good adaption of better engagement of Xor network coding and TCP sending rate. Furthermore, reducing the impact of congestion avoidance and emphasising more on slows-start had given WeNC-TCP superiority over other studied TCP versions. With such outcomes, WeNC-TCP could be implemented for applications where the need for low and short delay variation can be required on internet or a network of multi-hop nature with prolonged path. In addition, reliability of WeNC-TCP was proved to be near zero loss, it could be applied to a network where reliability is crucial factor, and required high reliability standard.

4.5.2 Discussion

When we look into table 4.6, WeNC-TCP has the extraordinary influence to reduce congestion avoidance effect in our simulation by putting more emphasis on slow-start mode rather than congestion avoidance. Therefore, the slow-start and congestion avoidance were calculated to be 50% each. Meanwhile, there was not any DupAck incidents. To observe the congestion window in WeNC-TCP Figure 4.14 illustrates the behaviour of the congestion window (Cwnd) value in this thesis. When it compares with other Tahoe, New Reno and Westwood. Besides, figure 4.16 shows the RTT against RTO timer, it hardly goes up, or it fluctuates, RTT of WeNC-TCP does not exceed the limit of RTO, which it means there was not an incident of re-transmission due to packet loss, which RTO timer would trigger the mode of re-transmission of TCP segment. However, Tahoe in figure 4.18 has rigid RTT and RTO; they are presenting the fact of why Tahoe suffers more DupAck and higher RLP ratio. New Reno's RTT and RTO are much similar to WeNC-TCP, although it has very few re-transmission incidents as figure 4.17 represents.

Finally, Cwnd of all TCP (Tahoe, New Reno, Westwood) are depicted in figure 4.9, figure 4.11 and figure 4.13. Tahoe and Westwood had similar congestion window behaviour but was not identical. However, Westwood had much intense Cwnd than Tahoe, and Westwood's Cwnd was spikier fluctuation than Tahoe. Moreover, Westwood's Cwnd had hardly exceeded the threshold of 20000 in two hops network. On other hand, New Reno had much higher Cwnd values with more spiky values which it exceeded the threshold, but it plunged back over simulation time. WeNC-TCP is much smoother than others. However, it looks as long compressed values over simulation time. Moreover, it has not gone over the threshold of 20000. The Number of coded packets WeNC-TCP is much lesser than other TCP variants, and it is due to regulating congestion instead of increasing sending rate, and it soothes the TCP flows, again it affects the number of coded packets as figure 4.15 illustrates. The point of WeNC-TCP is preventing packet loss and improving throughput in Xor network coding without incurring any delays to packets. Therefore, coding opportunities might reduce in case of two hops. In case of more than two hops, WeNC-TCP can find more coding opportunities, and consequently, it can improve throughput and RLP. Meanwhile, figure 4.8, figure 4.10 and figure 4.12 show the number of coded packets in simulation time of two hops. New Reno had the highest number of coded packets of 2500 packet. Tahoe and Westwood have around 1800 to 2000 coded packets. Although there were more chances to find matching pairs, it did not improve throughput or mask the number of lost packets. We can conclude this, having higher coding rate does not necessarily improve throughput's performance or reduce packet loss as we can see in case of TCP Tahoe, New Reno and Westwood. What matters is that right packet at the right time at right place matters. This conclusion is similar to the explanation given by authors in [7] about there no need for packets to wait in the queue until to find coding opportunity and need for coordination, it will affect the overall performance in the network. What this thesis confirms is that right time at right place matters.

It is undeniable that congestion avoidance affects the network coding process badly by restricting the number of useful packets – specifically native packets. If few native packets can be coded in an intermediate node at the right time, the ability of network coding to boost performance can be evident. On the other hand, WeNC-TCP has a touch-sensitivity to MAC-layer queue size and the maximum delay that can hold frames. It is the first version that needs better optimisation to be able to deal with variant queue size and different queue discipline. Another angle could be investigated to understand the role of the MAC layer on TCP when Xor network coding is used, such as in the [12] implementation.

4.5.3 Queue Size Impact

The impact of queue size has been investigated in various TCP implementations, such as Tahoe, Westwood and New Reno. Test results have shown lousy performance regarding throughput, delay and RLP. Simulation results have been analysed while setting the MAC queue capacity to 100 packets, and the MAC's waiting queue is set to 0.5 seconds. The TCP threshold is set to original settings. While TCP queue and packet size are set to the default value, 524,288 bytes and 100 bytes, respectively. TCP Westwood, using the setting as mentioned earlier, provided a throughput of 1.086 Mbps, which is less than the throughput value when the MAC queue is 50 packets.

Meanwhile, the RLP and average delays were almost 100% loss and more than 2.187 seconds

Improvement in				
Number of Hops	RLP %	Delay	Jitters	Throughput %
2	100	74	-	2
3	100	77	-	3
4	100	78	-	4
6	100	81	11	5
8	100	83	12	5
10	100	82	15	8
12	100	84	20	8
13	100	83	19	8

Table 4.2: Summary of Improvement in Multi-Hop in (%), (-) refers to no improvement.

TCP Tahoe	Slow Start	Cong. Avoid	DupAck	Retras
Average	10	1,489	34	4
Standard Deviation	96	909	23	3
Percentage %	10	88	2	0.214

Table 4.3: Tahoe statistics.

regarding delay, respectively. TCP Tahoe, however, had much better throughput (1.35 Mbps), and the RLP and average delays were around 4% and 0.440 seconds. On the other hand, TCP New Reno had marginally better throughput (1.38 Mbps) than Tahoe; however, the RLP and average delays were 12% and 1.211 seconds respectively.

When WeNC-TCP's MAC queue capacity is 100 packets, the throughput is similar to that of Tahoe's (1.36 Mbps), and the RLP is found to be 15%. The average delay of WeNC-TCP is much longer than all TCP variants at 1.5 seconds. However, when the slow-start threshold of WeNC-TCP is set to 20,000, the throughput becomes around 1.4 Mbps, while the RLP is more marginal at 1.4%. Moreover, the average delay was 0.54 seconds. If we apply the same TCP slow-start threshold to TCP new Reno, throughput would be around 1.4 Mbps with the RLP at 2.2%. Also, the average delay was more than WeNC-TCP – it was calculated to be 0.7 seconds.

COPE's queue does not affect overall throughput and end-to-end delay. It has a capacity of 600 packets, and is hardly full. No packets were recorded as dropped in the simulation. COPE's queue does not put any burden on the available resources, such as queue spaces, or cause any delay to WeNC-TCP, Tahoe or New Reno. On the contrary, the MAC queue has a higher rate of packet dropping due to frequent top-layer injections of packets at the MAC queue.

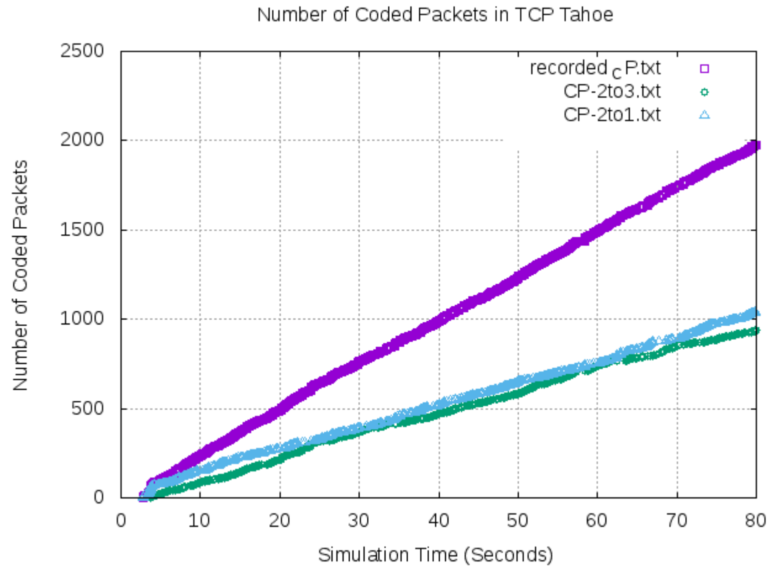


Figure 4.8: Number of coded packets in TCP Tahoe in two hops.

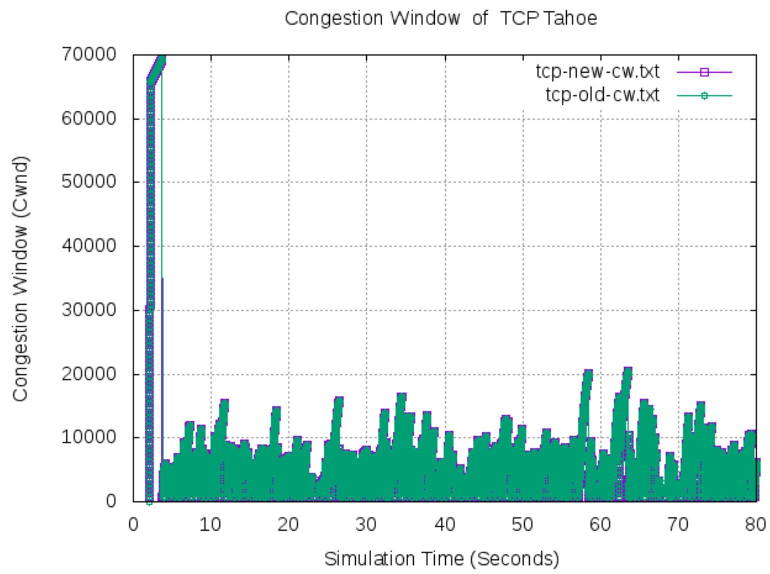


Figure 4.9: Congestion window in TCP Tahoe.

4.6 Summary

The current TCP functionality causes a drain in the network throughput when network coding is applied. The congestion avoidance algorithm affects the throughput negatively. It cripples the ability of the sender to inject packets based on its current mechanism of preventing congestion. Besides, wireless congestion avoidance could be easily misinterpreted as a congestion incident

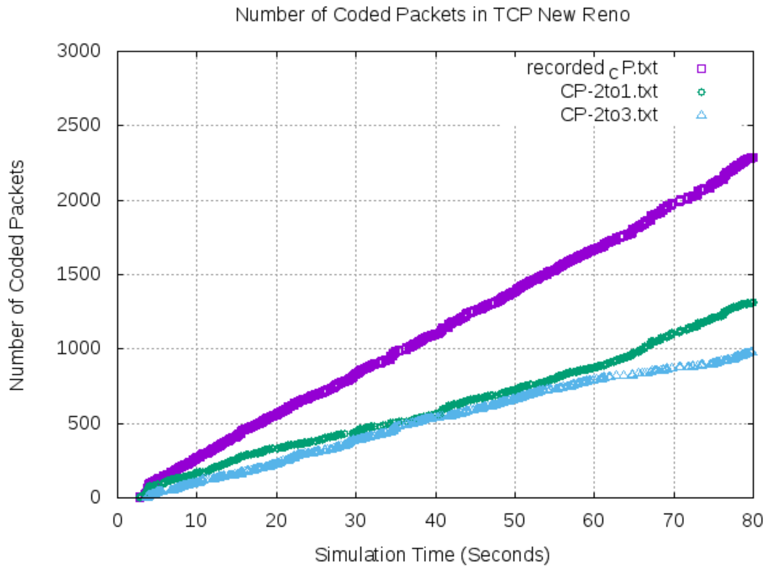


Figure 4.10: Number of coded packets in TCP New Reno in two hops.

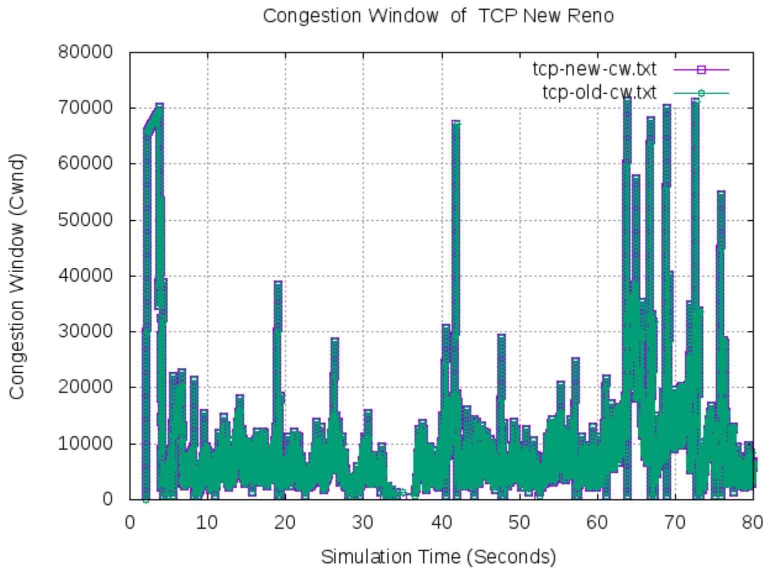


Figure 4.11: Congestion window in TCP New Reno.

due to link quality. However, the network Xor coding process is crippled by restricting the number of available native packets for finding a matching pair. In this work, we try to measure the difference between two different aspects: TCP reliability and network coding processes. The principle of current TCP cannot accept the role of the intermediate node, which finds the best match of a packet pair. WeNC-TCP achieves slightly better throughput than other TCP variants, although the best of WeNC-TCP can be seen in RLP, delays and jitters in two or more hops. Table

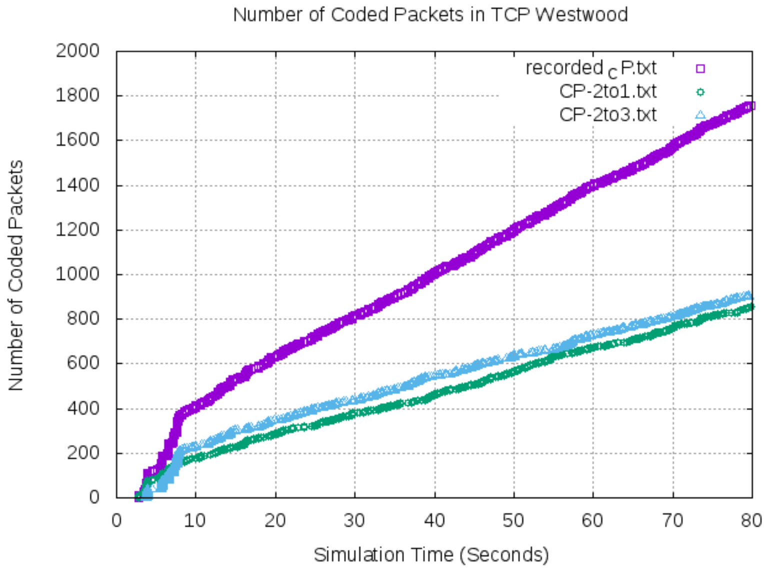


Figure 4.12: Number of coded packets in TCP Westwood in two hops.

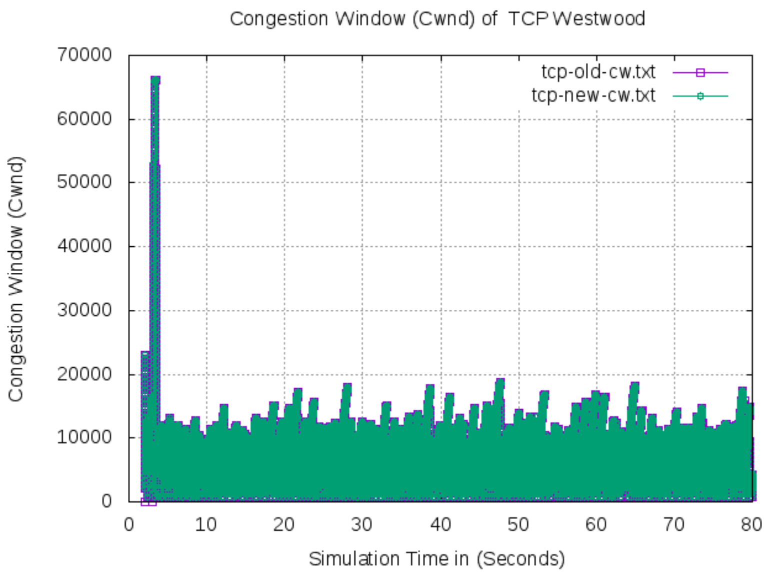


Figure 4.13: Congestion window in TCP Westwood.

4.2 has a summary of all achieved improvements over the increased number of hops. The need for better interaction between two layers such the transport and MAC layers becomes a necessity when network coding is applied in the wireless network. WeNC-TCP is still the first version to accommodate the need for network coding at the transport layer. It is far from perfect, and we can anticipate that the MAC layer could add a significant contribution to TCP flows when the network experiences different circumstances of packet loss, limited bandwidth or limited access

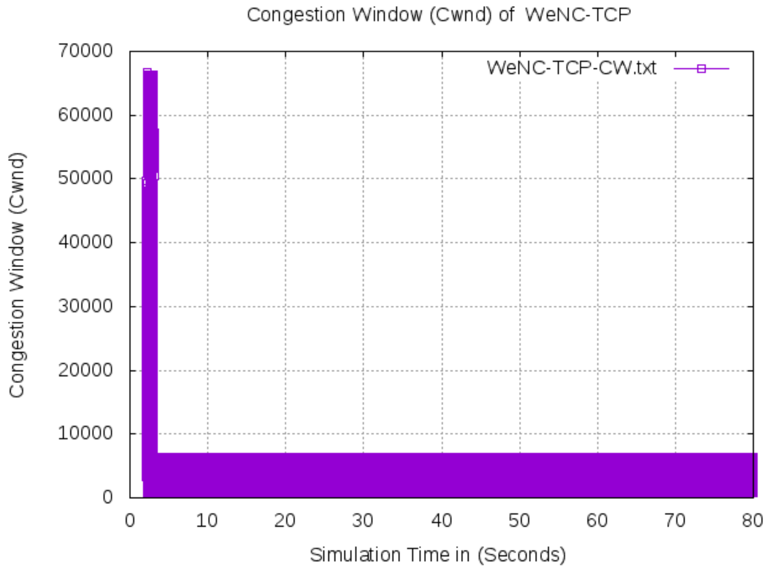


Figure 4.14: Congestion window in TCP WeNC-TC.

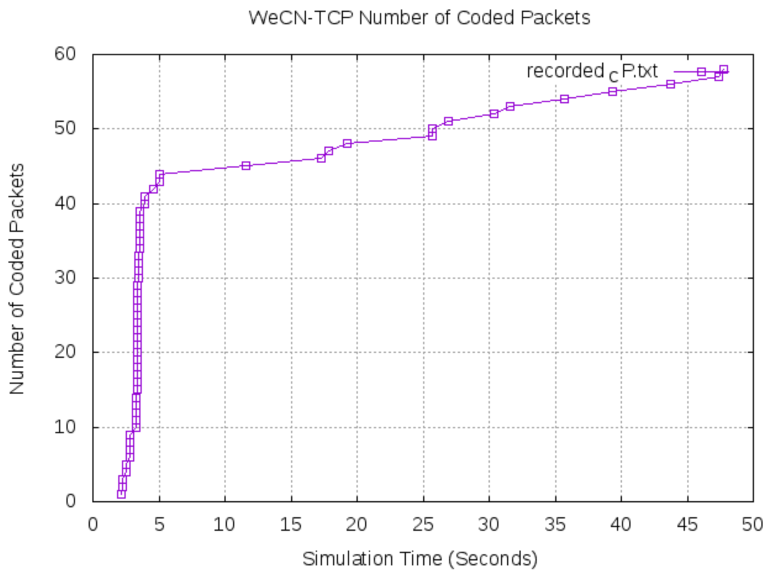


Figure 4.15: Number of coded packets in TCP WeNC-TC.

to the medium in a high contention scenario.

The main findings of this chapter are summarised as follows:

- The studied TCP variants (Tahoe, New Reno and Westwood) are affected by Xor network coding as they are not aware of the role of relays nodes in the network. Lack of transparency between the transport layer and the implemented Xor network coding at relay nodes can

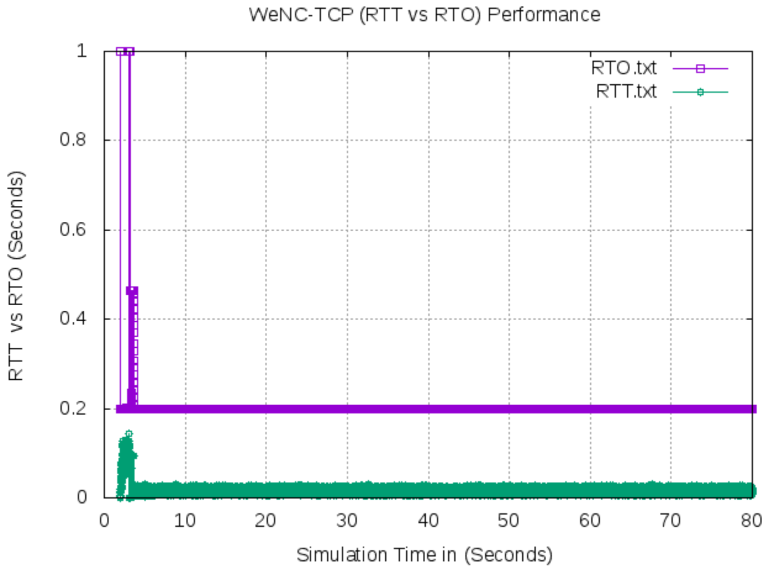


Figure 4.16: RTT and RTO performance in WeNC-TCP.

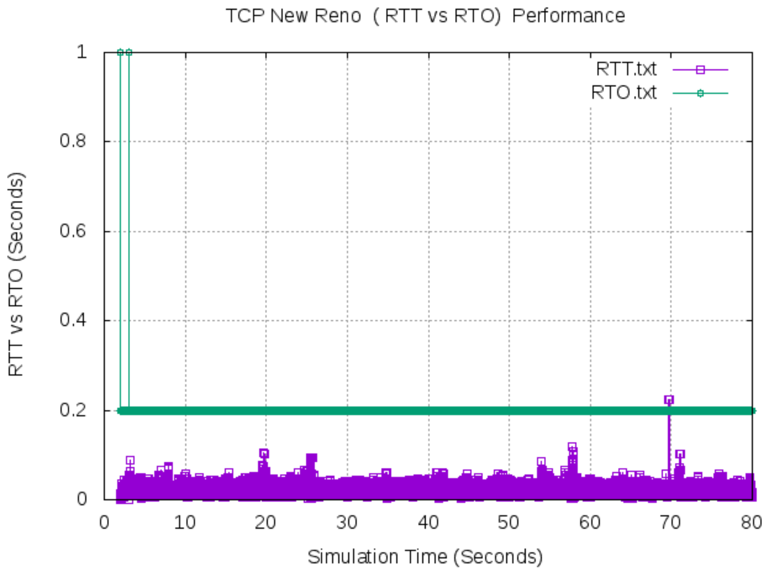


Figure 4.17: RTT and RTO performance in TCP New Reno.

affect throughput, RLP and delay.

- Network coding could improve TCP throughput if TCP is able to tolerate the core process of network coding, in which the role of relay nodes is to code TCP packets to reduce the number of transmissions and use the available resources for injecting more packets in medium, consequently this will improve throughput.

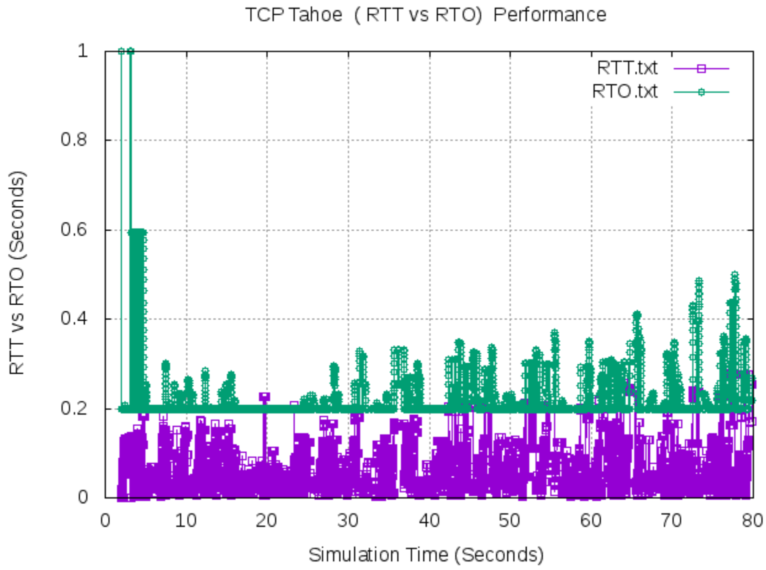


Figure 4.18: RTT and RTO performance in TCP Tahoe.

New Reno	Average	SD	Percentage%
Slow Start	41	25	2
Congestion Avoidance	1473	859	77
Partial Ack Fast recovery	21	14	1.1
Full Ack Fast recovery	38	24	2
DupAck	40	25	2
DupAck Fast recovery	300	180	16
Limited transmit Fast Recovery	0	0	0
Retransmission	3	2	0.15

Table 4.4: TCP New Reno statistics.

Westwood	Average	Standard Deviation	Percentage
Slow Start	160	72	5
Cong. Avoid	1891	1142	62
Fast Recovery	68	36	2
DupAck	903	6194	30
Retras	24	10	1

Table 4.5: TCP Westwood statistics.

- Although all TCP variants have higher coding rates and opportunities to find a matching pair, this does not improve throughput or mask any packet loss in term of RLP. What matters is that right packet at the right time at right place. Furthermore, the congestion avoidance at every TCP variant was the main culprit that affected both throughput and RLP. For instance, the congestion avoidance was the most called function in simulation

Westwood	Average	Standard Deviation	Percentage
Slow Start	166	97	50
Cong. Avoid	167	97	50
Fast Recovery	0	0	0
DupAck	0	0	0
Retras	0	0	0

Table 4.6: WeNC-TCP statistics.

time, Tahoe (88%), New Reno (77%) and Westwood (62%).

- The second issue of TCP when Xor network coding is applied is RLP. It has a negative impact on throughput and the smoothing of TCP flows, as it triggers more retransmission of some segments due to the duplicated ACK, or timeout that triggers retransmission. In New Reno duplicated ACKs accounted for 16% in new Reno. Westwood was more affected by duplicated ACKs, which were calculated to be 30%.

- WeNC-TCP has slightly better throughput and zero RLP, also, it has the lowest delay and slightly low jitter variance compared with other TCP variants. However, WeNC-TCP did not improve throughput significantly in network.

  WeNC-TCP has rewritten the effect of congesting avoidance to have a less restrictive impact on TCP flows. As a result, it has much better RLP and delay effects. For instance, the RLP was zero, and end-to-end delays have reduced to 80%.

- The slow-start has enjoyed a massive role in WeNC-TCP: it has been beneficial in term of throughput, delay and RLP, unlike other slow-starts in TCP variants (Tahoe, New Reno and Westwood) where it has played only a small role in improving overall throughput, delay and RLP. The slow-start was the second most called function after congestion avoidance, for instance, Tahoe called the slow-start up to 10%. In New Reno, slow-start was only 2%. However, Westwood's slow start was 30%, which is considered the highest of all TCP variants. Finally, WeNC-TCP's slow-start congestion avoidance were recorded to be around 50%.

- Jitters in WeNC-TCP have not been improved or reduced remarkably. Table 4.2 shows that the highest percentage improvement was in the 12th hop, up to 20%. However, jitters have marginally improved after the fourth hop. WeNC-TCP's jitters have similar behaviour to other TCP variants.

- Although, Cwnd of WeNC-TCP did not reach maximum limit of 65000 bytes, which means throughput could be improved further by injecting more packets into the network. However, due to misinterpretation of medium condition at lower layers, the probability to use the available resources are remote possibilities. Moreover, the principle of piggybacking makes it difficult to reach high throughput.

- The best of WeNC-TCP can be seen in case of ratio of lost packets(RLP) shown in figure 4.5, it has the best reliability of zero lost packets. The zero RLP ratio was because of no DupAck or Retransmission incidents were recorded. And that explained The RLP is being closed to zero. Threshold of 500 to 2000 bytes in TCP WeNC-TCP contributes to reduce the DupACK incidents, and that resulted in, delaying enter congestion avoidance state in WeNC-TCP, and it controls the increase before entering a critical point of congestion. Such behaviour will help to minimise the loss packets ratio.

- Finally , WeNC-TCP performance excellency was because of good adaption of better engagement of Xor network coding and TCP sending rate. Furthermore, reducing the impact of congestion avoidance and emphasising more on slows-start had given WeNC-TCP superiority over other studied TCP versions. With such outcomes, WeNC-TCP could be implemented for applications where the need for low and short delay variation can be required on internet or a network of multi-hop nature with prolonged path. In addition, reliability of WeNC-TCP was proved to be near zero loss, it could be applied to a network where reliability is crucial factor, and required high reliability standard.

CHAPTER 5

TCP-MAC Cross Layer integration in Coded NetworkS

Wireless Multi-Hop networks rely on the role of intermediate nodes to perform network coding. The superiority of network coding in the wireless environment is evident. Network coding exploits the nature of wireless as a broadcast environment, and that explains network coding superiority. Meanwhile, TCP is known for its friendliness as a congestion reliever in congested networks. Despite the friendliness of TCP, it is still a challenge when Xor network coding is applied. Furthermore, TCP is designed to guarantee the delivery of data packets. However, it does not behave as expected when network coding plays a part in the network. There is an absence of MAC-layer collaboration that could have positive effects on Xor network coding. In this chapter, the impact of MAC-layer vitality, and the TCP and MAC role are investigated to improve overall throughput, delays and jitters. This work proposes MACINT-TCP to improve network coding functionality and incorporate better interaction between TCP and the MAC layer to obtain favourable outcomes in term of throughput, delays and RLP. The role of intermediate nodes as means to perform Xor coding on matching pair can affect TCP. However, it is not the main culprit, and the delay is occurring due to increased access to the medium by all competing nodes. It will be shown that the current TCP behaviour is not favouring Xor network coding. We investigate the possibility of the role of the middle node to accommodate the coding process, and its relation when MAC priority as a technique is applied in the case of TCP bidirectional traffic.

5.1 Introduction

Network coding was introduced by Ahlswede et al [2]. Network coding concept focuses on using bandwidth efficiently to achieve higher throughput than traditional transmission theorem [130].

Packets are coded together at an intermediate node into one coded packet, which is then broadcast to the network. This reduces the number of transmissions and aids the flow of rich information. Therefore, bandwidth becomes available for new data to be transmitted [131]. Network coding depends on information from more than one OSI stack: incorporating two or more layers in OSI could benefit network coding, and could enhance the overall performance of wireless networks. If network coding is implemented in a single layer of a network stack, it could affect other layers' functionality, or leave other OSI redundant, with no positive effects. Authors in [132] describe how the process of network coding altered the payload of a frame in such a way that only the destination can decode the payload. OSI layers might not have enough information from payload alone. Therefore, the OSI layer may not be able to utilise their function. For example, network coding needs routing information or data content for frame priorities, but some functionality might be lost when network coding is implemented in a layer below the one responsible for routing or prioritisation [132]. In network coding literature, two solutions were promoted: a cross-layer approach [133], [134], [135], [136], [137], [138] and a specialised network coding layer [12].

The advantage of the cross-layer approach is that it takes into account efficient communication and better coordination between two layers or more. However, it has to change and alter some functionality to allow better cooperation. Therefore, the legacy of independent layers is relinquished. However, including a new layer for network coding seems to be a better solution, as in COPE [12]. Here, this was implemented between the network and the MAC layer. The new layer does not have to take on the functionalities of other layers. Moreover, it does not change the other layers (network or MAC layer).

COPE [12] was the first attempt to test the practicality of network coding. Although TCP is resilient, network-friendly, and eases congestion to guarantee smooth delivery of data packets it is not expected to enjoy the benefits of Xor network coding. It is not guaranteed to live up to the same expectations when network coding is being used in the wireless network. What would be the impact of the MAC layer on the overall throughput and RLP? A reception report is used by all nodes to coordinate the process of 'who has what packets'. Reception reports help nodes to decode packets and track the native and encoded packets on medium [12]. Moreover, Xor network coding is seen as a special case of random network coding, where the Galois field is of size 2 [139].

In the case of TCP, the need to have better interaction between the transport layer and the MAC layer is evident as network coding is performed by COPE [12].The TCP flows do not benefit greatly from the functionality of network coding. Therefore, the interaction between transport and MAC layers could influence the process of network coding and smooth TCP flows between layers. Accordingly, TCP traffic will enjoy more frequent access to medium and that will improve overall performance. On other hand, authors in [23] expressed an issue on the fairness allocated by the MAC layer which impacts the performance of the non-monotonic saturation. Using MAC propriety can eliminate this monotonic saturation, and can improve saturation throughput gains and fairness among flows rather than nodes.

5.2 Problem statement

COPE [12] was the first implementation to evaluate network coding practicality. Transport Control Protocol has not shown significance performance improvement when Xor network coding technique applied as in COPE [12]. For example, authors in [12] argued that TCP did not show any significant improvement regarding throughput (the average gain was 2–3%). Their explanation was based on the reaction to collision-related losses, as many nodes were sending packets to the bottleneck nodes (centred), but they were not within carrier sense range of each other (as a result of the classic hidden terminals problem). Therefore, many collision-related losses cannot be mended with the maximum number of MAC retries, which causes TCP flows to experience a 14% loss. Again, Katti et al. [12] noticed that nodes in their test-beds did not see enough traffic to make use of coding; most of their time was spent without any packets in their queues or just a single packet with no matching pair. When authors in [12] brought nodes closer together in sense range, COPE offered a marginal improvement in throughput.

In COPE, authors did not consider the MAC effects on TCP or cross-layer design, which could be crucial. Meanwhile, other works on cross-layer design focused on either using queue management models [110] or combining intersession and intrasession network coding [112]. However, the work in [113] argued that the inherent coordination problem of carrier sensing based random access in multi-hop wireless networks affected the performance in Xor network coding. They provided a theoretical analysis that improved throughput. Scholars in [113] modified the IEEE 802.11 DCF, based on tuning the back-off [113] mechanism using a feedback approach taken from the work in [140]. The work in [113] suggested altering back-off values regardless of which approach. Though our investigation showed that throughput was not greatly improved in a practical sense, but reliability could be improved superbly.

The previous works did not take into account the impact of MAC layer on TCP; they have not investigated MAC impact when MAC priority is considered at the intermediate node (bottleneck nodes) using TCP. On other hand, there was not any study actively considered the cross-layer design in case of TCP solely. Unlike other works in [1], [34], [41], [50], [116] which focused on UDP traffic solely. We investigate and apply MAC priority techniques ([1], [34], [41], [50], [116]) within TCP Tahoe, New Reno and Westwood. Therefore, we will examine the approach first introduced using MAC priority techniques Thus, we will look into how we can improve TCP throughput and reliability through MAC layer functionality.

5.3 System Model

This section describes the applied system model. The model has been implemented and tested. The implementation consists of two parts: TCP New Reno and TCP Westwood. The proposed

MACINT-TCP solution, which has the base structure of TCP Tahoe, is also described in this section.

5.3.1 TCP New Reno

The new Reno is an improved version of Reno that was released in 1990 BSD. In TCP Reno, the sender only retransmits a packet after a retransmit timeout has occurred, or after three duplicate acknowledgements have arrived, which trigger the fast retransmit algorithm. A single retransmit timeout might result in the retransmission of several data packets, while each process of the fast retransmit leads to the retransmission of only a single data packet. In other words, two problems arise with Reno TCP when multiple packet losses occur in a single window. First, Reno will often take a timeout. Second, even if a retransmission timeout is avoided, many multiple fast retransmits and window reductions can occur. When multiple packet losses occur, if the selective acknowledgement (SACK) option is on, the TCP sender has the information to make intelligent decisions about which packets to retransmit during fast recovery [128].

However, TCP New Reno improvements are on both the fast retransmit and fast recovery algorithms. The TCP sender can conjure from the arrival of duplicate acknowledgements, whether multiple losses in the same window of data would occur, and it can avoid taking a retransmit timeout or making multiple congestion window reductions due to such an event. The New Reno improvements apply to the fast recovery procedure that begins when three duplicate ACKs are received and ends when either a retransmission timeout occurs or an ACK arrives that acknowledges all of the data that was previously sent. Besides it includes the data that was outstanding when the fast recovery procedure began [128].

5.3.2 TCP Westwood and TCP Westwood+

TCP Westwood+ [129] is considered a sender-side modification based on the TCP Reno protocol stack, which improved the performance of congestion control (Cwnd) of TCP in both wired and wireless networks. Furthermore, TCP Westwood is designated on an end-to-end bandwidth estimation to set the congestion window and slow-start threshold after a congestion occurrence happens, or after three duplicate acknowledgements or a timeout. Authors in [129] noticed that low-pass filtering estimated the bandwidth for the rate of returning acknowledgement packets. Unlike TCP Reno, which hastily halves the congestion window after three duplicate ACKs, TCP Westwood strategically configures a slow-start threshold and a congestion window by taking into account the bandwidth used at the time of the previous congestion experience. Authors in [129] claimed that TCP Westwood significantly increased throughput over wireless links and fairness compared to TCP Reno, and New Reno in wired ne and wireless tworks [129]. TCP Westwood+ is the upgraded version of TCP Westwood, which is an end-to-end bandwidth estimation for setting control windows after congestion. The novelty of Westwood+ is its sole propriety of the available end-to-end bandwidth estimation algorithm. TCP Westwood's bandwidth estimation algorithm did not work well in the presence of reverse traffic due to ACK compression [129]. Westwood+ can work in the presence of reverse traffic using compressed ACK.

5.3.3 MACINT-TCP

The basic idea of the proposed solution is illustrated in figure 5.1. It shows the interaction between transport and MAC layers and where Xor coding is performed. COPE [12] was implemented between network layer and MAC layer, and it presented the Xor network coding process. The cross-layer design links Transport layer virtually with MAC layer, while network layer was not interact with upper or lower layers regarding the process of coding. Network layer was excluded and it only provided the path selections and packet forwarding including routing through intermediate nodes.

Moreover, the detailed flow chart for proposed MACINT-TCP (MAC Interaction-TCP) is depicted in figure 5.2, and has the framework of TCP Tahoe. However, it is designed to accommodate Xor network coding such as COPE [12]. MACINT-TCP starts when the sender begins sending its data packets to potential recipients and the receiving correspondent acknowledges the data packets. The congestion window (Cwnd) increases by one and a half times if it was less than threshold value.

$$Cwnd = Cwnd \times 1.5. \tag{5.1}$$

The multiplication of Cwnd is based on theoretical throughput gain of network coding in the classical two-hop example. The reason for multiplying Cwnd with one and half, it will helps the receiver to cope with the flow overload. It is decreasing the segment injection by 25% and that will help to reduce the incidents of duplicated packets.

If the threshold exceeds the current Cwnd (slow-start), it will be treated differently. First, Cwnd will be multiplied by the counter of congestion avoidance, which counts the occurrence of congestion in the network as follows:

$$Cwnd = SegmentSize \times Counter\_Cong\_Avoid. \tag{5.2}$$

Multiplication of the segment size by counter of congestion avoidance will encourage not to fall hard into congestion avoidance state for long equation 5.2, which was seen as crippling factor for Cwnd to grow gradually and use more available resources offered by coding at intermediate nodes. The next parameter will be adjusting ratio equation 5.3, the adjusting ratio is calculated by subtraction of old Cwnd and the normal calculated Cwnd, it will be used in the next calculation in 5.4 to boost up threshold, the new threshold will consider the number of times falling into congestion state and the adjusting ratio for better Cwnd in current state. Thus the time to recover from congestion will be shorten rather than prolonged for long period.

$$Adjust\_Ratio = \frac{OldCwnd - CurrentCwnd}{(OldCwnd + CurrentCwnd)}. \tag{5.3}$$

$$threshold = (Cwnd + Adjust\_Ratio) \times Counter - Cong - Avoid. \tag{5.4}$$

When three duplicated acknowledgements (DupAck) being received at the sender side, Cwnd is calculated differently to have two segment size (for simplicity this is assumed to be 1 MSS). Meanwhile, the threshold is calculated as follows:

$$ssThresh = MAX(cwnd, SegmentSize \times dupAck\_Counter) \tag{5.5}$$

The reason for using equation in 5.5 to make sure the threshold is located away from Cwnd but within the acceptable range before exiting DupAck state. Moreover, Thresh also ensure that after existing DupAck state the Cwnd will not return to congestion avoidance state gain.

$$ssThresh = MAX(\frac{cwnd}{2}, SegmentSize \times 2) \tag{5.6}$$

At the retransmission stage, the sender's Cwnd is increased by one segment size for every successfully transmitted lost packet. While multiplier is based on the last round trip time , it increases the round trip time for retransmission incident to recover lost packets. At the retransmission stage, the threshold is assigned the maximum value of either double segment size or division of Cwnd by two. It is important to make sure that threshold is slightly bigger than Cwnd current value and it helps to recover quickly from further congestion.

In our simulation, we noticed that DupAck is more likely to happen when Xor network coding is being used. It frequently happens in Tahoe and New Reno. Moreover, this results in the degrading of throughput and causes RTO (retransmission timeout) timer to expire. What makes it difficult to tackle is the concept of interaction of middle nodes and intersession network coding. However, our MAC improvement relies on the function of alerting DCF function of updating failed contention window (CW), as follows:

$$CW = (CW \times 0.618) + 3 \tag{5.7}$$

In the above equation, 0.618 was obtained after dividing success CW values by the sum of fail and success CW values. Similar to chapter 3, equation 3.1, MACINT-TCP assigns the minimum values for failed CW in nodes that are contending on accessing the medium. It serves middle nodes to access the medium more frequently but does not restrict edge nodes from accessing the medium evenly. On the other hand, MACINT-TCP assumes the sender and receiver have a large buffer, which is up to 52,428,800 bytes. Also, it restricts the slow-start threshold so it cannot exceed 4,000 bytes, and sets the value of initial CW to 3.

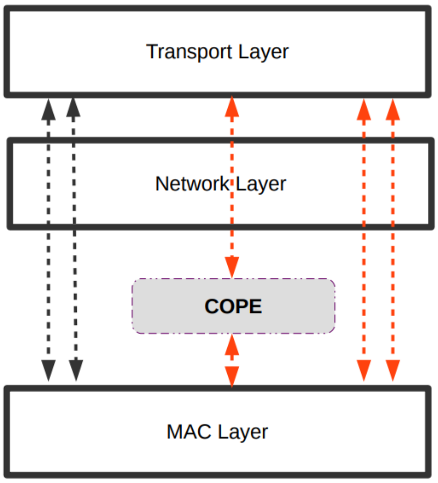


Figure 5.1: Proposed Solution for MAC and TCP interaction framework within network stack.

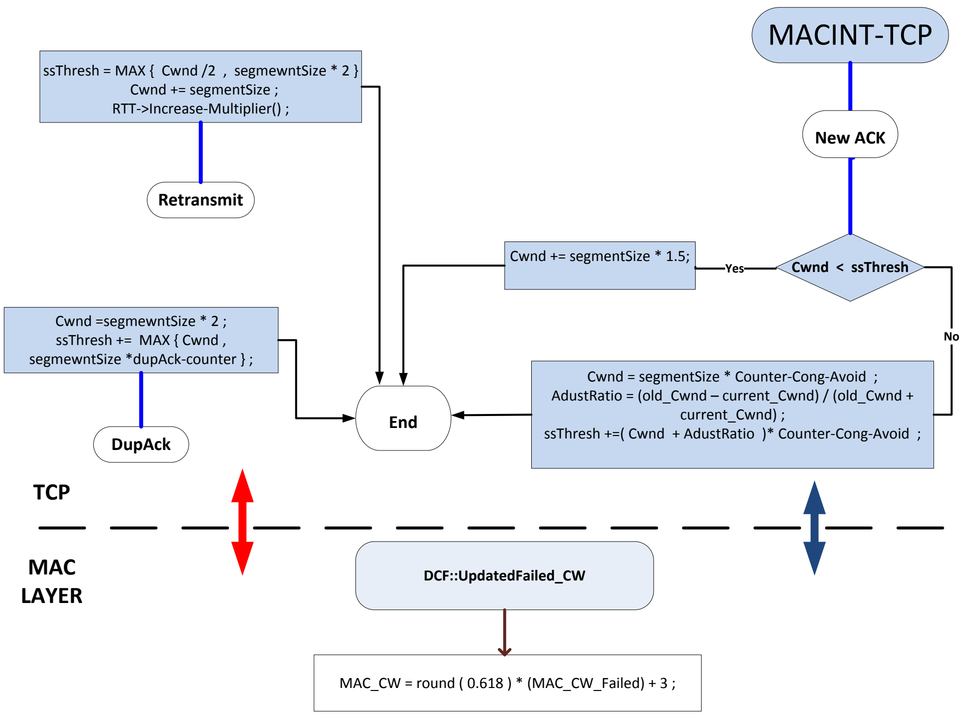


Figure 5.2: Proposed MAC and TCP layer interaction solution in Xor network coding flow chart.

NS3 settings	Value
Number of Packets	5000000
Distance between nodes	250 m
Running Time	200 Seconds
Number of Runs	10 times
MAC layer Queue Capacity	10 frames
Test Protocol	TCP

Table 5.1: Simulation settings.

5.4 Performance Metric and Simulation Settings

In order to evaluate the impact of the proposed TCP algorithm, the evaluation metrics are similar to chapter 3 and chapter 4 section 3.4 and 4.4. The metrics are stated briefly as:

1. Throughput: This is the number of packets successfully received by destination with respect to time (bits/s)

$$T = \frac{Number\ of Received\ packets \times PacketSize \times 8}{Simulation\ time\ (Seconds)} \tag{5.8}$$

2. RLP: This is the ratio of the number of packets lost and the total number of packets sent on the medium.

$$RLP = \frac{The\ number\ of\ packets\ lost}{The\ total\ number\ of\ packets\ sent} \tag{5.9}$$

3. Delays: This refers to the time taken for a packet to be transmitted across a network from source to destination.

4. Jitters: These are defined to be a variation in the delay of received packets.

5.4.1 Simulation Settings

The simulation was conducted using the Network Simulator 3 package (version ns.3.27). COPE was implemented and added to NS3 source files. The simulation environments are described as follows. IEEE 802.11 DCF with CSMA/ CA was adopted as the MAC layer protocol. Each node was equipped with a radio transceiver and a MAC-layer FIFO transmission queue of size 64 packets. We used ErpOfdm Physical layer mode [119] and with different data rates range from 10 to 24 Mbps. Fixed RSS Loss Model (RSS) [120] is applied to the received power which is constant independent of the transmit power, and it is set to -80 dBm. Packets size are varied from 100 to 1500 bytes. Furthermore, the number of packets is more than 1 million for each flow. In general, we use a data rate of 20 Mbs to test TCP. For each simulation scenario 3 to 14 nodes are placed in a chain topology. CBR traffic flows were injected into the networks from the sources. Each flow uses a data rate of 20 Mbps unless specified otherwise.

The Friis Propagation Loss Model model [121] was applied to estimate the signal power received by the receiver. The path loss exponent, which is a parameter of the shadowing model, was set to 4. Wifi channel in ns3 was set to constant Propagation delay model. In the MAC layer setting is set to default without the quality of services. Moreover, MAC type is on Adhoc mode. All nodes use Constant Position Mobility Mode. Therefore, all nodes are set in stationary positions on the edge of each other. Again, at the network layer, we used static routing tables such as in chain topology, to boost coding to follow COPE operations to study the behaviour of MAC layer and how TCP traffic influences by the MAC priority principle.

Every node handles a set of independent ns3::DcfState, each of which represents a single DCF within a MAC stack. In addition, Each DcfState has a priority implicitly associated with it. For example, the priority is determined when the ns3::DcfState is added to the DcfManager: the first DcfState to be added gets the highest priority, the second, the second-highest priority, which is used to handle "internal" collisions. When two local DcfState are expected to get access to the medium at the same time, the highest priority local DcfState wins access to the medium, and the other DcfState suffers an "internal" collision. Moreover, nodes are assigned different Contention Window (CW), each node assigns maximum and minimum values for its own CW through DCA. DCA class implements the packet fragmentation and retransmission policy. It uses the ns3::MacLow and ns3::DcfManager helper classes to respectively send packets and decide when to send them. Packets are stored in ns3::WifiMacQueue until relay nodes can send them in the topology.

TCP protocol is configured by using ns3::TcpL4Protocal class in ns3. It started with specifying the TCP socket buffer in both sender and receiver. It has the capacity of 524288000 bytes. The Slow Start Threshold is set to the default value, and the initial Cwnd is set to 3. Moreover, the TCP socket type is determined by choosing from different varieties from Tahoe, new Reno, and Westwood for every TCP test. Moreover, TCP sender is assigned different ports for each flow such as 80 or 81, later assigned destination address and activating TCP socket factory in TCP sender side by creating a socket on the sender for application. TCP Congestion Window is tracked all time using OutputStreamWrapper class and pointer in ns3, and all changes in Cwnd is print in files for further analysis.

For every simulation run, the seeds are configured by using changing global seed and re-run the simulation. These values are stored in two $(ns3 :: GlobalValue)$ instances: $g\_rngSeed$ and g_rngRun. We used the class $(ns3 :: RngSeedManager)$ provides an API to control the seeding and run number behavior. This seeding and substream state setting must be called before any random variables are created; e.g:

- $RngSeedManager :: SetSeed(30)$; // Changes seed from default of 1 to 30.
- $RngSeedManager :: SetRun(400)$; // Changes run number from default of 1 to 400.

In this simulation, we tested Tahoe, New Reno, Westwood and MACINT-TCP in a saturated environment with an increased number of nodes and TCP flows. Traffic is generated at constant rates using NS3 features of TCP socket factory, with at least two bidirectional flows. The topology used consisted of multi-hop with two flows in reverse direction, figure 1.3 in chapter 1 depicts the used topology.

The simulation is based on NS3, COPE [12] is Xor coding network implementation, and chain topology is tested using a multiple hops scenario. Since the size of the TCP's queue at receiver and sender is limited to 32,768 bytes, the TCP layer queue does not have a maximum delay limit. The queue follows the queue discipline of FIFO. Furthermore, the MAC layer is set to the default setting, for example, the maximum packet number is set to be 30 packets with a maximum delay of 0.5 seconds. Other settings are summarised in table 5.1.

The proposed algorithm is tested with a different number of hops. Moreover, the total average throughput is recorded, and average delay and jitters are calculated for the whole simulation time. Data is presented in prospective graphs. RLP is also calculated and presented to determinate the reliability of TCP flows when network coding is being used in the wireless network.

The simulation results were collected using ns3 flow statistic tool. The above settings have been applied to all scenarios. The simulation results were recorded in a spreadsheet, and the data has been plotted in graphs using QiPlot and MATLAB. Thus all information is gathered and presented in figures. for example, each The graphs show throughput, packet loss ratio, delay and jitters values, and packets numbers (received and transmitted).

5.5 Results and Discussion

5.5.1 Results

In multi-hop networks, throughput of Tahoe, New Reno and Westwood are identical in some degree. It is obvious that the performance of New Reno and Westwood variants are homogeneous as the number of hop increases. Most of TCPs box-plot are narrow and small, as a result of marginal differences of values, and most throughput values are close in values . The fluctuations in throughput are very small. And that is reflected on box-plot in shown figure 5.4. Furthermore, Westwood and Westwood+ did not perform as expected in [129], as it was suggested that Westwood would outperform TCP New Reno. It was because of Xor network coding, which has affected the overall performance as both flows of TCP participants are not aware of Xor network coding taking place.

In general, the maximum throughput value recorded in the proposed MACINT-TCP and New Reno using our MAC priority is 6 Mbps. Meanwhile, this value with other TCPs were around 5 Mbps. Hence, at least 1 Mbps throughput difference was accounted for. In above, the sentence "the New Reno using the proposed MAC priority" refers to New Reno using equation 5.7. The calculation in equation 5.7 replaces the conventional Contention Window calculation at MAC layer. The CW is adjusted based on the priority and position of the intermediate nodes at the network. The minimum throughput that is obtained using this calculation was around 5.003 Mbps.

On other hand, RLP of Tahoe and New Reno was higher than other. And Figure 5.5 depicted such higher RLP values. Box-plots of Tahoe, New Reno and Westwood have some similarity, their box-plots are more wider and at 6th hop started to skewed upwards where RLP is more than 5%. Tahoe showed some abnormality in the last hop as well as New Reno and Westwood. The maximum value was recorded for Tahoe at 15.3% . Meanwhile Westwood had a maximum value of 9 % of RLP. The proposed MACINT-TCP has the lowest RLP ratio, and that was because of its DCF function updating the failed CW. It can guarantee no lost packets due to congestion. The strategy of MACINT-TCP is simple. It is cored on the principle of minimum waiting time to access the medium, as MAC priority itself cannot guarantee a zero RLP. The only abnormality seen in MACINT-TCP box-plot figure in 5.5 was at the 12th hops, but it was only 2% of RLP at maximum. The same can be seen in New Reno using our MAC priority. The reason for better RLP in New Reno using our MAC priority was ability of New Reno to deal with Duplicated acknowledgment and packet loss by assigning the lowest Contention window in DCF function in MAC layer. Therefore, there was very few DupAck and retransmission incidents. Furthermore, RTT time did not exceeded the retransmission timer as a result, RLP was nearly zero.

In case of delays , MACINT-TCP had the highest value at maximum of 5.6 seconds at 10th hop. The MACINT-TCP has doubled the delay up to 19%-35%, when we comapre it with other TCPs (Tahoe, New Reno, Westwood). The delay of MACINT-TCP grew up steadily until 10th hop before it drops to 3.5 seconds at 12th hop because of more incident of retransmission and DupAcks. Again, the same can be said in case of New Reno using our MAC priority except it drops slightly with only 0.5% of RLP, which is much better result than our proposed MACINT-TCP. On other hand, Tahoe , New Reno and Westwood had less delay value than the proposed MACINT-TCP, they did not exceed average delay of 2 seconds. Furthermore, Tahoe , New Reno and Westwood box-plots increased gradually until 6th hop, before they dropped back to around 2 seconds of average delay.

In the case of jitters, Tahoe, New Reno and Westwood had reached a maximum of 0.25 second, with increased jitters from second hop to 10th hop. The fluctuations or skewing of values of their box-plots were not an issue, they were dense and small with no abnormalities. In the case of MACINT-TCP and New Reno using our MAC priority, jitters were much smaller than Tahoe, New Reno and Westwood, with a maximum of 0.2 seconds. Also, no abnormalities were found in MACINT-TCP or New Reno used MAC priority, box-plots were very dense and gradually increased as hop increased until 12th hop. While the jitters of MACINT-TCP was improved up to 29%. For example, jitters were reduced down to 29% when it compares with New Reno, and 20% against Tahoe, and 13% in case of Westwood. The improvement in jitters was because of the minimum value in the relay nodes in the wireless topology. For example, the relay nodes have been assigned values for CW at MAC layer between 2 to 3 at the initial stage, while edge node were assigned default values 15. At simulation running time, relay nodes will obtain the value for failed CW between 14 and 17. The most counted values were 14 and 15 – a little over 200,000

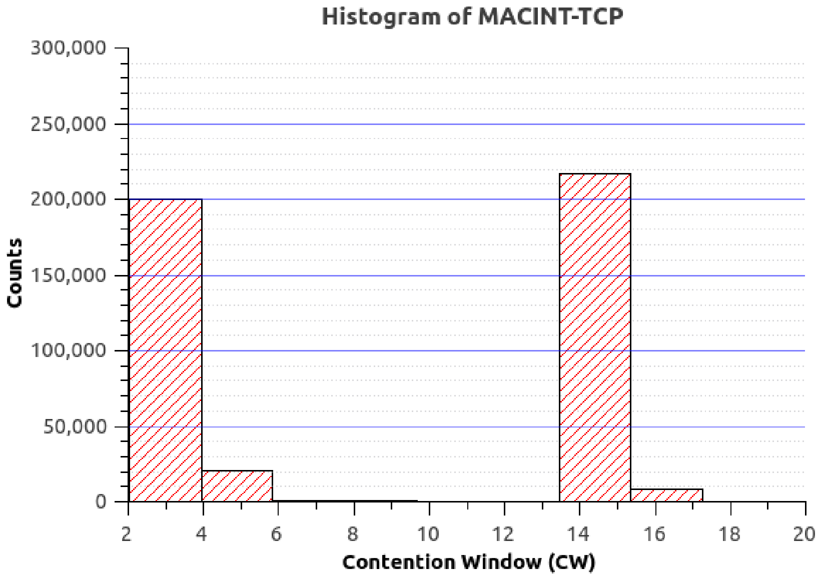


Figure 5.3: Histogram of MACINT-TCP.

counts, as shown in the histogram of MACINT-TCP in figure 5.3. The edge nodes have shown values between 2 and 6, and the most counted values were between 2 and 4. Thus, it explains why Jitters of TCP senders were much better; the heavy load of waiting was transferred to relay nodes. Besides this was the reason for the prolonged delay at All TCP variations including MACINT-TCP. It seems that MAC propriety techniques have trad-off. It can give opportunities to relay nodes, and it also enhanced jitters and RLP. Nevertheless, it causes more damage in term of delays and throughput regarding TCP. Table 5.2 summaries all MACINT-TCP improvements in percentage in terms of throughput, RLP, delays and jitters. Some values were replaced with (-), which indicates no improvement in the perspective columns.

5.5.2 Discussion

The impact of MAC is evident. The role of middle nodes as a means to perform Xor coding on matching pairs can affect TCP's end-to-end delay. However, there are few chances to find the matching pair, and that adds an extra delay in the case of MAC priority. However, it is not the main culprit, the delay occurs due to increased access to the medium by the centred node which holds edge nodes which generate bidirectional TCP traffic. It can be explained that the current TCP behaviour does not favour Xor network coding. It is harder for the intermediate node to accommodate coding process, and MAC priority is about giving more access and transmission opportunities to intermediate nodes. It does not help to increase throughput as the TCP mechanism is suspicious towards lost or damaged packets. Therefore, MAC priority can help intermediate nodes to gain access, but the TCP sender is unable to inject more packets as it sees the link unreliable or congested.

What makes MACINT-TCP superior to other TCP versions is that MACINT-TCP has a different calculation for slow-start, congestion-avoidance and DupAck. Figure 5.2 depicts a detailed calculation of MACINT-TCP components which shows better outcomes against other TCP variants (New Reno, Westwood and Westwood+). To conclude, without MAC priority, New Reno and Westwood+ suffer timeouts and excessive increases in back-off values at the MAC layer. As a result, there is degradation of the overall throughput. In addition, congestion avoidance affects the network coding process by restricting the number of available packets – specifically native packets – if there are a few native packets that can be coded in an intermediate node. Thus, the ability of network coding to interact with the network is crippled. For example, in TCP New Reno, slow-start and congestion-avoidance and DupAck are the most called functions, and congestion-avoidance is counted to be holding 56% of TCP New Reno's overall operations in NS3 simulation. Slow-start and DupAck were around 20% and 22%, respectively. It is evident that the issue with TCP components such as slow-start, congestion-avoidance and DupAck cause real challenges to throughput in Xor network coding. There is a need to redesign these components to fit the network coding process. MACINT-TCP spent most of its time in slow-start at all time, very few incident less than 0.25% in congestion avoidance state.

MACINT-TCP has a different calculation to ensure the state of congestion avoidance is not being called or trigged intensively. Figure 5.8 shows the slow-start count function, which it records every time slow-start has been called in our simulation. In Figure 5.8, slow-start has been called more than 2,500 times over the simulation time of 80 seconds. However congestion avoidance in Figure 5.9 has been called only 25 times over the duration of the simulation. If we compare the congestion window with RTT in Figure 5.10, MACINT-TCP has distinct behaviour. Here, it has a lower RTT while the congestion window is reaches 65,000 bytes, which is the maximum value of the congestion window. Moreover, the T shape column was between 0.05 and 0.21 second of the RTT time. Comparing the MACINT-TCP in 5.10 with the New Reno in Figure 5.14, the TCP New Reno using the normal MAC-layer contention window has a higher RTT value over a different number of congestion window values.

Most values of Cwnd are arranged between 15,000 and 60,000, while RTT is increased up to 0.45 RTT second. Similar but not identical is Figure 5.15, which uses the proposed contention window in the MAC layer and has a similar range of congestion window. However, the RTT time is lower: most of the congestion windows are between 0.01 and 0.25 seconds of RTT time. In the same Figure 5.15, it is clear that after 0.3 seconds of RTT there were very few congestion window values, which means there were not any sent segments exceed 0.3 seconds regarding RTT. Therefore, New Reno using MAC interaction has less RLP and much better average delay. Moreover, Figure 5.11 shows the Cwnd of MACINT-TCP, which has fluctuated as expected. MACINT-TCP has the same framework as TCP Tahoe, it is our first attempt to understand how we can influence MAC layer to cooperate with the transport layer in cases of TCP when Xor network coding is applied. In New Reno using MAC interaction, Figure 5.16 has depicted the Cwnd behaviour when it uses our MAC-layer contention window calculation. It has similar behaviour as the normal TCP New Reno. Furthermore, the value of Cwnd fluctuates as in MACINT-TCP. It is still similar to the current TCP New Reno. The value of Cwnd is much smaller than MACINT-TCP's Cwnd. However, the value of Cwnd in MACTINT-TCP is much higher, and is formed gradually before plunged due to packet loss or DupAck incidents. In New Reno when there is packet loss Cwnd declines and enters congestion avoidance, it gets in slow increase state, congestion-avoidance occurrences force the Cwnd to decrease for every incident of packets loss, it cannot exit the state of congestion avoidance. Unlike MACINT-TCP, it can enjoy the state of slow-start without affecting the throughput, RLP and causing delays. Figure 5.12 and Figure 5.13 present congestion windows versus the instantly achieved throughput per segment transmission. As shown in Figure 5.12, MACINT-TCP can achieve the same throughput as New Reno using MAC interaction: the difference between the two graphs is how the throughput spread over congestion window values. For example, in Figure 5.12 the highest throughput is achieved when Cwnd increases in case of MACINT-TCP. However, Figure 5.13 has distribution which is not gradual as in MACINT-TCP, it is more spread over values of Cwnd, although the highest throughput was at the maximum value of Cwnd (65,000 byte).

In the case of heterogeneous multi-hop paths, some hops use our priority MAC introduced in equation 5.7 and other nodes use different MAC layer technology. In this case, we gain access to the DCF function by actively altering the calculation of failed contention window, and we also use the QoS (Quality of Service) aspects. Within Qos, we initialised the starting values for every CW in our network. Moreover, we keep track of the state needed for each single DCF function. The DcfState in ns3 can be controlled as in the 802.11e-style relative QoS implementation. Furthermore, that will allow us to assign all setting for QoS parameters, besides it allows us to control the relative QoS differentiation. It can work well with any 802.11 protocols using QoS. Moreover, if there is any deployment of different MAC layer technologies in heterogeneous multi-hop, It can be applied to our work with a small modification. The interoperability of our MAC priority with other MAC layer technologies can be achieved through the use QoS parameters.

5.6 Summary

We have investigated the MAC layer impact on TCP flows in interaction with Xor network coding using COPE. MACINT-TCP showed 20% improvement in throughput. However, it gives up to 100% improvement in PLR. Delays and jitters of MACINT-TCP are at their lowest when compared to other TCP variants that apply the MAC priority principle. In this investigation, the core function of TCP congestion avoidance causes a drain in the overall throughput, and the ability to provide relief from congestion might not present. The current congestion window mechanism does not comprehend the role of middle nodes to perform coding. However, MAC-layer priority can eliminate any bottleneck issues at centred nodes. Moreover, MAC layer priority

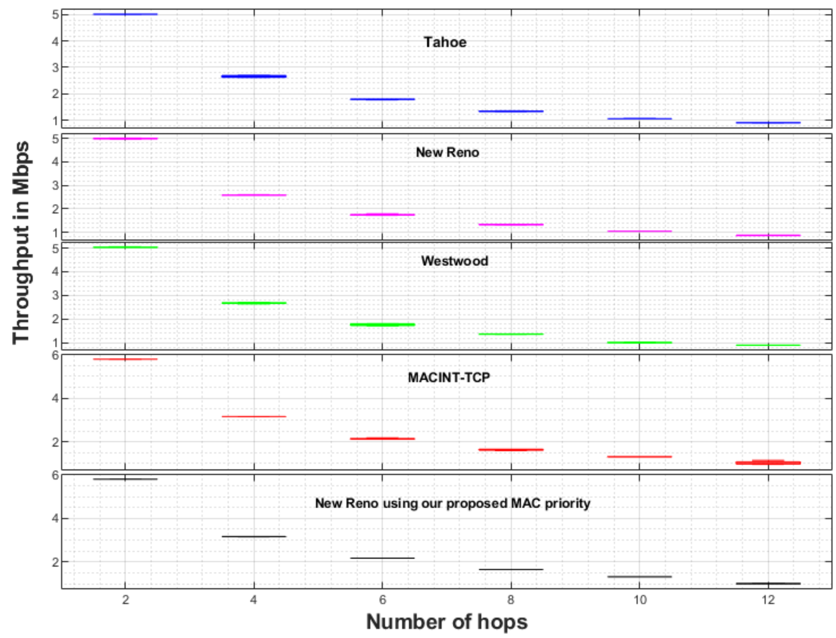


Figure 5.4: Increased number of hops vs. throughput of TCPs with MAC priority and MACINT-TCP.

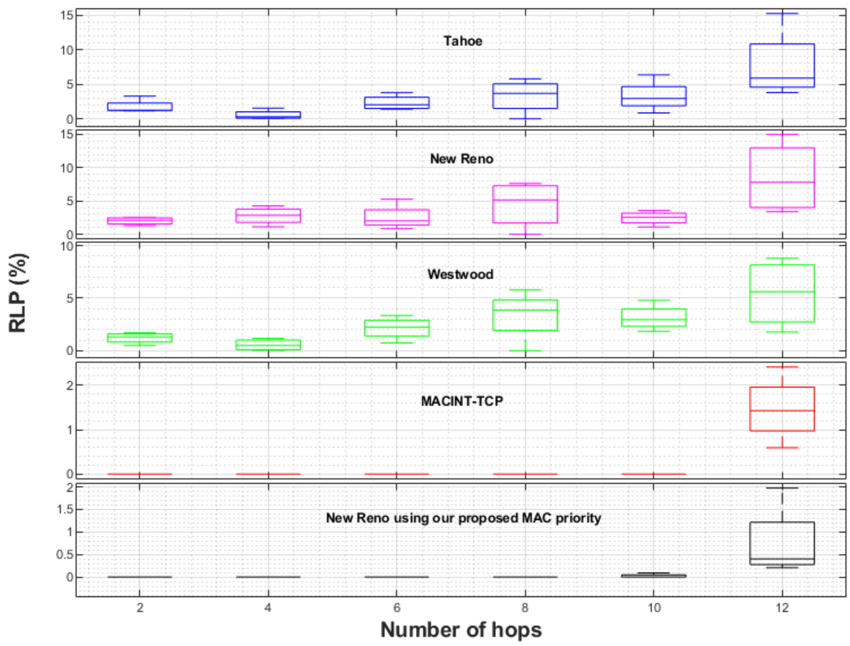


Figure 5.5: Number of hops vs. RLP TCPs with MAC priority and MACINT-TCP.

brings some benefits in term of RLP, but it can lead to longer delays and jitters. On the other hand, we proposed MACINT-TCP to improve the network coding nature functionality and incorporate better interaction between TCP and the MAC layer to obtain favourable outcomes in terms of

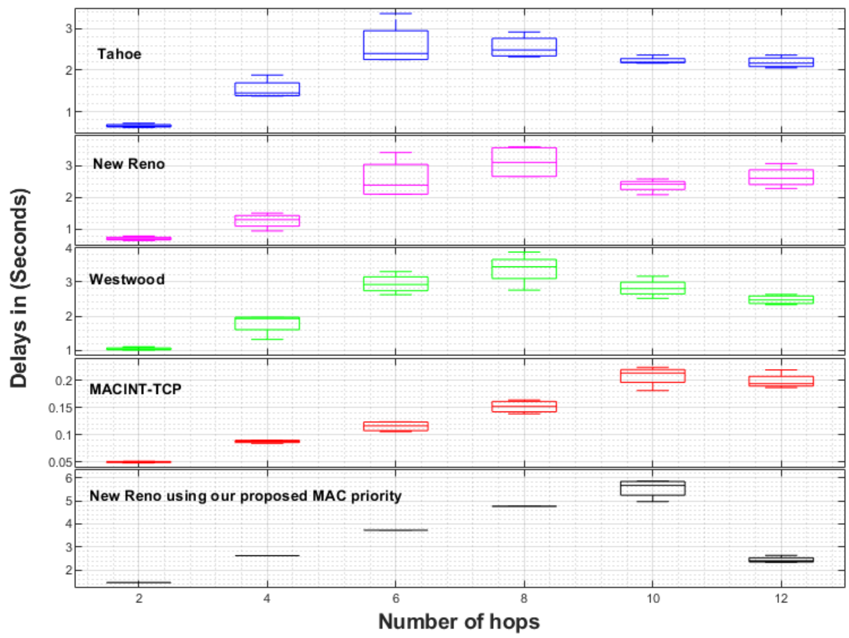


Figure 5.6: Number of hops v. delays of TCPs with MAC priority and MACINT-TCP.

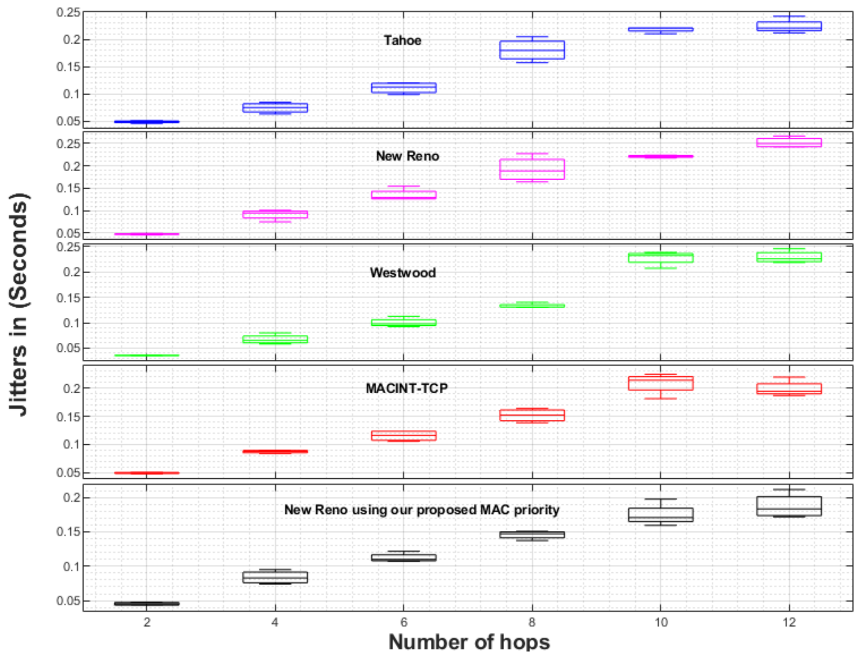


Figure 5.7: Number of hops vs. jitters of TCPs with MAC priority and MACINT-TCP.

throughput, RLP, delays and jitters. MACINT-TCP overtook all tested TCP variants using MAC priority or without the MAC-priority principle.

We can summarise our findings in points as follows:

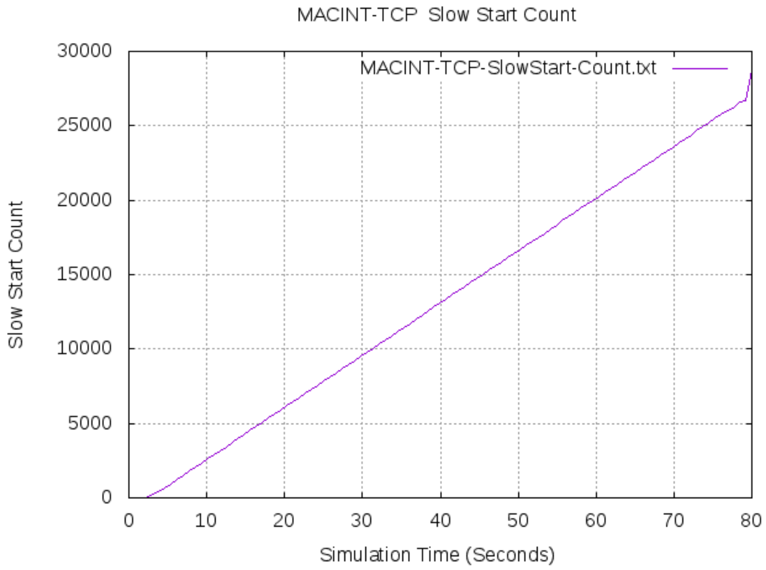


Figure 5.8: MACINT-TCP Slow Start Count.

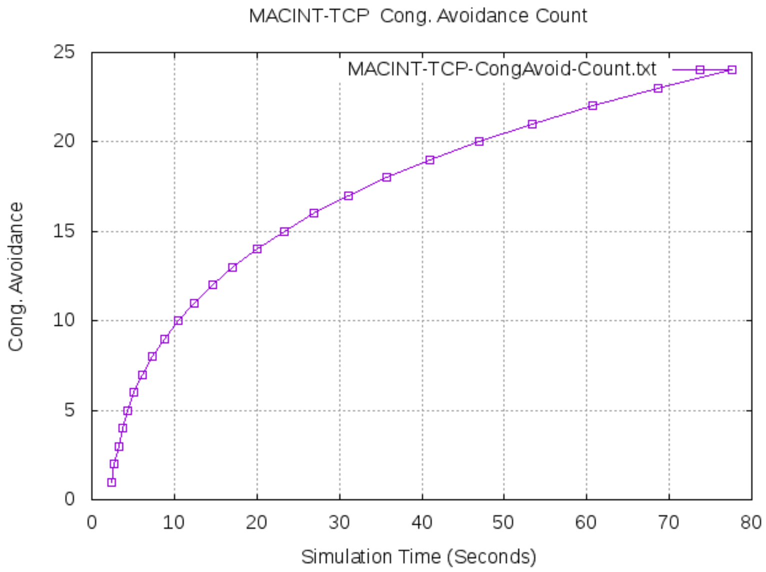


Figure 5.9: MACINT-TCP congestion avoidance count.

- The current congestion window in all TCP variants is not compatible with Xor network coding, and it affects throughput and RLP. Furthermore, delays and jitters are also affected and increase when the number of hops increases. The role of middle nodes as a means to perform Xor coding on the matching pair can affect TCP. Moreover, there are few chances to find the matching pair, and that adds an extra delay in cases of MAC priority. Delays occur frequently due to increased access to the medium by a centred node. It can be concluded

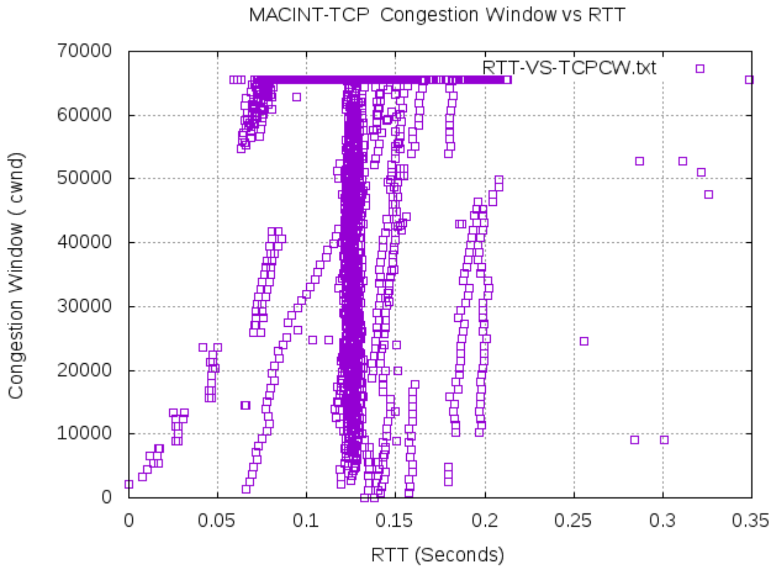


Figure 5.10: MACINT-TCP congestion window vs. RTT.

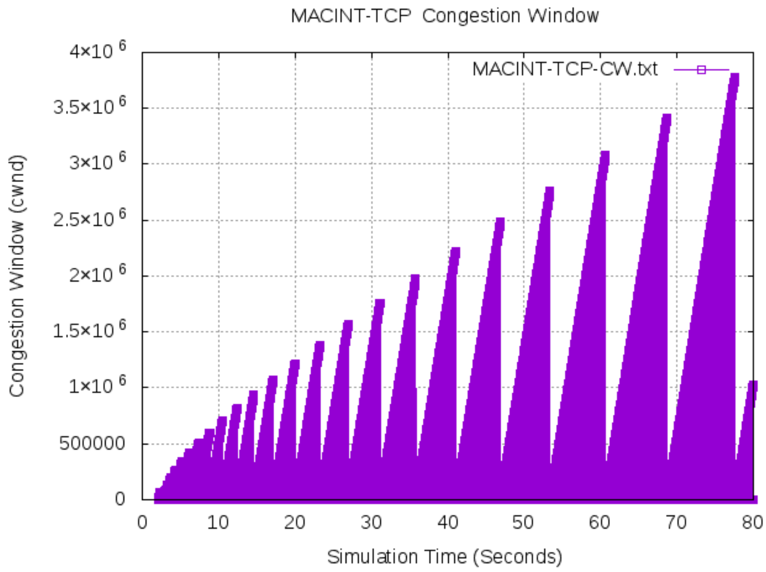


Figure 5.11: MACINT-TCP congestion window.

that the current TCP behaviour does not favour Xor network coding.

- The impact of MAC priority is negative, because it adds an extra delay. However, it is not the main issue, the delay occurs due to increased access to the medium by the intermediate node which holds edge nodes which generate bidirectional TCP traffic. And that can explain why MACINT-TCP and Reno using MAC priority have higher delay values.

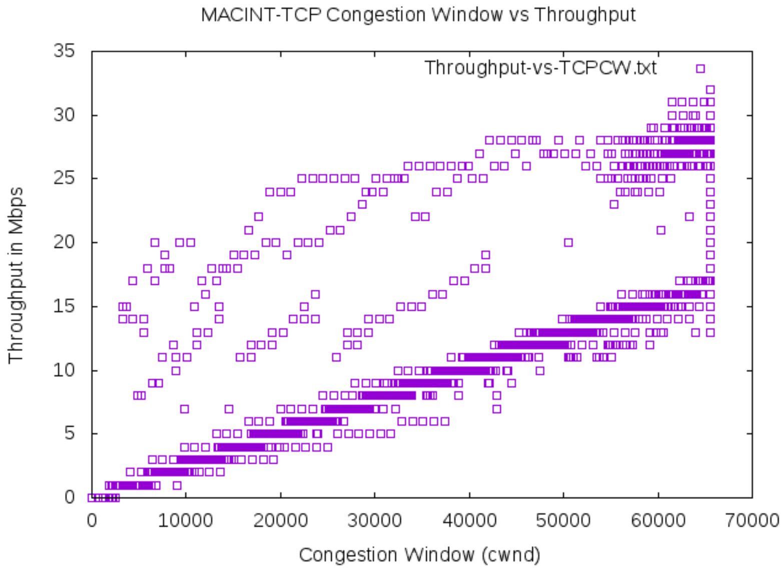


Figure 5.12: MACINT-TCP congestion window vs. throughput.

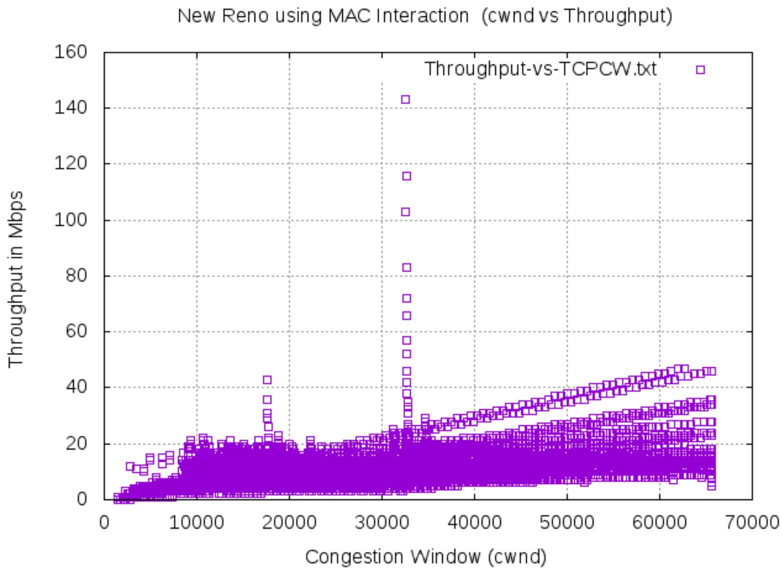


Figure 5.13: New Reno using MAC interaction (Cwnd vs. throughput).

- Without MAC priority, New Reno and Westwood+ suffer timeouts and excessive increases in back-off values at the MAC layer. As a result there is degradation of the overall throughput. In addition, congestion avoidance affects the network coding process by restricting the number of available packets – specifically native packets – if there are a few native packets that can be coded in an intermediate node. Thus, the ability of network coding to interact with the network is crippled.

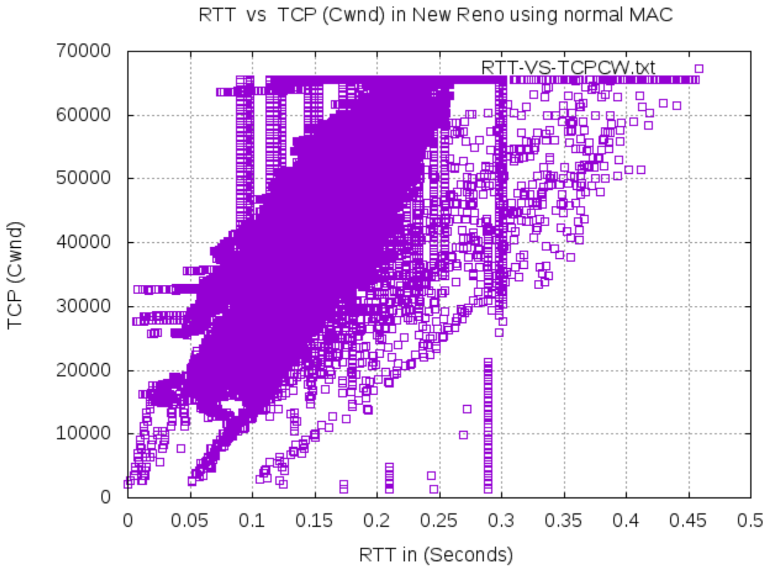


Figure 5.14: RTT vs. TCP Cwnd New Reno using normal MAC.

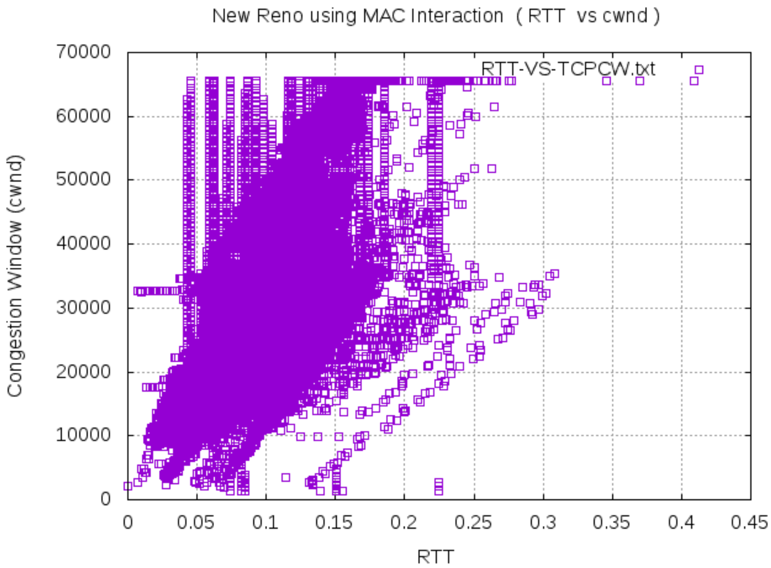


Figure 5.15: New Reno using MAC Interaction (RTT vs. Cwnd).

- It is evident that the issue with TCP components such as slow-start, congestion-avoidance and DupAck cause real challenges to throughput in Xor network coding. There is a need to redesign these components to fit the network coding process.

- Both Westwood and Westwood+ failed to outperform TCP New Reno as the authors in [129] claimed. TCP New Reno behaved better than both Westwood and Westwood+ in terms of

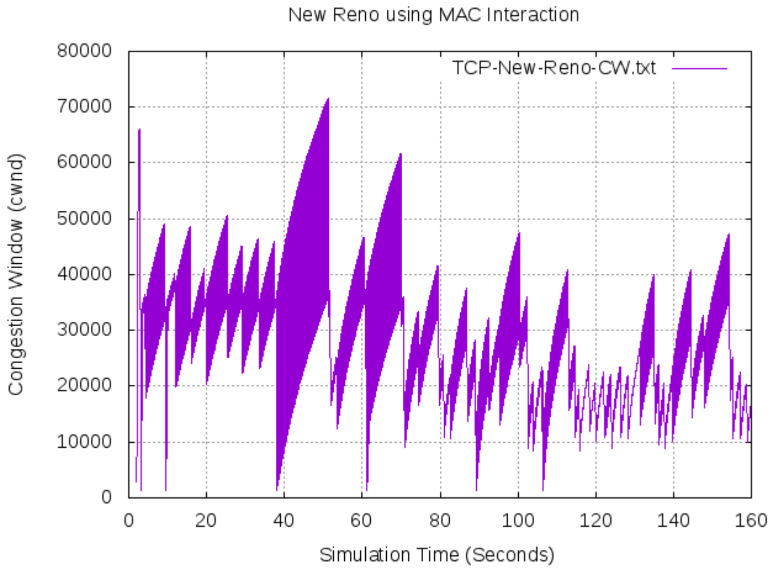


Figure 5.16: New Reno using MAC interaction.

throughput, RLP, delays and jitters.

- All TCP variants using MAC priority has improved the RLP due to the fact that delay variations could be regulated and consequently reduced as intermediates nodes have more access to medium when it has a packet to send. However, all TCP variants using MAC priority suffer increased long delays. Because MAC priority adds an extra delay, also the intermediate node holds back edge nodes from injecting more packets in the network.

- It is harder to understand the role of the intermediate node in accommodating the coding process, and MAC priority is about giving more access and transmission opportunities to middle nodes. It does not help to increase throughput, as the TCP mechanism is suspicious towards lost or damaged packets. Therefore, MAC priority can help middle nodes to gain access, but the TCP sender is unable to inject more packets as it sees the link as unreliable or congested.

- The proposed MACINT-TCP has the lowest RLP ratio, and that was because of its DCF function of updating failed CW. It can guarantee no lost packets due to congestion. The strategy of MACINT-TCP is simple. It is cored on the principle of minimum waiting time to access the medium, as MAC priority itself cannot guarantee a zero RLP.

- New Reno using MAC interaction using our proposed MAC layer contention window calculation, enjoys the benefits of frequent and smooth access to the medium which is proved to provide better throughput and a lower RLP.

- What contributes to the improvement of jitters in MACINT-TCP was because of providing the minimum value for relay nodes in the wireless topology. For example, relay nodes have been assigned values for CW at MAC layer between 2 to 3 at the initial stage, while edge node were assigned default values 15. At simulation running time, relay nodes will obtain the value of 15. Thus, it explains why Jitters of TCP senders were much better; the heavy load of waiting was transferred to relay nodes. Besides this was the reason for the prolonged delay at All TCP variations including MACINT-TCP.
- The reason for better RLP in New Reno using our MAC priority was ability of New Reno to deal with Duplicated acknowledgment and packet loss by assigning the lowest Contention window in DCF function in MAC layer. Therefore, there was very few DupAck and retransmission incidents. Furthermore, RTT time did not exceeded the retransmission timer as a result, RLP was nearly zero.
- It seems that MAC propriety technique has trad-off. It can give opportunities to relay nodes, and it also enhanced jitters and RLP. Nevertheless, it causes more damage in term of delays and throughput regarding TCP.

Improvement in				
Number of Hops	Throughput	RLP	Delay	Jitters
2	16	100	-	-
4	24	100	-	24
6	22	100	3	35
8	51	100	-	40
10	24	100	-	31
12	19	83	-	17

Table 5.2: Improvement summary over the number of Hops in (%), (-) refers to no improvement.

Chapter 6

Conclusions and Future Work

This thesis was aimed at achieving improved throughput and reducing latency in networks. It has investigated the MAC, Transport layers, and proposed Cross-layers for effective and practical using Xor Network coding on multiple hop networks. Networks. This chapter presents a summary of the study, findings and conclusions of this thesis.

An investigation of the role of relay node at the MAC layer was carried out in this Chapter 3. This has allowed to understand the impact of MAC layer on the Xor network coding process.The following are the major findings in Chapter 3 with respect to COPE:

- It provides very low throughput.
- It does not reach the maximum utilisation or throughput.
- It does not use the medium efficiently.

The reason behind low throughput in COPE was because of the use of conventional contention window calculation in the legacy of 802.11. Also, the protocol of 802.11 was suited to be fair to all stations in the network, wherein the case of COPE with MAC priority is about providing more access to relay nodes rather than fairness. We also learnt that COPE with MAC priority could reach a reasonable level of throughput, but the number of coded packets is low. These packets contribute slightly to the overall throughput. COPE with MAC-layer priority results in few or no coded packets due to the realisation of no matching pair to be coded together at a time. Also, COPE using current MAC and COPE with MAC-layer priority techniques inherited overheads that exist in state-of-the-art MAC 802.11. As a result, they achieved low throughput. Moreover, the random back-off algorithm in DCF increases the overhead and provides low throughput. Unconventionally, when the CW's calculation comes into play and doubles the CW's value of the relay node, it creates a real challenge to network coding relay nodes to gain access to the medium.

Additionally, the contention between nodes increases the probability of collisions and decreases the available throughput.

The simulation results show that the proposed solution guarantees the least waiting time for stations, in a more controlled manner. For example, the proposed solution allows the relay nodes to transmit the coded packets, which is crucial to boost the throughput in the network. The two edge nodes were also found to enjoy the benefits of no doubling times when they had failed to reach the medium, and that is because of the proposed scheme allocates DCF channel in a better manner, favouring relay nodes that perform coding by using the best CW value for relay node without constraint from other nodes or edge nodes in a wireless network. It maximises the chances for more throughput in two and more hops. As a result, the proposed contention window scheme at MAC layer has improved the throughput when compared with other existing schemes, and thus throughput is ranged from 24% - 97% over 11 hops. Again, the RLP of the proposed scheme was the lowest when put against COPE using normal MAC and COPE using MAC-layer priority. However, it is significant until the fifth hop. For example, a second hop RLP improved 61% and continues to improve less than 10% until fifth hops. Such improvement in RLP was because of allocating best CW based on our calculation in a manner suits only relay nodes between two edges. On the contrary, COPE using normal MAC did not enjoy access to the medium frequently when it has a packet in the queue. Moreover, the RLP was high because of the significant CW value in the traditional 802.11 DCF. We also learnt that Jitters and delays have proved to be the lowest against COPE using normal MAC layer and COPE with MAC- layer priority. It saw 96% improvements in terms of jitters. The proposed model has guaranteed the relay nodes a least possible waiting time and quick access to the medium, hence the jitters has improved. Contrary to the 802.11 standards, it helped the relay nodes to empty their coded packet queue without imposing medium access fairness. Furthermore, the overall delay improvement in the proposed scheme was around 7% against MAC-layer priority, wherein cases of standard MAC there is not any improvement. In general, MAC priority technique adds a bit of delay at relay nodes, and as mentioned in chapter 3, the CW of relay nodes reached the maximum value of CW 15 after some times. Meanwhile, the edge nodes got the least CW of 3, At the beginning of simulation the relay nodes had the value of 3, but this has changed to 15 during simulation time. This shows the delay issue related to the MAC priority technique and our scheme. The above process increases the the end-to-end delays for edge nodes, and this is the drawback of MAC priority technique.

The impact of MAC priority on non-prioritised flows is limited; and without MAC priority, the flow does not have any support which is helpful to gain frequent medium access. Because of that, the throughput of non-MAC priority is low. It has the same throughput in case of COPE using normal MAC. On the other hand, the non-prioritised flows suffer prolonged jitters and delays. This is due to the lack of flow assistance in the MAC layer. For example, MAC priority has the small value of jitters and delays due to the assignment of a small value of CW in the intermediate node, so it can assist the flow packets to pass down the medium. Furthermore, COPE using normal MAC has a similar attitude; it has no MAC priority feature. Therefore, it has higher delays and jitters values. Lastly, in the case of non-priority flows, the fair index was fair when the offered load is low. MAC-priority technique is only fair to nodes which are applying the principle, and non-priority flows are subject to struggle for gaining the medium. In fact, that is very reasonable as MAC-priority technique is about giving more access to designated nodes rather than competing fairly.

In Chapter 4, an investigation on how to improve the throughput and reliability of TCP protocol in Multi-Hop scenarios via Xor network coding has been carried out. The lack of transparency between the transport layer and the implemented Xor network coding has affected the throughput, RLP and delay. The studied TCP variants (Tahoe, New Reno and Westwood) were severely affected by Xor network coding as they are not aware of the role of relays nodes in the network. However, the network coding could improve the TCP throughput if it was able to tolerate the core process of network coding. In this case, the role of the relay nodes is to encode the TCP packets to reduce the number of transmissions and use the available resources for injecting more packets in the medium. Consequently, this will improve the throughput. It is crucial to notice that although all TCP variants have higher coding rates and opportunities to find a matching pair, this does not improve the throughput or mask any packet loss in term of RLP. What matters is that right packet at the right time at right place. Furthermore, the congestion avoidance at every TCP variant was the main culprit that worsen both throughput and RLP. For instance, the congestion avoidance was the most called function in simulation time, Tahoe (88%), New Reno (77%) and Westwood (62%). The second issue of TCP when Xor network coding was applied, is RLP deterioration. High RLP ratio harms throughput and the smoothing of TCP flows, as it was as a result of retransmission of some segments due to the duplicated ACK, or timeout. In New Reno duplicated ACKs accounted for 16% Westwood was more affected by duplicate ACKs, which were calculated to be 30%.

WeNC-TCP has slightly better throughput and zero RLP. Also, it has the lowest delay and slightly low jitter variance compared with other TCP variants. WeNC-TCP has eased the effect of congesting avoidance, and hence has a less restrictive impact on TCP flows. As a result, it has much better RLP and delay effects. For instance, the RLP was zero, and end-to-end delays have reduced to 80%. However, the throughput improvement using WeNC-TCP was the insignificant. On the other hand, the slow-start of WeNC-TCP has enjoyed a massive role: it has been beneficial in term of throughput, delay and RLP, unlike other slow-starts in TCP variants (Tahoe, New Reno and Westwood) where it has played only a small role in improving overall throughput, delay and RLP. The slow-start was the second most called function after congestion avoidance; for instance, Tahoe called the slow-start function up to 10%. In New Reno, slow-start was only 2%. However, Westwood's slow start was 30%, which is considered the highest of all TCP variants. In WeNC-TCP's slow-start congestion avoidance were recorded to be around 50%. The best of

WeNC-TCP can be seen in case of the ratio of lost packets(RLP), it has the best reliability of zero lost packets. The zero RLP ratio was because of no DupAck or Retransmission incidents were recorded. This explains the reason why the RLP ratio is close to zero. The threshold of 500 to 2000 bytes in TCP WeNC-TCP contributes to reducing the DupACK incidents, and this delays entering congestion avoidance state in WeNC-TCP. This controls the increase of Cwnd before reaching a critical congestion point. Such behaviour will help to minimise the loss packets ratio. Finally, WeNC-TCP performance excellency was because of good adaption of better engagement of Xor network coding and TCP sending rate.

Furthermore, reducing the impact of congestion avoidance and emphasising more on slows-start had given WeNC-TCP superiority over other studied TCP versions. With such outcomes, WeNC-TCP could be implemented for applications where the need for low and short delay variation can be required on the internet or a network of multi-hop nature with a prolonged path. Also, the reliability of WeNC-TCP was proved to be near zero loss, and it could be applied to a network where reliability is a crucial factor. Importantly, Cwnd of WeNC-TCP did not reach the maximum limit of 65000 bytes, which means throughput could be improved further by injecting more packets into the network. However, due to the misinterpretation of the medium conditions at lower layers, the probability of using the available resources are remote possibilities. Because of the principle of piggybacking makes it challenging to reach high throughput.

In Chapter 5, the MAC layer and the interaction between TCP and MAC are studied in terms of the overall performance of multi-hop wireless coded networks. The MAC priority impacts negatively, as it adds an extra delay. However, this is not the main issue, as the delay occurs due to increased access to the medium by the intermediate node. The intermediate node holds back edge nodes which generate bidirectional TCP traffic. Moreover, that can explain why the proposed MACINT-TCP and Reno using MAC priority technique have higher delay values. Although all TCP variants using MAC priority technique have improved the RLP, delay variations could be regulated and consequently reduced as intermediates nodes have more access to the medium. However, all TCP variants using MAC priority suffer the same increased prolonged delays. Because MAC priority adds an extra delay, also the intermediate node holds back edge nodes from injecting more packets in the network. It is a contradicting role: the intermediate node accommodates the coding process, and MAC priority is about giving more access and transmission opportunities to middle nodes when there is a coded packet. Such situation does not help to increase throughput, as the TCP mechanism is suspicious towards lost or damaged packets. Besides, MAC priority adds extra delays which doomed TCP packets to be lost or reached timeout. Therefore, MAC priority can help middle nodes to gain access, but the TCP sender is unable to inject more packets as it sees the link as unreliable or congested. Therefore, this contradiction resulted in low throughput and RLP with increased delays. But the jitters were much enhanced because of MAC priority extra time at relay nodes and less waiting time to inject more packets at the sender side. On the other hand, Without MAC priority technique, New Reno and Westwood+

suffer timeouts and excessive increases in back-off values at the MAC layer. As a result, there is a degradation of the overall throughput. Besides, congestion avoidance affects the network coding process by restricting the number of available packets – specifically native packets – if there are a few native packets that can be coded in an intermediate node. Thus, the ability of network coding to interact with the network is crippled. It is evident that the issue with TCP components such as slow-start, congestion-avoidance and DupAck cause real challenges to throughput in Xor network coding. There is a need to redesign these components to fit the network coding process.

The proposed MACINT-TCP has the lowest RLP ratio, and that was because of its DCF function of updating failed CW. It can guarantee no lost packets due to congestion. The strategy of MACINT-TCP is simple. It is cored on the principle of minimum waiting time to access the medium, as MAC priority itself cannot guarantee a zero RLP. While the reason for better RLP in New Reno using our MAC priority was the ability to deal with Duplicated acknowledgement and packet loss by assigning the lowest Contention window in DCF function in MAC layer. Therefore, there were very few DupAcks and retransmission incidents. Furthermore, RTT time did not exceed the retransmission timer. As a result, RLP was nearly zero. Finally, the extra regulated time provided at the relay nodes in the wireless topology has contributed to the improvement of jitters in MACINT-TCP. For example, relay nodes have been assigned values for CW at MAC layer either 2 or 3 at the initial stage, while edge node was assigned default values 15. While simulation running time, relay nodes will obtain the value of 15. Thus, it explains why Jitters of TCP senders were much better; the heavy load of waiting was transferred to relay nodes. Besides this was the reason for the prolonged delay at all TCP variations including MACINT-TCP. It seems that MAC propriety techniques have trad-off. It can give opportunities to relay nodes, and it also enhanced jitters and RLP. Nevertheless, in TCP, this can cause more damage in term of delays and throughput.

6.1 Future Work

6.1.1 Mobility

Mobility is a pressing issue in ad-hoc wireless networks. COPE in [12] overlooked mobility feature. If COPE was able to provide basic mobility functionality, it could make COPE in [12] the prominent protocol for unicast transmission. Mobility in COPE could be achieved using MAC-layer cooperation, this could be taken as possible future work. TCP flows suffer when nodes are in motion. TCP needs a new role to accommodate the Xor network coding process while nodes are in motion. If the future work could also investigate if the MAC layer could help nodes to ensure better delivery and provide more access to the medium when the nodes have encoded packets. This could be done by assigning different CW according to the state and position of moving node. Alternatively, a new routing algorithm could be developed to ensure link availability to support network coding technique in case of mobility. The mobility of nodes can affect the encoding of incoming packets, and can lead to the re-transmission of encoded packets, as the author suggested in [141] and [46]. These challenges are still open questions; the need to accommodates such necessity is crucial. We could take this idea further to be applied in VANETs and 5G.

6.1.2 Improving WeNC-TCP

WeNC-TCP has not been investigated to bring the MAC layer closer; such close cooperation could bring some benefits to WeNC-TCP. It is evident that MACINT-TCP could be salient and beneficial as it acknowledges the MAC layer role in Multi-Hop wireless networks. MACINT-TCP boosted TCP reliability considerably and used the advantage of prioritised access to the medium to enhance jitters. Such improvements can be desirable to WeNC-TCP. WeNC-TCP could relay on some MAC-layer features such as fast acknowledgment and re-transmission of lost frames. These features could help to improve throughput and reliability. Throughput is highly desirable property that needs to be considered, but piggybacking makes it challenging to improve. There remains a need to find a new way to address the reliability in TCP without restricting throughput. Using the 802.11ax standard will be a new gateway to test and address this issue of TCP throughput. Moreover, 802.11ax is believed to be more efficient 802.11 traffic management, and to increase the average throughput four times per user in high-density WLAN environments [142]. The future work could evaluate the efficiency of WeNC-TCP using 802.11ax, and see if the ultimate performance could be reached. Such mutual benefits also can put WeNC-TCP to operate on more future platforms such as IoT (Internet of things) and 5G. The future 5G is characterised by the unprecedented necessity for high rate, ubiquitous availability, ultra-low latency, and high-reliability [143]. Therefore, WeNC-TCP could be taken a step forward into a brand-new wireless platform for future use. WeNC-TCP has high reliability and small variation regarding jitters.

BIBLIOGRAPHY

[1] Z. Fang, M. Medard, M. Hundeboll, J. Ledet-Pedersen, S. A. Rein, and F. H. P. Fitzek, "Comparison of analytical and measured performance results on network coding in ieee 802.11 ad-hoc networks," in *Network Coding (NetCod), 2012 International Symposium on*, pp. 43–48.

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

[3] A. Sprintson, "Network coding and its applications in communication networks," in *Algorithms for Next Generation Networks*, pp. 343–372, Springer, 2010.

[4] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory, 2003. Proceedings.*, pp. 442–, June 2003.

[5] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proceedings of the annual Allerton conference on communication control and computing*, vol. 41, pp. 40–49, The University; 1998, 2003.

[6] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pp. 143–, June 2004.

[7] M. Medard, F. H. P. Fitzek, M. J. Montpetit, and C. Rosenberg, "Network coding mythbusting: why it is not about butterflies anymore," *IEEE Communications Magazine*, vol. 52, pp. 177–183, July 2014.

[8] R. Koetter, M. M, 233, and dard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.

[9] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 2608–2623, June 2006.

[10] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, Oct 2006.

[11] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, pp. 2745–2759, Aug 2005.

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

[13] Z. Li and B. Li, "Network coding : The case of multiple unicast sessions," 2004.

[14] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1956–1970, 2011.

[15] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, Feb 2003.

[16] S. Omiwade, R. Zheng, and C. Hua, "Practical localized network coding in wireless mesh networks," in *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 332–340, June 2008.

[17] J. Le, J. C. S. Lui, and D. M. Chiu, "Dcar: Distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 596–608, April 2010.

[18] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *2006 2nd IEEE Workshop on Wireless Mesh Networks*, pp. 157–159, Sept 2006.

[19] Y. Benfattoum, S. Martin, and K. A. Agha, "Irocx: Interference-aware routing with opportunistically coded exchanges in wireless mesh networks," in *2011 IEEE Wireless Communications and Networking Conference*, pp. 1113–1118, March 2011.

[20] Y. Benfattoum, S. Martin, I. Gawedzki, and K. A. agha, "I 2 swp : Routing algorithm with intra-flow interference consideration in ad hoc network," 2010.

[21] H. Seferoglu, A. Markopoulou, and M. Medard, "Ncapq: Network coding-aware priority queueing for udp flows over cope," in *2011 International Symposium on Networking Coding*, pp. 1–8, July 2011.

[22] H. Seferoglu and A. Markopoulou, "Network coding-aware queue management for unicast flows over coded wireless networks," in *2010 IEEE International Symposium on Network Coding (NetCod)*, pp. 1–6, June 2010.

[23] J. Cloud, L. Zeger, and M. Medard, "Effects of mac approaches on non-monotonic saturation with cope - a simple case study," in *2011 - MILCOM 2011 Military Communications Conference*, pp. 747–753, Nov 2011.

[24] F. Zhao and M. Medard, "On analyzing and improving cope performance," in *2010 Information Theory and Applications Workshop (ITA)*, pp. 1–6, Jan 2010.

[25] T. Nage, F. R. Yu, and M. St-Hilaire, "Adaptive control of packet overhead in xor network coding," in *2010 IEEE International Conference on Communications*, pp. 1–5, May 2010.

[26] Z. Ning, D. Zhang, and X. Liu, "A novel approach of network coding in wireless networks overhearing," *Chinese Journal of Electronics*, vol. 27, no. 6, pp. 1297–1304, 2018.

[27] E. Pertovt, K. Alic, A. Svigelj, and M. Mohorcic, "Cancar - congestion-avoidance network coding-aware routing for wireless mesh networks," *KSII Transactions on Internet and Information Systems*, vol. 12, pp. 4205–4227, 09 2018.

[28] A. Argyriou, "Wireless network coding with improved opportunistic listening," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 2014–2023, April 2009.

[29] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, and S. Gheorghiu, "Multipath code casting for wireless mesh networks," in *Proceedings of the 2007 ACM CoNEXT Conference*, CoNEXT '07, (New York, NY, USA), pp. 10:1–10:12, ACM, 2007.

[30] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 622–634, June 2009.

[31] J. Cha, J. Kim, and J. Kim, "Novel joint network coding and scheduling scheme in distributed tdma-based wmns," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pp. 1739–1743, Nov 2013.

[32] D. T. C. Wong, S. Zheng, and C. K. Tham, "Performance analysis of a multi-rate ieee 802.11 mac with network coding," in *2010 IEEE International Conference on Communication Systems*, pp. 512–516, Nov 2010.

[33] D. Umehara, C. H. Huang, S. Denno, M. Morikura, and T. Sugiyama, "Enhancement of ieee 802.11 and network coding for single-relay multi-user wireless networks," in *2010 4th International Conference on Signal Processing and Communication Systems*, pp. 1–9, Dec 2010.

[34] A. Paramanathan, P. Pahlevani, D. E. Lucani, and F. H. P. Fitzek, “On the need of novel medium access control schemes for network coding enabled wireless mesh networks,” in *2013 IEEE International Conference on Communications Workshops (ICC)*, pp. 306–311, June 2013.

[35] X. Yang, W. Wang, and J. Wang, “A cross-layer optimization mechanism for inter-session network coding in wireless mesh network,” in *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 264–271, Nov 2013.

[36] F. Farhadi and F. Ashtiani, “A network coding-based packet forwarding scheme for unicast random access networks with exponential backoff,” in *2013 Iran Workshop on Communication and Information Theory*, pp. 1–6, May 2013.

[37] M. B. Iraji, M. H. Amerimehr, and F. Ashtiani, “A queueing model for wireless tandem network coding,” in *2009 IEEE Wireless Communications and Networking Conference*, pp. 1–6, April 2009.

[38] F. Farhadi and F. Ashtiani, “Throughput enhancement of a random access wlan by combination of digital and analog network coding,” in *2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, June 2011.

[39] M. H. Amerimehr, F. Ashtiani, and M. B. Iraji, “An analytical approach for throughput evaluation of wireless network coding,” in *2009 IEEE International Conference on Communications*, pp. 1–5, June 2009.

[40] J. Cloud, L. M. Zeger, and M. Medard, “Mac centered cooperation x2014; synergistic design of network coding, multi-packet reception, and improved fairness to increase network throughput,” *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 341–349, February 2012.

[41] X. Tian, Q. Zhou, and Z. Cheng, “An algorithm of mac-based network coding for butterfly wireless networks,” in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, vol. 03, pp. 1097–1101, Oct 2012.

[42] G. Wei, L. Shengli, G. Peng, and Z. Fuzhong, “Finding coding opportunities in the air,” in *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 270–273, April 2010.

[43] R. Prasad, H. Wu, D. Perkins, and N. F. Tzeng, “Local topology assisted xor coding in wireless mesh networks,” in *2008 The 28th International Conference on Distributed Computing Systems Workshops*, pp. 156–161, June 2008.

[44] M. Orang, I. Keivanloo, and A. Movaghar, “A joint packet coding and attaching scheme for wireless networks,” in *2008 Fourth International Conference on Wireless Communication and Sensor Networks*, pp. 224–229, Dec 2008.

[45] D. T. Hai, “Re-designing dedicated protection in transparent wdm optical networks with xor network coding,” in *2018 Advances in Wireless and Optical Communications (RTUWO)*, pp. 118–123, Nov 2018.

[46] P. T. Compta, A. Antonopoulos, G. Schulte, L. Alonso, and C. Verikoukis, “Ncmob-mac: A network coding-based mac protocol with mobility support,” in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, May 2014.

[47] P. Chaporkar and A. Proutiere, “Adaptive network coding and scheduling for maximizing throughput in wireless networks,” in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom ’07, (New York, NY, USA), pp. 135–146, ACM, 2007.

[48] I. Tinnirello, L. Scalia, and F. Campoccia, “Improving ieee 802.11 performance in chain topologies through distributed polling and network coding,” in *2009 IEEE International Conference on Communications*, pp. 1–6, June 2009.

[49] S. Kafaie, Y. Chen, M. H. Ahmed, and O. A. Dobre, “Network coding with link layer cooperation in wireless mesh networks,” in *2015 IEEE International Conference on Communications (ICC)*, pp. 3672–3677, June 2015.

[50] J. Zhang, Y. P. Chen, and I. Marsic, “Mac-layer proactive mixing for network coding in multi-hop wireless networks,” *Computer Networks*, vol. 54, no. 2, pp. 196 – 207, 2010.
Wireless Multi-Hop Networking for Infrastructure Access.

[51] S. Kafaie, M. H. Ahmed, Y. Chen, and O. A. Dobre, “Performance analysis of network coding with ieee 802.11 dcf in multi-hop wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 17, pp. 1148–1161, May 2018.

[52] S. Shi, W. Ni, R. P. Liu, and S. Li, “Performance analysis of two-way mac layer network coding under finite relay buffer and non-negligible signalling overhead,” *Wireless Communications and Mobile Computing*, vol. 16, no. 18, pp. 3230–3243, 2016.

[53] F. C. Kuo, K. Tan, X. Li, J. Zhang, and X. Fu, “Xor rescue: Exploiting network coding in lossy wireless networks,” in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 1–9, June 2009.

[54] J. Qureshi, C. H. Foh, and J. Cai, “An efficient network coding based retransmission algorithm for wireless multicast,” in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 691–695, Sept 2009.

[55] A. Argyriou, "Network coding in ieee 802.11 wireless lans with an enhanced channel access scheme," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pp. 1–5, Nov 2008.

[56] Z. Zhang, T. Lv, X. Su, and H. Gao, "Dual xor in the air: A network coding based retransmission scheme for wireless broadcasting," in *2011 IEEE International Conference on Communications (ICC)*, pp. 1–6, June 2011.

[57] Y. Yan, Z. Zhao, B. Zhang, H. Mouftah, and J. Ma, "Mechanism for coding-aware opportunistic retransmission in wireless networks," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pp. 1–6, Nov 2009.

[58] Z. Zheng and P. Sinha, "Xbc: Xor-based buffer coding for reliable transmissions over wireless networks," in *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, pp. 76–85, Sept 2007.

[59] Z. Wang and M. Hassan, "Blind xor: Low-overhead loss recovery for vehicular safety communications," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 35–45, Jan 2012.

[60] T. Nage, F. R. Yu, and M. St-Hilaire, "Tcp-aware network coding with opportunistic scheduling in wireless mobile ad hoc networks," *2010 7th Ieee Consumer Communications and Networking Conference-Ccnc 2010*, pp. 354–358, 2010.

[61] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, "Tcp performance in coded wireless mesh networks," in *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 179–187, June 2008.

[62] S. Hassayoun, P. Maille, and D. Ros, "On the impact of random losses on tcp performance in coded wireless mesh networks," *2010 Proceedings Ieee Infocom*, 2010.

[63] D. Gomez, S. Hassayoun, A. Herrero, R. Aguero, and D. Ros, "Impact of network coding on tcp performance in wireless mesh networks," *2012 Ieee 23rd International Symposium on Personal Indoor and Mobile Radio Communications (Pimrc)*, pp. 777–782, 2012.

[64] J. Hansen, J. Krigslund, D. E. Lucani, and F. H. P. Fitzek, "Bridging inter-flow and intra-flow network coding for video applications: Testbed description and performance evaluation," in *2013 IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 7–12, Sep. 2013.

[65] D. S. Meitis, D. S. Vasiliev, A. Abilov, and I. Kaysina, "Comparison of tcp congestion control algorithms in network coded relaying scheme," in *2019 International Siberian Conference on Control and Communications (SIBCON)*, pp. 1–4, April 2019.

[66] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 63–68, Jan. 2006.

[67] Z. Narmawala, S. S. D. Ambani, and S. Srivastava, "Survey of applications of network coding in wired and wireless networks," 2007.

[68] S. Biswas and R. Morris, "Exor: Opportunistic multi-hop routing for wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 133–144, Aug. 2005.

[69] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 169–180, Aug. 2007.

[70] S. Zhang, S. Li, and D. Huang, "Mac independent routing methods for multi-services supporting wireless mesh networks," in *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Sept 2012.

[71] T. V. Vu, N. Boukhatem, T. M. T. Nguyen, and G. Pujolle, "Adaptive redundancy control with network coding in multi-hop wireless networks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1510–1515, April 2013.

[72] K. Mahmood, T. Kunz, and A. Matrawy, "Adaptive random linear network coding with controlled forwarding for wireless broadcast," in *2010 IFIP Wireless Days*, pp. 1–5, Oct 2010.

[73] P. Pahlevani, D. E. Lucani, M. V. Pedersen, and F. H. P. Fitzek, "Playncool: Opportunistic network coding for local optimization of routing in wireless mesh networks," in *2013 IEEE Globecom Workshops (GC Wkshps)*, pp. 812–817, Dec 2013.

[74] N. Papanikos and E. Papapetrou, "Deterministic broadcasting and random linear network coding in mobile ad hoc networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 3, pp. 1540–1554, 2017.

[75] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi, "I2mix: Integration of intra-flow and inter-flow wireless network coding," in *2008 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, pp. 1–6, June 2008.

[76] A. T. Lee and M. Gerla, "Performance evaluation of wireless network coding in tdma networks," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, pp. 1–6, Oct 2012.

[77] S. Tarapiah, C. Casetti, and C. F. Chiasserini, "Network coding in ad hoc networks: A realistic simulation study," in *IEEE INFOCOM Workshops 2009*, pp. 1–2, April 2009.

[78] C.-M. Cheng, H. T. Kung, C. K. Lin, C. Y. Su, and D. Vlah, “Rainbow: A wireless medium access control using network coding for multi-hop content distribution,” in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, pp. 1–10, Nov 2008.

[79] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “On mac scheduling and packet combination strategies for practical random network coding,” in *2007 IEEE International Conference on Communications*, pp. 3582–3589, June 2007.

[80] C. Fragouli, J. Widmer, and J. Y. L. Boudec, “A network coding approach to energy efficient broadcasting: From theory to practice,” in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pp. 1–11, April 2006.

[81] A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “Toward network coding-based protocols for data broadcasting in wireless ad hoc networks,” *IEEE Transactions on Wireless Communications*, vol. 9, pp. 662–673, February 2010.

[82] N. Rui, Z. Wuyang, and W. Guo, “Nc-aloha: A novel mac protocol,” in *2008 11th IEEE Singapore International Conference on Communication Systems*, pp. 1135–1139, Nov 2008.

[83] M. Amanowicz and J. Krygier, “On applicability of network coding technique for 6lowpan-based sensor networks,” *Sensors*, vol. 18, no. 6, p. 1718, 2018.

[84] T. Do-Duy and M. A. Vazquez-Castro, “Design of virtualized network coding functionality for reliability control of communication services over satellite,” *CoRR*, vol. abs/1803.04390, 2018.

[85] L. F. Vieira and M. A. Vieira, “Network coding for 5g network and d2d communication,” in *Proceedings of the 13th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 113–120, ACM, 2017.

[86] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, “Network coding meets tcp: Theory and implementation,” *Proceedings of the Ieee*, vol. 99, no. 3, pp. 490–512, 2011.

[87] M. Kim, M. Médard, and J. a. Barros, “Modeling network coded tcp throughput: A simple model and its validation,” in *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS ’11, (ICST, Brussels, Belgium, Belgium), pp. 131–140, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[88] J. Cloud, F. du Pin Calmon, W. Zeng, G. Pau, L. M. Zeger, and M. Medard, “Multi-path tcp with network coding for mobile devices in heterogeneous networks,” in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, pp. 1–5, Sept 2013.

[89] X. Lan, S. Li, and S. Zhang, "Network coded tcp throughput analysis in wireless mesh networks," in *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 225–232, Dec 2013.

[90] C. Casetti and M. Meo, "A new approach to model the stationary behavior of tcp connections," in *in Proceedings of IEEE INFOCOM 2000*, pp. 367–375, 2000.

[91] H. M. Ruiz, M. Kieffer, and B. Pesquet-Popescu, "Redundancy adaptation scheme for network coding with tcp," in *2012 International Symposium on Network Coding (NetCod)*, pp. 49–54, June 2012.

[92] H. Zhang, K. Xue, P. Hong, and S. Shen, "Congestion exposure enabled tcp with network coding for hybrid wired-wireless network," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, Aug 2014.

[93] T. V. Vu, N. Boukhatem, T. M. T. Nguyen, and G. Pujolle, "Dynamic coding for tcp transmission reliability in multi-hop wireless networks," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pp. 1–6, June 2014.

[94] H. M. Ruiz, M. Kieffer, and B. Pesquet-Popescu, "Tcp and network coding: Equilibrium and dynamic properties," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 1935–1947, Aug 2016.

[95] G. Bhat and J. McNair, "Analyzing effect of generation size in intra-session network coding for multiple flows of tcp traffic," in *2011 - MILCOM 2011 Military Communications Conference*, pp. 729–734, Nov 2011.

[96] C. Senger, S. Schober, T. Mao, and A. Zeh, "End-to-end algebraic network coding for wireless tcp/ip networks," in *2010 17th International Conference on Telecommunications*, pp. 607–612, April 2010.

[97] H. Liu and Y. Gu, "Tcp with hop-oriented network coding in multi-radio multi-channel wireless mesh networks," *IET Networks*, vol. 1, pp. 171–180, 2012.

[98] S. Gheorghiu, A. L. Toledo, and P. Rodriguez, "A network coding scheme for seamless interaction with tcp," in *2011 International Symposium on Networking Coding*, pp. 1–6, July 2011.

[99] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Allerton Conference on Communication, Control, and Computing*, October 2003.

[100] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros, "Network coding meets tcp," in *IEEE INFOCOM 2009*, pp. 280–288, April 2009.

[101] P. Garrido, D. Gomez, F. H. P. Fitzek, and R. Agueero, “When tcp and network coding meet wireless links,” in *European Wireless 2016; 22th European Wireless Conference*, pp. 1–5, May 2016.

[102] N. V. Ha and M. Tsuru, “Tcp/nc performance in bi-directional loss environments,” in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4, Jan 2019.

[103] N. V. Ha and M. Tsuru, “On the characteristics of tcp/nc tunneling in heterogeneous environments,” in *Advances in Intelligent Networking and Collaborative Systems* (F. Xhafa, L. Barolli, and M. Greguš, eds.), (Cham), pp. 340–349, Springer International Publishing, 2019.

[104] W. Bao, V. Shah-Mansouri, V. W. S. Wong, and V. C. M. Leung, “Tcp von: Joint congestion control and online network coding for wireless networks,” in *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 125–130, Dec 2012.

[105] S. Gheorghiu, A. L. Toledo, and P. Rodriguez, “Multipath tcp with network coding for wireless mesh networks,” in *2010 IEEE International Conference on Communications*, pp. 1–5, May 2010.

[106] C.-C. Chen, C. Chen, S. Y. Oh, J.-S. Park, M. Gerla, and M. Sanadidi, “Combocoding: Combined intra-/inter-flow network coding for tcp over disruptive manets,” *Journal of Advanced Research*, vol. 2, no. 3, pp. 241–252, 2011.

[107] C. C. Chen, G. Tahasildar, Y. T. Yu, J. S. Park, M. Gerla, and M. Y. Sanadidi, “Codemp: Network coded multipath to support tcp in disruptive manets,” in *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pp. 209–217, Oct 2012.

[108] N. Otsuki and T. Sugiyama, “A new ack packet transmission scheme for wireless network coding systems based on ieee802.11a,” in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pp. 1–5, May 2011.

[109] J. Rischke, F. Gabriel, S. Pandi, G. Nguyen, H. Salah, and F. H. P. Fitzek, “Improving communication reliability efficiently: Adaptive redundancy for rlnc in sdn,” in *2019 IEEE Conference on Network Softwarization (NetSoft)*, pp. 291–295, June 2019.

[110] H. Seferoglu and A. Markopoulou, “Network coding-aware queue management for unicast flows over coded wireless networks,” in *2010 IEEE International Symposium on Network Coding (NetCod)*, pp. 1–6, June 2010.

[111] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I2nc: Intra- and inter-session network coding for unicast flows in wireless networks," in *2011 Proceedings IEEE INFOCOM*, pp. 1035–1043, April 2011.

[112] D. Gómez, E. Rodríguez, R. Agüero, and L. Muñoz, "Reliable communications over wireless mesh networks with inter and intra-flow network coding," in *Proceedings of the 2014 Workshop on Ns-3*, WNS3 '14, (New York, NY, USA), pp. 4:1–4:8, ACM, 2014.

[113] P. S. David and A. Kumar, "Network coding for tcp throughput enhancement over a multi-hop wireless network," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pp. 224–233, IEEE, 2008.

[114] Y. E. Sagduyu and A. Ephremides, "Some optimization trade-offs in wireless network coding," in *2006 40th Annual Conference on Information Sciences and Systems*, pp. 6–11, March 2006.

[115] L. Scalia, F. Soldo, and M. Gerla, "Piggycode: A mac layer network coding scheme to improve tcp performance over wireless networks," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, pp. 3672–3677, Nov 2007.

[116] A. Paramanathan, S. Thorsteinsson, D. E. Lucani, and F. H. Fitzek, "On bridging theory and practice of inter-session network coding for csma/ca based wireless multi-hop networks," *Ad Hoc Networks*, vol. 24, pp. 148–160, 2015.

[117] A. Duda, "Understanding the performance of 802.11 networks," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, Sept 2008.

[118] B. O'Hara and A. Petrick, *The IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network IEEE Press, 1999.

[119] K. Szczypiorski and J. Lubacz, "Saturation throughput analysis of ieee 802.11 g (erp-ofdm) networks," *Telecommunication Systems*, vol. 38, no. 1-2, pp. 45–52, 2008.

[120] S. Mazuelas, F. A. Lago, D. González, A. Bahillo, J. Blas, P. Fernandez, R. M. Lorenzo, and E. J. Abril, "Dynamic estimation of optimum path loss model in a rss positioning system," in *2008 IEEE/ION Position, Location and Navigation Symposium*, pp. 679–684, IEEE, 2008.

[121] F. Lassabe, P. Canalda, P. Chatonnay, F. Spies, and O. Baala, "A friis-based calibrated model for wifi terminals positioning," in *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pp. 382–387, IEEE, 2005.

[122] M. G. Zapata, "Secure ad hoc on-demand distance vector routing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 106–107, 2002.

[123] S. Brahma, M. Chatterjee, and K. Kwiat, "Congestion control and fairness in wireless sensor networks," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 413–418, March 2010.

[124] P. Mehra, A. Zakhor, and C. De Vleeschouwer, "Receiver-driven bandwidth sharing for tcp," *Ieee Infocom 2003: The Conference on Computer Communications, Vols 1-3, Proceedings*, pp. 1145–1155, 2003.

[125] J. F. Kurose, *Computer networking: A top-down approach featuring the internet, 3/E*. Pearson Education India, 2005.

[126] V. Jacobson, "Congestion avoidance and control," *SIGCOMM Comput. Commun. Rev.*, vol. 18, pp. 314–329, Aug. 1988.

[127] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models for the latency and steady-state throughput of tcp tahoe, reno, and sack," *Ieee-Acm Transactions on Networking*, vol. 11, no. 6, pp. 959–971, 2003.

[128] A. Gurtov, T. Henderson, S. Floyd, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm." RFC 6582, Apr. 2012.

[129] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, "Tcp westwood: congestion window control using bandwidth estimation," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 3, pp. 1698–1702 vol.3, 2001.

[130] D. Katabi, S. Katti, W. J. Hu, H. Rahul, and M. Medard, "On practical network coding for wireless environments," *2006 International Zurich Seminar on Communications: Access - Transmission - Networking, Proceedings*, pp. 84–85, 2006.

[131] J. Jin and B. Li, "Adaptive random network coding in wimax," in *2008 IEEE International Conference on Communications*, pp. 2576–2580, May 2008.

[132] J. L. van Wyk, A. S. J. Helberg, and M. J. Grobler, "An evaluation of configuration parameters affecting network coding on the data link layer in a wireless network," in *AFRICON, 2011*, pp. 1–5, Sept 2011.

[133] X. Zhang, H. Zhu, and J. Zhang, "Ranc: Relay-aided network coding in multi-hop wireless networks," *Computer Communications*, vol. 32, no. 5, pp. 974 – 984, 2009.

[134] Y. E. Sagduyu and A. Ephremides, "Crosslayer design for distributed mac and network coding in wireless ad hoc networks," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pp. 1863–1867.

[135] Y. E. Sagduyu and A. Ephremides, "Cross-layer optimization of mac and network coding in wireless queueing tandem networks," *IEEE Transactions on Information Theory*, vol. 54, pp. 554–571, Feb 2008.

[136] Y. E. Sagduyu and A. Ephremides, "Joint optimization of mac and network coding for cooperative and competitive wireless multicasting," in *12th International Conference on Parallel and Distributed Systems - (ICPADS'06)*, vol. 2, pp. 8 pp.–, 2006.

[137] S. Thangam and E. Kirubakaran, "A survey on cross-layer based approach for improving tcp performance in multi hop mobile adhoc networks," in *2009 International Conference on Education Technology and Computer*, pp. 294–298, April 2009.

[138] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *41stAllerton Conf.Communication,Control and Computing, Monticello, IL, US.*, Oct. 2003.

[139] S. Wunderlich, F. Gabriel, S. Pandi, F. H. P. Fitzek, and M. Reisslein, "Caterpillar rlnc (crlnc): A practical finite sliding window rlnc approach," *IEEE Access*, vol. 5, pp. 20183–20197, 2017.

[140] M. Prabakaran, A. Mahasenan, and A. Kumar, "Analysis and enhancement of tcp performance over an ieee 802.11 multi-hop wireless network: single session case," in *2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005.*, pp. 29–33, Jan 2005.

[141] K. Saravanan and B. Venkatalakshmi, "Mobility based minimal network coding (mbmnc) algorithm for mobile ad-hoc network," in *2013 International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 290–296, July 2013.

[142] P. Aguilera, *802.11Ax: A Hyperconnected World and the Next-Generation WiFi*. Independently published, 2018.

[143] M. Ruffini, "Multidimensional convergence in future 5g networks," *Journal of Lightwave Technology*, vol. 35, pp. 535–549, Feb 2017.