2012

Improving Fairness and Utilisation in Ad Hoc Networks

Arabi, Mohamed

http://hdl.handle.net/10026.1/1167

http://dx.doi.org/10.24382/4537
University of Plymouth

Improving Fairness and Utilisation in Ad Hoc Networks

by

Mohamed Arabi

A thesis submitted to the University of Plymouth

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Communications and Electronics

Faculty of Technology

February 2012

Improving Fairness and Utilisation in Ad Hoc Networks

Mohamed Arabi

Abstract

Ad hoc networks represent the current de-facto alternative for infrastructure-less environments, due to their self-configuring and resilience characteristics. Ad hoc networks flexibility benefits, such as unrestrained computing, lack of centralisation, and ease of deployment at low costs, are tightly bound with relevant deficiencies such as limited resources and management difficulty.

Ad hoc networks witnessed high attention from the research community due to the numerous challenges faced when deploying such a technology in real scenarios. Starting with the nature of the wireless environment, which raises significant transmission issues when compared with the wired counterpart, ad hoc networks require a different approach when addressing the data link problems. Further, the high packet loss due to wireless contention, independent of network congestion, requires a different approach when considering quality of service degradation and unfair channel resources distribution among competing flows. Although these issues have already been considered to some extent by researchers, there is still room to improve quality of service by reducing the effect of packet loss and fairly distributing the medium access among competing nodes.

The aim of this thesis is to propose a set of mechanisms to alleviate the effect of packet loss and to improve fairness in ad hoc networks. A transport layer algorithm has been proposed to overcome the effects of hidden node collisions and to reduce the impact of wireless link contention by estimating the four hop delay and pacing packet transmissions accordingly. Furthermore, certain topologies have been identified, in which the standard IEEE 802.11 faces degradation in channel utilisation and unfair bandwidth allocation. Three link layer mechanisms have been proposed to tackle the challenges the IEEE 802.11 faces in the identified scenarios to impose fairness in ad hoc networks through fairly distributing channel resources between competing nodes. These mechanisms are based on monitoring the collision rate and penalising the greedy nodes where no competing nodes can be detected but interference exists, monitoring traffic at source nodes to police access to the channel where only source nodes are within transmission range of each other, and using MAC layer acknowledgements to flag unfair bandwidth allocation in topologies where only the receivers are within transmission range of each other. The proposed mechanisms have been integrated into a framework designed to adapt and to dynamically select which mechanism to adopt, depending on the network topology. It is important to note that the proposed mechanisms and framework are not alternatives to the standard MAC protocol but are an enhancement and are triggered by the failure of the IEEE 802.11 protocol to distribute the channel resources fairly.

All the proposed mechanisms have been validated through simulations and the results obtained from the experiments show that the proposed schemes fairly distribute channel resources fairly and outperform the performance of the IEEE 802.11 protocol in terms of channel utilisation as well as fairness.

Table of Contents

Table of Figures:

Acknowledgments

Thanks go to my supervisor Dr. Bogdan V. Ghita for his ongoing encouragement, advice and support throughout the research, my family in particular my dad Haj Abdelkader, my mother for their ongoing support, to my brothers Moulay Abdallah, Houari, Mustapha, Ibrahim, to my sister Fatima, to my wonderful supportive wife who played a great role in the completion of this research, to my children Abdelkader and Kheira, also to my friends including Meftah and my colleagues from the CSCAN group.

Glossary

ACK	Acknowledgement
AODV	Ad hoc On-demand Distance Vector Protocol
ARQ	Automatic Repeat-reQuest
ATCP	Adaptive Transmission Control Protocol
BEAD	Best Effort ACK Delivery
BEB	Binary Exponential Back off algorithm
COPAS	Contention-based Path Selection
CW	Contention Window
CTS	Clear to Send
CSMA/CS	Carrier Sense Multiple Access with Collision Avoidance
DARPA	Defence Advanced Research Projects Agency
DCF	Distributed Coordination Function
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
ECN	Explicit Congestion Notification
EPLN	Early Packet Loss Notification
FBDMAC	Fair Bandwidth Distribution Medium Access Control
FCRD	Fair Channel Resources Distribution
FHDMAC	Four Hop Delay Medium Access Control
ICMP	Internet Control Message Protocol
IOS	International Organisation for Standards
MAC	Medium Access Control
NAV	Network Allocation Vector
NS	Network Simulator
NTRC	Neighbourhood Transmission Rate Control
PRNET	Packet Radio Networks
RED	Random Early Detection
RREP	Route Reply
RREQ	Route Request
RTS	Request to Send

RTT	Round Trip Time
SURAN	Survivable Radio Network
TCP	Transmission Control Protocol
TCP-AP	Transmission Control Protocol with Adaptive Pacing
TRCAF	Transmission Rate Control through Acknowledgement Feedback
TULIP	TCP Unaware Link Improvement Protocol

AUTHOR’S DECLARATION

At no time during the registration for the degree of Doctor of philosophy has the author been registered for any other university award. The word count is 37886.

Chapter 1. **Introduction**

An ad hoc network is a distributed, wireless and multi hop network architecture that relies on no pre-existing network infrastructure for its deployment and functionality. The nodes of an ad hoc network can be mobile, which means that they can move in and out of the network at any time. Consequently, the network topology itself can change over time, based on the mobility and connection or disconnections of the nodes. An ad hoc network is considered to be an unreliable and unstable communications environment as the nodes are bandwidth and energy constrained. However, during their evolution, new challenges have risen. The applications of ad hoc networks have expanded into many areas and environments such as emergency operations in natural disasters, communicating between vehicles and between vehicles and roadside equipment, and recently commercial and educational purposes. Due to the various advantages and services ad hoc networks could offer such as search and rescue operations where the area is likely to be large, hostile and time is very critical. In 1979, mobile ad hoc networks were deployed in the search of the lives and yachts which were lost during the storm that hit the Fastnet race. a high interest in this kind of communication technology has arisen by many industries and governments. This led to extensive research in all aspects involved in the area such as security, quality of service and management issues. Furthermore, this research focuses on the weaknesses which ad hoc networks encounter such as bandwidth loss and unfair channel resources allocation in certain topologies.

The main focus of prior research was on improving channel utilisation in multi hop wireless environments aiming to overcome the challenges raised by hidden nodes in a single flow. The proposed schemes varied from optimising the transmission layer and delaying acknowledgements to optimising routing protocols, controlling and monitoring the queue size as well as prioritising packets, introducing cross layer interaction for a better synchronisation between layers, and modifying the binary exponential back off algorithm. However, previous research did not extensively explore the area of fair bandwidth allocation and channel utilisation in the presence of multiple flows with multiple hops competing for the same channel resources. This led to the investigation of the performance of the standard MAC protocol in such conditions, which was found to be extremely poor in scenarios where the senders or receivers are outside the transmission range of each other, whereas the intermediate nodes are not. In addition, in scenarios where the competing flows are symmetric with each other and all nodes are outside the transmission range, but interfere with each others transmission, the MAC protocol proved to be inefficient is such cases in terms of sharing access to the medium and favoured the last successful transmission leaving the other competing flows to starve. These observations and the poor performance of the 802.11 protocol in such scenarios led to the proposal of four mechanisms. The initial scheme paces TCP packet transmission when the number of hops to destination exceeds three hops, to tackle the challenges raised by the hidden nodes and intra flow contention and improve the channel utilisation in terms of the throughput achieved. The other three mechanisms proposed, tackle the unfair bandwidth allocation in the presence of multiple flows competing for the single medium available while exceeding or at least maintaining the throughput achieved by the standard 802.11 protocol. It is important to stress that the three mechanisms are not replacements to the MAC protocol but enhancements to it and are only engaged when the MAC protocol fails to fairly share the channel resources equally among the competing flows.

1.1 Aims and Objectives of the Research

The focus in this thesis is improving throughput in ad hoc networks and distribute the available bandwidth fairly among competing flows over the single shared medium by alleviating the effect of packet loss through pacing packet transmission and distributing the channel resources fairly amongst competing nodes through monitoring the transmission and collision rates of the contending nodes.

The main objectives of this research are as follow:

- Investigate and understand the state of the art and the background literature within the ad hoc network research domain in terms of the transmission control protocol discrimination between congestion and transmission induced losses, the proposed solutions both at the transport and data link layers, fair bandwidth allocation and channel utilisation including the solutions proposed such as modifying the binary exponential back off algorithm and controlling the contention window size growth. Furthermore, mobility induced losses are also investigated to determine the impact of the routing protocols on the performance of TCP. Also, the interaction between the transport and link layers and how they interfere with each others performance is investigated.

- Improve the throughput achieved by an ad hoc network on a multi hop chain at the transport layer by introducing a four hop delay to pace packet transmissions on chains consisting of four hops and more which would alleviate contention induced losses and reduce the impact of the hidden nodes on throughput as the nodes in the same flow would refrain from contending with each other over the available medium.

- Improve fair bandwidth distribution between competing flows in multi flow and multi hop scenarios by proposing three solutions based on the topologies to enhance the performance of the MAC protocol where it experiences unfair bandwidth distribution between the competing nodes. The first mechanism is triggered when the packets experience collisions and no competing node from another flow is detected as it is outside the transmission range but within interference range. In this case, the node would monitor the collision rate and rather than backing off transmission, the node requests access to the channel aggressively as a greedy node is capturing the available medium. Furthermore, if the source nodes are within transmission range, hence, can detect each other’s transmissions, then the transmitted packets are monitored by the competing nodes to determine whether to continue transmitting or refrain if the competing nodes are starving in the second mechanism. In the third mechanism, if the receiving nodes are within transmission range and compete with each other over the medium, then the nodes would monitor their competing nodes transmission and set a flag in the MAC layer acknowledgement to indicate contention that propagates along the path instructing the intermediate nodes to stop transmitting to relieve contention and allow the starving nodes to pick up transmission.

- Produce a proof-of-concept implementation that combines the proposed solutions and integrate them into a framework that dynamically applies the appropriate mechanism to the chosen scenario in a simulation environment.

- Evaluate the performance of the proposed solution independently as well as integrated in the framework and compare the results to prior research in terms of throughput, overall channel utilization, and fairness as well as the standard 802.11 MAC protocol.

In summary, the following figure illustrates a model of the final framework.

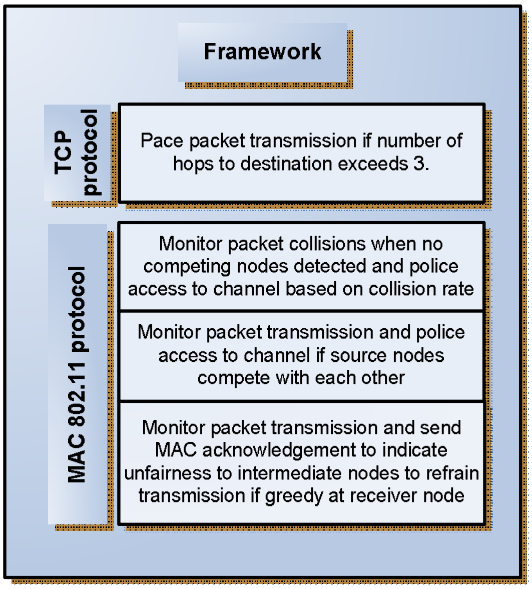

Framework
TCP protocol
Pace packet transmission if number of hops to destination exceeds 3.
MAC 802.11 protocol
Monitor packet collisions when no competing nodes detected and police access to channel based on collision rate
Monitor packet transmission and police access to channel if source nodes compete with each other
Monitor packet transmission and send MAC acknowledgement to indicate unfairness to intermediate nodes to refrain transmission if greedy at receiver node

Figure 1: Overview of the framework

Figure 1 shows a template for the framework to be proposed and the four mechanisms that it integrates. The first mechanism is a transport layer solution which paces TCP packet transmission via introducing a four hop delay by which the successive packet is not transmitted until the previous packet has travelled four hops away. The other three mechanisms are designed to enhance the performance of the IEEE 802.11 protocol in scenarios where it fails to fairly distribute the channel resources among competing nodes.

In relation to performance evaluation, the identified methods will be benchmarked against prior studies as well as more traditional approaches. The evaluation will be performed using a simulation environment of variable complexity, in order to ensure the efficiency as well as the robustness and the scalability of the proposed solutions.

1.2 Thesis Structure

Chapter 2 looks into the state of the art of the current challenges faced by ad hoc networks such as channel utilisation, classification of congestion loss versus transmission loss, the hidden node problem, issues related to mobility as well as challenges relating to fair medium access, as well as the proposed solutions by previous researchers in the field of ad hoc networks. Chapter 3 introduces four algorithms to tackle the channel utilisation challenges and fair bandwidth distribution based on the topologies for which the MAC protocol exhibits unfair bandwidth allocation. Chapter 4 evaluates the proposed mechanisms independently in a simulation environment. An integration of the proposed solutions to form a framework is presented in Chapter 5 and the efficiency of such a framework is given in Chapter 6. Finally, a conclusion to the research as well as its limitations and future work to further enhance the performance of the proposed solutions is provided in Chapter 7.

Chapter 2. **Challenges in Ad Hoc and TCP Environments**

2.1 Introduction

In this Chapter, a brief overview of ad hoc networks and the technologies involved for their successful establishment and operation are provided. Also, the challenges that ad hoc networks face are discussed, in conjunction with the research that has been undertaken in order to improve their performance. These challenges are divided into two areas: channel utilisation and bandwidth distribution fairness among competing nodes sharing the same channel resources available.

The history and the evolvement of ad hoc networks can be split into three categories or generations.

The first generation of ad hoc networks were developed in the 1970’s by the Defence Advanced Research Projects Agency **(**DARPA) and were called “Packet Radio Networks”. This new communication technology was initially developed and primarily used for fast establishment of military communication during the deployment of forces in unknown and hostile terrains. Military operations in battle fields are not static and do not rely on a fixed pre-placed communication infrastructure. Ad hoc networks provide a suitable framework as they use radio frequency technology to transmit and receive data and they offer multi hop wireless network connectivity and can span over a large area. It is important to stress that Packet radio networks existed before the Internet and in fact were the reason behind its existence in 1977 where the International Organisation for Standards established a subcommittee to develop the Open Systems Interconnection model which was adopted in 1983.

The second generation of ad hoc networks (1980’s to mid 1990’s) focused on the issues faced with the Packet Radio Networks (PRNET) and include the Survivable Radio Network (SURAN) which scales to tens of thousands of nodes and is resilient to security threats [1]. The major developments in this period were Global Mobile Information Systems and the Near Term Digital Radio Systems [1].

The period from the 1990’s onward witnessed a huge advancement in terms of new and viable technologies being introduced such as Bluetooth, wireless ad hoc sensors and notebook computers. The availability of these technologies and their fast development brought the idea of commercial ad hoc networks (third generation) [1]. Some examples include business environments such as the gaming industry, military battlefields, disaster relief operations, and many others [2].

2.2 Ad Hoc Networks Characteristics

The main properties of ad hoc networks are summarised as follows [3]:

- May operate in a standalone fashion or may be connected to the wider Internet

- Ad hoc networks are independent and infrastructure-less networks as they require no fixed or pre-placed communication architecture.

- Ad hoc networks are multi hop wireless networks as routes between two communicating nodes may include multiple hops. Nodes will be able to communicate with each other directly as long as they are all located within the transmission range of each other and inside the same cluster. However, for nodes residing far from the transmission range or in other words outside the cluster; intermediate nodes are used to pass on the transmitted data hop by hop to destination.

- Because nodes can move arbitrarily and freely in an unpredictable way, the routes to other nodes can change rapidly resulting in network partitions and in some cases packets being lost due to route establishment and route failures.

- In ad hoc networks, each node may be equipped with one or more radio interfaces that differ in transmission and/or receiving capabilities and operate across different frequency bands. In addition, nodes might be configured with different software and hardware. This results in variation of processing capabilities of each node. Therefore, designing network protocols and algorithms can be complicated and complex to some level, requiring adjustments to the changes that might occur to the network's nodes such as power and channel conditions; in order to prevent the instability of the network and maintain an acceptable level of quality of service.

- The scalability of ad hoc networks is an important necessity to the successful operation of such networks; as many ad hoc networks engage relatively large networks, as in sensor networks for example [4]. Therefore, optimising protocols to handle large amounts of data, as well as nodes in terms of routing protocols, has a direct impact on the effectiveness and usability in such a type of network.

2.3 Review of Ad Hoc Networks

Ad Hoc Networks may contribute, together with other emerging networking technologies, to the foundations for future communications. The strength of an Ad Hoc Network lies in the fact that it is dynamic and easy to set up at a very low cost, through the lack of centralised management and control restrictions. On the other hand, the topological variations cause a state of network instability, which affects quality of service and the performance of applications in such networks.

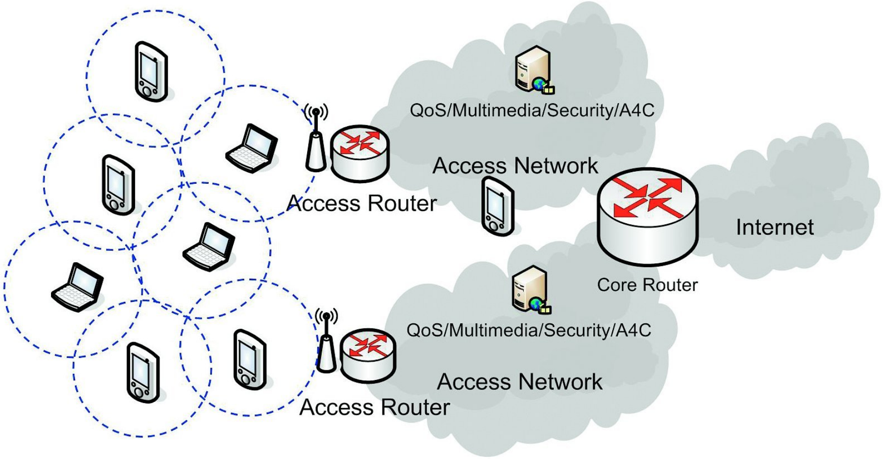


Figure 2: Illustration of an Ad Hoc Network

Figure 2 illustrates an Ad Hoc Network being deployed, where the nodes communicate with each other to forward each others packet and connect to the wider Internet through two access routers and a core router. The figure presents only an example on how an Ad Hoc Network may operate. However, such networks may take various forms and as their name suggests they can be deployed in any manner and shape. In the past, ad hoc networks were initially developed and primarily used for fast establishment of military communication during the deployment of forces in unknown and hostile terrains. Military operations are dynamic and do not rely on a fixed pre-placed communication infrastructure. Ad hoc networks create a suitable framework as they provide multi hop wireless network connectivity [3].The first ad hoc networks were called "packet radio" networks, and were sponsored by the Defence Advanced Research Projects Agency (DARPA) in the early 1970s. It is interesting to note that these packet radio systems existed before the invention of the Internet, and were behind the motivation of the original Internet Protocol Suite. Later DARPA experiments included the Survivable Radio Network (SURAN) project, which took place in 1983 to address main issues of the Packet Radio Network (PRNet). The main objectives were to develop network algorithms to support networks that can scale to tens of thousands of nodes and resist security attacks [5].

Although the IEEE 802.11 ("Wi-Fi") wireless protocol technology is considered a very low-grade ad-hoc protocol by specialists in the field, it incorporates an Ad Hoc Networking system where no wireless access points are present. The IEEE 802.11 system only handles traffic within a local "cloud" of wireless devices where each node transmits and receives data, but does not route anything between the network's nodes. However, higher-level protocols can be used to combine various IEEE 802.11 ad-hoc networks into mobile ad hoc networks [6].

One of the main issues in ad hoc networks is energy constraint; nodes typically have limited power supply and mainly rely on rechargeable batteries. Thus, processing power is limited which, in turn, limits the services and applications that can be performed by each node. Additional power is required by each node for transmitting traffic, as a node can act as both an end system and a router to forward packets.

The aforementioned issues pose great challenges to the deployment of ad hoc networks in general; especially the limited resources obstacle. Other issues that are still to be solved include areas such as: security, high capacity wireless technologies and routing[7]. This research addresses and focuses on the challenges related to channel utilisation and fair distribution to the channel resources.

2.3.1 Environment

Improving quality of service in wireless ad hoc networks has been a topic of extensive research for many years due to their important role in situations where there is no backbone infrastructure in place, from disaster scenarios to military applications. Ad Hoc Networks gained their popularity for their ease of establishment and flexibility as nodes can forward each other’s traffic or serve as communication hosts. Also, some wireless nodes may be connected to the wired networks thus may serve as edge routers and connect the Ad Hoc Network to the Internet. Despite their advantages, ad hoc networks suffer from unfair bandwidth distribution among competing flows due to the single channel access, interference, hidden nodes and the absence of a central station to manage the network. The IEEE 802.11 [8] MAC protocol uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [9] protocol to reduce collisions by sensing the channel before each transmission. If the channel is not idle, the node backs off transmission by a period specified by the Exponential Back off Algorithm [10] implemented in the Distributed Coordination Function (DCF) [11] and schedule its transmission to start when the existing one finishes by updating the Network Allocation Vector (NAV) with the transmission time required and which is specified in the RTS/CTS messages when the competing nodes are within the same transmission range of each other, hence, the nodes can hear each other’s transmission and schedule them accordingly. On the other hand, if the nodes are in the interference range of each other, updating NAV is not possible as the RTS/CTS message cannot be read and the nodes may transmit at the same time leading to collisions which exponentially increases the back off time after each collision encountered. This leads to unfair channel distribution as the MAC protocol has no means of scheduling transmission appropriately and would always favour the last successful transmission as the other nodes would still be backed off.

2.3.2 Medium Access Control protocol 802.11

The 802.11 MAC Protocol was designed to provide wireless connectivity between nodes. In order for this to happen, the MAC protocol implements a Distributed Coordination Function (DCF) which employs the Carrier Sense Multiple Access with Collision Avoidance protocol. Each node can capture the wireless medium and send or receive data at a time to prevent packet collision. In order to illustrate how DCF operate consider the following illustration of the MAC four way handshake presented in figure 3.

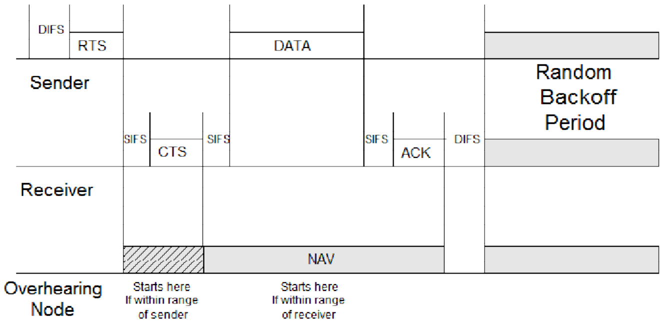


Figure 3: 802.11 DCF with RTS-CTS messages[1]

Each node wishing to send packets will have to sense the channel for 50 microseconds [12], this interval is referred to as a DIFS (Distributed Inter-Frame Spacing) period. If the channel is found busy, the node will have to back off its transmission by a value defined by multiplying a randomly chosen contention window value (CW) in the range [0, 31] which is exponentially increased after each collision until it reaches the maximum contention window size of 1023 (CWmax - 1), which is reset to 31 after a successful transmission and a time slot, defined as 20 microseconds in the IEEE 802.11 standard [13] , as described by the following formula [14]:

$$BackOffTime = random() \times aSlotTime \qquad (1)$$

Once this back off time expires, the node sends a request to send (RTS) message to its down stream node and the downstream node replies with a clear to send (CTS) message. Then the sender node transmits the frame and the downstream node replies with MAC acknowledgement. In Figure 3, SIFS is a Short Inter-Frame Space and is defined as 10 microseconds [12]. If the frames are not received within a SIFS period, they are assumed lost and the MAC layer will retransmit them up to seven times before it drops them. If a CTS message is not received after four consecutive transmissions of the RTS message, the MAC layer assumes there is no route to destination and a route discovery is initiated by the routing layer.

[1] http://www.vocal.com/network/802_11_dcf.html

2.3.3 Transmission Control Protocol (TCP)

TCP was first introduced in 1981. It remained the predominant transport layer protocol of the Internet as it always evolved to meet the changes the Internet had undergone. TCP is a connection oriented transport layer protocol that ensures reliable and end to end transmission rather than a node to node transmission. It supports flow control in order to prevent the sender from overflowing the receiver’s buffer. In addition, the protocol uses sequence numbers to monitor and provide in-order data delivery. TCP is very efficient in wired networks where the main cause for packets loss is congestion. TCP implements a congestion control mechanism in which the congestion window size is increased until a packet is lost then reduced accordingly in order to use up all the available bandwidth [15]. However, in wireless networks, congestion is not the only reason for packet loss as other factors such as interference, hidden node problem, and contention also lead to packet loss. These factors make TCP inefficient in wireless networks and improvements have been recommended in prior research which will be investigated in this thesis. Furthermore, this thesis addresses the contention caused by TCP and an algorithm to reduce this contention as well as the effects of hidden nodes on the performance of TCP will be provided in order to improve channel utilisation.

2.3.4 Ad hoc On-demand Distance Vector Protocol (AODV)

Upon reception of a data packet to be forwarded to destination at the network layer, the routing table is checked to verify if a route to destination exists. If there is a route available, then the packet is forwarded to the intermediate node. Otherwise, a route discovery process is initiated to find a route to destination using control packets. The node broadcasts a route request (RREQ) packet to the wireless nodes within range which in turn forward the packet to the nodes within their range. In order to avoid routing loops, the RREQ packets are labelled with sequence numbers. Upon reception of the RREQ packet, the destination responds by sending back a route reply (RREP) packet to the source node using a reverse route rather than flooding the network as with RREQ packets. Once the path has been found, the node starts transmitting data packets. If the RREP packet is not received after sending the RREQ packets seven times, the node drops the data packets as no route to destination can be found. This would lead to severe degradation to quality of service because if the RREQ packets were lost because of interference, the node would assume that there is no route to destination available. The protocol drops all data packets in its queue, even though there is actually a route but it is not possible to reach the destination due to other factors such as interference.

2.3.5 Dynamic Source Routing Protocol (DSR)

DSR is an efficient routing protocol designed for multi hop wireless networks. It is similar to AODV in that it constructs a route to destination on demand. To establish a route to destination, the source node floods the network with route request packets. The address of each node that the route request packet traverses is stored in the packet and, if the destination node receives the route request packet, it replies with a route reply packet back through the same route. If a node received the route request packet recently or if its address is listed in the route record of the route request packet, then the node discards the packet and does not forward it to stop the packet from entering a loop. The DSR protocol is a reactive protocol; to avoid flooding the network, it periodically checks if the route is still available. However, if a link fails, the DSR protocol enters a route discovery mode rather than fixing the route locally, unless there is an alternative route cached in the protocol.

2.3.6 Destination Sequenced Distance Vector Protocol (DSDV)

In DSDV, the routes to destination are tagged with a sequence number to indicate how old the route is. Upon reception of a new route with a higher sequence number the old route is discarded and replaced with the new one. Two mechanisms are used in DSDV to anticipate route failures. The first one is based on not receiving replies to broadcast messages from destination and the second one is named *settling time*, which predicts when the route becomes stable [16]. DSDV differs from AODV in that it is a proactive protocol and the entire routing table is updated periodically and transmitted to the neighbouring nodes and not on demand. In addition, in DSDV the routes are chosen based on the best signal strength and the least delay.

This research does not attempt to improve on any of these routing protocols; however, AODV, DSR and DSDV are the most suitable routing protocols for ad hoc networks as they take into account the frequent route changes in such networks. Furthermore, the AODV routing protocol has been chosen to be used in all simulations to validate the proposed algorithms for compatible comparison with prior research conducted. Furthermore, the AODV and DSDV routing protocol are the most suited for ad hoc networks as they respond to route failures promptly and are resolved locally [19].

2.4 An overview of Ad Hoc Networks Challenges

The research areas in ad hoc networks can be split into five sections: quality of service, routing, security, resource management and auto configuration.

Quality of service is an important factor for the successful deployment of any network. It varies from network to network. In an ad hoc network, quality of service is a challenging issue due to their unique characteristics as described previously. In order to improve quality of service in an ad hoc network, the protocols implemented at the transport, network and link layers have to be revised and adapted to the nature of ad hoc networks. As a first step towards improving quality of service, the transmission control protocol will be enhanced. TCP works very well in wired networks because it assumes that if a loss occurs; it is due to congestion which is true most of the time. However, this is not the case with wireless networks as there are other reasons to packet loss apart from congestion, like interference, lossy links, transmission errors, node mobility, and route failures in ad hoc networks. In such environments, the standard TCP mis-reacts to non-congestion induced packet losses, which leads to poor performance in terms of channel utilisation. At this stage, three questions can be asked, are there other transport protocols apart from TCP that perform better in such conditions? If no, how can TCP determine the exact cause of the lost packet? Was it congestion, link failure, transmission error or hidden nodes and what strategy is best suited to overcome these problems? Many researchers believe that resolving transmission errors has to be done at the link layer then if necessary instructing TCP to take the right action by providing a cross-layer interaction between the transport and the medium access protocols [17]. This research proposes an algorithm to pace packet transmission at the transport layer which prevents congestion and reduces the effects hidden nodes have on packet loss.

The network layer receives requests from the transport layer and sends requests to the link layer. Researches in [18] argue that any routing challenges in ad hoc networks are better solved at the network layer as it is the first layer to discover a link failure. Improved routing protocols and algorithms for ad hoc networks were proposed by prior studies [19], [20] such as proactive protocols like DSDV and reactive protocols like AODV. However, none of them has become a standard protocol for ad hoc networks' configurations [21]. The reason is due to the nature of ad hoc networks as nodes may not be static and the topology may be changing rapidly and nodes themselves act as routers in contrast to wired networks were there are fixed and preplaced routers [18]. This issue will be looked into in detail in the following sections.

Security is a critical issue to the deployment of ad hoc networks. As nodes in ad hoc networks are self-organised it is not possible to propose a universal security mechanism. In addition, it is not an easy task to monitor the activities of each node due to the lack of a central network administration [4]. Also, it is a very difficult task to provide an efficient security mechanism that has low power consumption, which is an important factor in ad hoc networks where nodes rely largely for their functionality on their battery power.

Resource Management is to some extent a unique area to ad hoc networks. Also, it is a crucial service to have as nodes have limited resources especially battery power. So, being able to monitor the battery power of the nodes gives an indication as to which applications the nodes are capable of running, which nodes to use to route traffic and which nodes are expected to disconnect from the network [22]. Other limited resources are: bandwidth, CPU power, RAM and others.

Auto configuration is a very useful service to be provided in ad hoc networks as nodes operate in standalone fashion. It is often too complex for end users to configure routing protocols, security mechanisms and other services. Having an Auto Configuration mechanism will ease the deployment of such networks [23].

As mentioned earlier, the research motivation is to improve quality of service in terms of throughput by alleviating the effect of packet loss and distributing the channel resources, or in other words bandwidth, fairly among competing nodes in the presence of multiple flows sharing the same channel. In order to achieve this, the following sections address the challenges faced by ad hoc networks in detail, specifically, channel utilisation, fair bandwidth allocation and mobility induced losses, and the proposed solutions by previous researchers.

2.5 Channel Utilization Challenges in Ad Hoc Networks

The following subsections list the challenges that TCP faces in terms of its inability to differentiate between congestion and interference losses in ad hoc networks and the prior research to improve channel utilisation and reduce packet loss.

2.5.1 Classification of Congestion Loss versus Transmission Loss

As described before, TCP proves to be inefficient in ad hoc networks, due to its inability to distinguish between congestion induced losses and transmission “mainly interference” losses. The high packet loss, caused by interference and contention, requires a different approach when considering quality of service degradation. Providing an accurate classification of congestion versus transmission packet losses enables TCP to discriminate between them and react accordingly. Researchers in [17] used the information provided by the explicit congestion notification (ECN) messages in their proposed algorithm to detect congestion and differentiate between congestion loss and error loss. Researchers have followed two different paths: link layer-based and transport layer-based solutions. Since the problem in wireless networks is high link error rate and it lies at the physical layer, the link layer can sense the problem faster than any other layer. One of the studies focusing on the link layer introduced a Hybrid Automatic Repeat Request scheme where the forward error correction bits are added to the existing error detection bits to not only detect corrupted data but correct it as well [24]. This would leave the standard TCP unchanged to maintain compatibility with the standard TCP/IP suite. The drawback of this proposal is degraded throughput and performance[25] because not only too much overhead is created but also duplicated effort is created since TCP also tries to ensure reliable packet delivery. In order to avoid duplicating effort a cross layer interaction mechanism is a necessity [4] and modifying TCP may not be avoidable. Therefore, other researchers proposed versions of TCP such TCP Reno, Westwood, Tahoe and improved versions such as TCP New Reno have also been proposed [26]. However, none of them became a standard for ad hoc networks as they fail to differentiate between congestion and mobility or interference losses.

In the following subsections some techniques proposed by researchers to solve the aforementioned problem will be looked into and investigated.

Snoop Protocol

Resolving the high link error rate at the link layer by using Hybrid Automatic Repeat Request (HARQ) protocol does create too much overhead to ensure reliable transmission. In order to avoid creating too much packet overhead, TCP has to be aware that the retransmission of the lost packet is being taken care of at the link layer and therefore refrains from retransmitting the lost packets. To achieve this TCP awareness, researchers in [15] proposed the snoop protocol, a data link layer enhancement that detects lost packets if duplicates have been received. The snoop protocol also uses a local timeout based on the round trip time and favours retransmission from the link layer over TCP retransmission and discards any duplicate acknowledgements. The Snoop protocol resides between a wired node and a wireless node at the base station. It caches data packets and forwards them to their destination and waits for an acknowledgement. If no ACK has been received after the timeout has expired, or if a duplicate acknowledgement is received, the snoop protocol locally “at the link layer” retransmits the lost packet. This protocol proves to be efficient to some extent as it reduces overhead and delay in contrast to the HARQ protocol. However, the need for a base station to regulate traffic is a must as it cannot be deployed otherwise. Also, setting a timer at the base station requires significant system resources as it needs to determine an accurate estimation of the round trip time for packets on the wireless link [27].

TCP aware Link Layer retransmission

TCP awareness of link layer local retransmission has been explored in the snoop protocol. However, researchers in [27] do not detect a lost packet based on the round trip time as in the Snoop protocol but use packet timestamps at the link layer to detect out of order reception of packets which in turn means a packet has been lost and use Explicit Retransmission Notification to avoid TCP reducing its transmission window size “congestion window size” and preventing degradation in throughput. The scheme proposed achieves the desired results as long as packet error rate was less than 0.1, from which the mechanism experienced longer delays to acknowledgement packets. In addition, the Snoop protocol can only detect the missing packets after TCP times out which results in retransmissions and delays. In [28], the authors propose to utilise the bit vector returned with the cumulative acknowledgements to precisely detect the missing packet. Furthermore, the Snoop protocol requires the presence of a base station for its deployment and would not be a solution to be adopted in ad hoc networks.

TCP Unaware Link Layer retransmission (TULIP)

The authors in [28] propose a TCP Unaware Link Improvement Protocol (TULIP). TULIP takes advantage of the generous TCP timeout and retransmits the lost packet before TCP notices the loss.

With TULIP, TCP forwards a data packet to the link layer to be transmitted and initialises a timer. TULIP implements a stop and wait algorithm different from the traditional stop and Wait ARQ in that TULIP does not wait to receive an acknowledgement; instead, if no ACK has been received before TULIP times out, it sends a new packet. The idea behind is that a packet could have been received correctly but the ACK for it has been lost. TULIP acknowledgements contain a sequence number for cumulative acknowledgements to show the highest in order packet received correctly and a bit vector to show missing packet. TULIP then retransmits the missing packets and the process starts again. Also, TULIP takes advantage of being implemented on top of the MAC protocol. TULIP includes a MAC Acceleration feature in order to reduce link delays and overhead. If there is a data packet to send, it can be encapsulated in the Link ACK if it is big enough to carry it. If the packet is too big, then the channel is reserved immediately for a long packet to be transmitted. Although TULIP was originally implemented for use between a base station and mobile node, it can easily be deployed in ad hoc networks. The authors of [28] also proved that TULIP has better throughput and performance over Snoop and pre-existing TCP improvement. However, the experiment conducted showed that TULIP suffered higher packet drops than Snoop. The proposed scheme recovered the dropped packets before TCP timeout and the achieved throughput was higher than the Snoop protocol.

ATCP

The previous protocols were not designed specifically for ad hoc networks, but for cellular networks where the base station provides enough information about the state of the wireless link. On the other hand, TULIP can easily be deployed in ad hoc networks. However, the very good performance exhibited in cellular networks does not hold for ad hoc networks due to their unique characteristics. For this reason, researchers in [17] proposed ATCP, which is a thin layer inserted between IP and TCP. The standard TCP semantics are preserved in order to ensure compatibility and easy deployment. Nodes not running ATCP can still communicate with nodes running ATCP but they will suffer from the degraded TCP performance in wireless environment. ATCP listens to network state information provided by the network layer feedback (Explicit Congestion Notification and Internet Control Message Protocol messages). If an ECN message is received, ATCP puts TCP into congestion control state without waiting for TCP timeout. If an ICMP message is received, ATCP puts TCP onto a persist state where TCP stops transmitting packets and waits until a route has been found. Once a new route has been found TCP re-computes the congestion window and does not use the old congestion window size as techniques in [29] and [30] do. The reason is that the congestion window is based on the state of the link and resuming transmission at the same rate as the old congestion window size could lead to congestion. In the case where packets are being received out of order, duplicate acknowledgements are generated. Once the third duplicate acknowledgement is received, the standard TCP enters congestion mode. ATCP prevents this by discarding the third duplicate acknowledgement then freezes TCP and retransmits packets from TCP's buffer until an acknowledgement is received then TCP is put back to normal state. The authors of [17] showed through simulations that ATCP outperforms the standard TCP in terms of bandwidth utilisation. They further claim that it is an ideal solution for the problems that TCP encounters in ad hoc networks. However, the performance of the proposed scheme was only compared with the performance of TCP which proves to be inefficient in ad hoc networks and was not compared to other solutions in the field nor compared to TCP variants such as New Reno and TCP-SACK.

In [30], a link breakage is detected upon the reception of an Explicit Link Failure Notification (ELFN) message. When the source node receives the ELFN message, TCP stops transmitting packets and periodically sends a packet. Once an acknowledgement is received then TCP resumes transmission. The technique used in [28] is better than that of [30]. If an intermediate router detects a link failure, it generates a Route Failure Notification (RFN) message. When the source node receives the RFN message, it stops TCP from transmitting any packets until a Route Re-establishment Notification (RRN) is received. Both papers do not address high bit error rate, out of order packets, and TCP's misinterpretation of losses in terms of BER and congestion.

2.6 MAC 802.11 behaviour in terms of Fairness

The flexibility and the absence of a base station in ad hoc networks can lead to severe degradation in the network's performance in terms of collisions, throughput and, implicitly, fair bandwidth allocation. From an architectural perspective, the MAC 802.11 protocol, with its Distributed Coordination Function (DCF) [11], aims to eliminate packet collisions and to reduce the hidden node phenomenon [31] where all nodes are within transmission range of each other. This is achieved by the use of the Binary Exponential Back-off (BEB) Algorithm [10] which randomly delays transmission to avoid collisions. When the communicating nodes transmit data to nodes that are outside the transmission range of each other and within interference range, the receiving nodes are not capable of scheduling their transmissions and the BEB algorithm becomes inconsistent due to the use of exponential contention window and leads to unfair channel distribution among the competing nodes of different flows.

In order to reduce collisions between nodes in the same transmission range, the Distribution Coordination uses the four way handshake RTS/CTS/DATA/ACK before any data transmission. When a node has a packet to transmit it senses the channel for an interval specified in the Distributed Inter Frame Space (DIFS). If the channel is found to be busy, the node defers its transmission by a duration calculated using the BEB algorithm. If the channel is sensed idle, the node transmits an RTS message to the next hop in the route to destination. The nodes that are in the same transmission range also hear this RTS and update their Network Allocation Vector (NAV) with the required transmission duration specified in the RTS message in order to prevent transmitting while there are still another transmission taking place. On the other hand, the nodes that are outside the transmission range but within the interference range cannot extract this information and may transmit while the channel is busy which leads to packet collisions. In the event of a collision the BEB algorithm exponentially increases the contention window, which exponentially increases the chances of the node not capturing the channel. In contrast, the node that did not experience collisions dominates access to the channel leaving other nodes to starve.

The problem of fair distribution of channel resources in ad hoc wireless networks has been investigated in several papers. Researchers in [32] show that TCP performance in ad hoc networks faces a great challenge in terms of fairness. In [33] and [14], the authors have pointed that the 802.11 protocol does not provide good fairness in environment where multiple flows are in the same contention region which leads some TCP flows to starve. They also pointed out that the reason for such unfairness is due to the exponential increase of the contention window size (CW) after each calculation of the back off time when a collision is detected. Therefore, nodes that have experienced collisions have to wait much longer time before they can try to access the channel and the nodes that have not experienced collisions will have more chances accessing the channel as their contention window size has not been exponentially increased.

In order to solve this problem, researchers in [33] show the weaknesses of the back off algorithm in the presence of many nodes in the same contention region. They proposed a linear increase of the contention window size (CW) based on how many bits the node has sent and received during a time interval and a Fairness Index Metrics previously proposed in [34] to categorise the channel resources distribution. The researchers acknowledge the use of empirical thresholds chosen after being varied in different simulations. In addition, the authors believe that the proposed scheme can be further enhanced by optimising the threshold values used in their mechanism. On the other hand, in [35], the authors proposed a logarithmic increase of the contention window based on how many bytes the nodes sent and received. However, the authors of [14] increased the contention window by the first back off duration multiplied by its log and by a time slot after the second collision. The logarithmic increment proposed by the authors in [14] improves throughput in large size systems of 80 nodes and more. However, in smaller size networks the throughput achieved is not significant in comparison with the exponential back of algorithm.

In [36], the authors show that 802.11 does not differentiate between frames that are sensed out of the transmission range of a node and differs its transmission by the same extended inter-frame space (EIFS) duration which results in collisions. This resulted in the nodes transmitting at the same time, causing collisions and leading to unfair bandwidth utilisation. In order to overcome this problem, the authors of [36] observed the use of the fixed extended inter-frame space (EIFS) and if the nodes happen to back off by the same duration, collisions occur. The researchers proposed to calculate the EIFS based on the length of the frames sensed in the network. On the other hand, rather than looking for a solution at the MAC layer, other researches proposed fixes to the unfairness problem of the 802.11 MAC protocol at the transport and network layers.

In [37], the authors proposed a fair bandwidth distribution scheme among TCP flows by controlling the queue output rate at the network layer according to the severity of the contention experienced by the network and preventing TCP from reaching its maximum congestion window size so that the node does not become greedy and prevents its competitors from transmitting. Despite the efforts the researchers have made in order to tackle the unfairness of the MAC protocol, there are limitations and drawbacks to their work in terms of throughput reduction as the scheme experienced 11.2% throughput degradation, collisions and delays. On the other hand, the authors of [38] proposed a solution at the link layer and presented a proportional fair contention resolution algorithm that uses a persistent back off for all nodes in order to fairly distribute contention loss and achieve fairness in a clique and based on packet loss, the node gets more aggressive. The authors recognise that the performance of the mechanism has not been explored in the presence of mobile nodes and random channel errors. These characteristics are crucial for the practical deployment of the MAC protocol in ad hoc networks. Researchers in [39] proposed a max-min-per-link fair share algorithm that indicates whether to increase or decrease the contention window size in order to achieve fair bandwidth share at each node based on the amount of traffic sensed in its carrier sensing range. Therefore, the maximum fairness that could be achieved is determined by severity of the contention in the wireless network [39]. Furthermore, the researchers examined the performance of the proposed algorithms under static conditions and will include mobile environments in future work. Also, the scenarios evaluated included multiple flows competing for the shared medium. However, testing the proposed schemes in multi hop scenarios would have strengthened their practicability in ad hoc networks.

The researchers in [40] proposed a contention window based fairness back off algorithm that introduces the concept of authority and ordinary nodes where authority nodes are granted access to the channel with more priority in contrast to ordinary nodes that access the channel only after N successive failures. The proposed scheme was only tested on a single static chain topology consisting of five nodes where the only challenge is the hidden node. Furthermore, the authors claim that the throughput achieved is the same as the theoretical throughput derived from the proposed generic theoretical model. However, no calculations are demonstrated nor the topology conditions such the number of nodes, flows and distances have been integrated in the theoretical model proposed.

In [41], a neighbourhood random early detection (NRED) algorithm was implemented at the network layer hence no modification to the MAC protocol in order to detect contention and improve fairness. Each node monitors its queue size and broadcasts it through network congestion notification (NCN) control packets to the nodes within its transmission range to decide whether to drop packets from the queue to ease contention. The proposed method is interesting as the authors make use of the random early detection typically based on the queue size. However, in NRED the random early detection is based on the channel utilisation. Also, the performance of NRED was tested in a mobile environment and fair bandwidth distribution was improved but not totally achieved. Furthermore, the authors acknowledge that broadcasting the NCN packets by the bottleneck node created some packet overhead. In [42], the researchers proposed a fair share estimation algorithm of the channel resources between nodes sharing the same channel. Each node in the network estimates how much channel resource is being assigned to other nodes based on how many packets they have transmitted and then modify their contention window size according to a predefined fairness metrics in order to achieve the desired fairness. The researchers also noticed that the proposed algorithm does sacrifice some throughput in order to achieve an acceptable level of fair bandwidth distribution.

The MAC 802.11 protocol operates at different speeds. For example the 802.11b operates at 1, 2, 5.5 and 11 Mb/s. Assume two nodes (one at 11 Mb/s and the other at 1 Mb/s) are transmitting data via an access point “can be an intermediate node acting as a router”. The two nodes can detect each other transmission as they are in the same carrier sense range. So, one node can only transmit when the other node is not. The question is: How much does the node have to wait before it can start transmitting? According to [12] there are different ways to provide fairness namely *packet-level* fairness that is based on the number of packets allowed for transmission per node at a time and *rate-proportional* fairness which based on time division. Researchers in [12] propose a different technique to control fairness based on adjusting the Maximum Transmission Unit (MTU) to better utilise the bandwidth available to each node and to ensure that all nodes have an equal time for transmission. The access point would instruct each node connected to it to use a different MTU depending on the channel bandwidth available. This method would make the 802.11 carrier sense multiple access with collision avoidance (CSMA/CA) protocol imitate the time division multiple access (TDMA) mechanism to achieve fair access to the channel.

2.7 Route Failures and the impact on Channel Utilisation

The throughput achieved by TCP greatly suffers from the frequent route change in mobile ad hoc networks. In reality, most packet losses occurring in such networks are due to frequent route failures [43] and interference. Frequent route change is inevitable due to the nature of these networks. However, the impact on TCP performance is immense because not only packets are lost but also TCP times out and enters onto congestion mode where no congestion has been experienced. To address this problem many mechanisms where introduced. Explicit link failure notification (ELFN) [30] is a mechanism that works in the way that if a link fails, the sender node will be notified, which in turn freezes TCP. Periodically, TCP will send a packet until it receives an acknowledgement, then resume transmission at the same transmission rate as before the link failed without entering onto congestion mode. This technique improves overall throughput. However, researchers in [44] further improved this mechanism by adding two new techniques: the first one being Early Packet Loss Notification (EPLN) and the second is Best Effort ACK Delivery (BEAD) that intend to avoid initiating a route failure by first exploring alternative route.

2.7.1 Best Effort ACK Delivery

If acknowledgements have encountered a link failure, the node will try and retransmit the ACK with the highest sequence number among all acknowledgements encountering the route failure via a cached route if available. If it is not possible to retransmit the ACK; the node will send a notification to the intermediate nodes to indicate that the acknowledgement has not been delivered using a route obtained by reversing the source route. The intermediate nodes will do the same until it reaches the TCP receiver, which will retransmit the acknowledgement using a cached route. If it is not possible to retransmit the acknowledgement and it is not possible to send a notification because of a link failure, the acknowledgement gets dropped and the node detecting the link failure sends a notification to the source of the original notification, which will try and send another notification via a different cached route.

2.7.2 Early Packet Loss Notification

If an intermediate node encounters a link failure; it tries to retransmit the data packet via a cached route if available. If it is not possible, the intermediate node will send a notification to the previous node which will try and do the same “retransmit the data packet” but only if it has previously salvaged the data packet until it reaches the TCP sender. The network layer at the TCP sender sends an Internet Control Message Protocol (ICMP) message to TCP for each lost packet. The ICMP message includes the sequence number of the lost packet and also an added feature which is whether the packet has been salvaged or lost. Upon reception of an ICMP message TCP freezes its transmission, retransmit lost packets if they have been salvaged and resumes transmission when it receives an ACK.

2.8 Interaction between TCP and MAC layer for wireless networks

In [45]-[46], researchers show that there is a great impact of the MAC protocol on the performance and throughput of TCP. This is because TCP and MAC protocols do not operate at the same transmission window size. In other words, TCP grows its transmission window much larger leading to contention on the wireless channel[45]. Researchers in the following sections present some mechanisms to solve the problem and improve throughput and avoid packet loss.

2.8.1 Link RED and Adaptive Pacing

The authors in [45] and [47] show that buffer overflow, collisions due to hidden nodes and reduced spatial reuse, have a great impact on TCP performance and throughput. In addition, the experiments conducted by the researchers show that there exists an ideal size for the transmission window called Contention Avoidance Region at which TCP achieves best throughput [45] as it does not lead to buffer overflow at the MAC layer. Extensive simulations and experiments showed that, if the transmission window size W is $<h/4$ where h is the number of hops from source to destination for flows consisting of more than 4 hops, then the channel is less utilised and if W is $>h/4$ than the throughput reduction increases. In [45] and [48] the authors propose two techniques: link RED and an adaptive link layer pacing to improve TCP throughput in a multi hop wireless ad hoc network and better utilise the channel by distributing network load over intermediate nodes.

The Link RED algorithm

This algorithm is based on monitoring the average number of retries at the link layer. The MAC protocol drops a packet after sending the RTS message seven times and does not receive a CTS message or it drops the packet after sending it four times and does not receive an ACK. If the average number of retries is smaller than a minimum threshold the packet is not dropped nor marked because the effects of the hidden node are rare. On the other hand, if the average number of retries is larger than the minimum number of retries then the packet is dropped or marked and the dropping/marking probability is calculated as follows:

$$\text{Dropping/Marking probability} = \min\left\{ \frac{\text{average retry } - \text{ minimum threshold}}{\text{maximum threshold} - \text{minimum threshold}}, \max P \right\} \quad (2)$$

Where P is the packet loss probability[45], [47].

Adaptive Pacing

When the average number of retries is larger than the minimum threshold, the adaptive pacing technique is used in order to utilise the available channel in a better way. The standard MAC protocol prevents nodes from contending the channel via the use of a random back off time plus the time needed to transmit one packet to the intermediate node. In Adaptive pacing technique one more packet transmission time is added to the standard postponement timing [45], [47].

The authors of [48] proposed a similar solution but at the transport layer and continuously adapted TCP's transmission rate based on the Round Trip Time i.e. if a link is experiencing high contention the RTT for a packet will be longer and the TCP window size has to be reduced accordingly to reduce the effect of contention.

2.8.2 Dynamic Delayed ACK

In [45], the authors show that the degradation in TCP throughput is not mainly due to buffer overflow in wireless environments but due to the hidden node phenomenon. In order to improve the throughput they proposed link RED and adaptive pacing in order to detect contention on the wireless link and to maximise the number of packets transmitted simultaneously. To achieve this they proved that there exists an optimal transmission window size at which TCP achieves best throughput. If this window is larger than the optimal size, then loss occurs and on the other hand if the window is smaller than the channel is not fully utilised. However, in order to be able to work out the optimal transmission window size; the number of hops has to be known which is not possible in practice [49].

Researchers in [49] took a different approach to tackle the same problem. They support researchers in [45] which argue that, in order to improve TCP throughput, spatial reuse has to be improved by increasing the number of TCP packets transmitted simultaneously. To achieve this, the authors noticed that in a wireless environment where nodes use the 802.11 MAC protocol as the access medium, it is necessary to do a four way handshake before the transmission of each TCP packet in order to reserve the channel and the same applies to acknowledgement packets and route discovery packets and they all compete over the same channel. The authors in [49] prove that delaying acknowledgements "an ACK is sent for every *n* packets received" rather than sending an ACK for every packet leads to more TCP packets flow and thus increases throughput and improves channel utilisation. The default ACK delay is fixed to two packets in the delayed acknowledgement mechanism [50] but the authors believe that this has to be flexible and they proposed a dynamic delayed acknowledgement mechanism where the delay is increased according to the sequence number of acknowledged packets. Therefore, if the congestion window size is smaller then the delay in sending acknowledgements is small and vice versa. The authors in [49] showed through simulations that the dynamic delayed acknowledgement approach increases TCP throughput by about 50%, because when the window is small, packets are acknowledged quickly thus leading to increased throughput and also when the window is large "more packets awaiting transmission" the acknowledgements are delayed to increase the TCP packets flow in the channel.

2.8.3 Contention-based Path Selection

In an ad hoc network, a node can capture the channel for a long period of time and prevents other nodes from transmitting data. This behaviour is due to a number of reasons namely the interaction of the MAC layer and TCP back off policies [46]. Researchers in [46] noticed that TCP throughput degradation is caused by packets not getting acknowledged because intermediate nodes captured the channel and prevented the receiver from acknowledging received packets. This results in TCP backing off and retransmitting packets already received. They proposed a contention-based path selection (COPAS) algorithm to resolve the conflict between sender's and receiver's transmissions. COPAS works by choosing two disjoint routes - one for forward transmission and a reverse route for acknowledgements. Disjoint path routing is not a new idea as it has been previously explored in relation to DSR [51] and AODV [52] where the source floods the network with route request packets in order to determine all possible routes. In COPAS, in order to determine the contention experienced on a route, nodes append a weighted average of the number of times it has backed off transmission to the route request packets during route discovery process until it reaches destination and then rebroadcast it [46]. The destination waits a certain period of time for route request packets from the various routes in order to learn about all possible routes available and then selects the two least contented routes and replies with two route reply packets for the two disjoint routes. COPAS also monitors all available routes over time and selects a new less contended route for forward and reverse transmissions with up to date information in case of route failure or if the existing route experiences high contention.

2.8.4 Hidden Node Problem and intra flow contention

In figure 4, if node A is sending packets to node B, node D, which is out of the interference range of node A, cannot hear the RTS messages and may transmit packets at the same time as node A which will cause collision at node B. The previous can happen since node A is out of the interference range of node D and node D cannot receive the CTS messages sent by node B as it is out its transmission range. So it cannot update its Network Allocation Vector by the duration set in the CTS and RTS messages and it will not reschedule its transmission after the end of the communication between nodes A and B. This will yield packets being dropped at node B [53], [31]. This problem significantly reduces the network's performance and the channel utilisation.

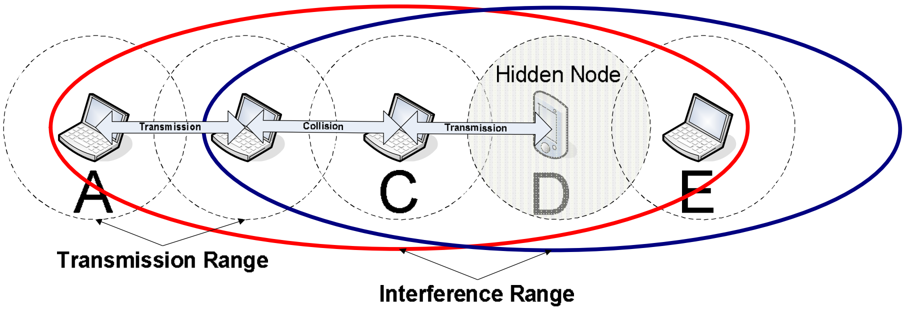


Figure 4: Illustration of Hidden node problem

The MAC 802.11 protocol listens to the channel state. If the channel is busy, the MAC backs off transmission using the exponential back off algorithm before trying retransmission. If the MAC backs off transmission seven times it drops the packets. The MAC listens to the channel state by sending a RTS message. If it gets back a CTS message, it transmits data and waits for a link layer acknowledgement from the downstream node. Now, the downstream node has data to transmit so it senses the channel by sending a RTS to its downstream node. Here, the source node could have data to transmit and senses the channel too. At this stage, both the source and downstream nodes are trying to capture the channel to send data. However, only one node is allowed to capture the channel otherwise a collision will happen and packets will be dropped and because of this, one node will not be able to transmit the data and will back off transmission. If it retries seven times it will drop the packet and initiate route recovery process. In a multi hop route, all nodes compete with each other from source to destination to forward packets. Researchers in [54] referred to this scenario as intra flow contention. This aforementioned hidden node problem is further illustrated in figure 5 below in addition to the challenges raised by intra flow contention.

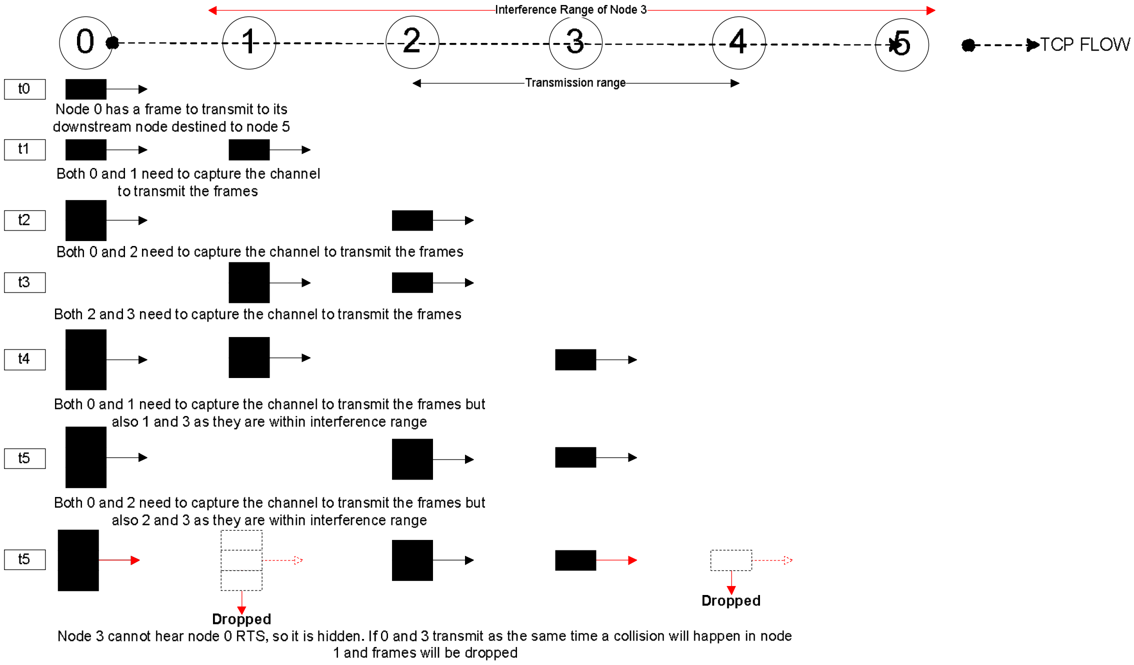

Interference Range of Node 3
0
1
2
3
4
5
TCP FLOW
Transmission range
t0
Node 0 has a frame to transmit to its downstream node destined to node 5
t1
Both 0 and 1 need to capture the channel to transmit the frames
t2
Both 0 and 2 need to capture the channel to transmit the frames
t3
Both 2 and 3 need to capture the channel to transmit the frames
t4
Both 0 and 1 need to capture the channel to transmit the frames but also 1 and 3 as they are within interference range
t5
Both 0 and 2 need to capture the channel to transmit the frames but also 2 and 3 as they are within interference range
t5
Dropped
Dropped
Node 3 cannot hear node 0 RTS, so it is hidden. If 0 and 3 transmit as the same time a collision will happen in node 1 and frames will be dropped

Figure 5: Intra Flow Contention

In addition to the hidden node problem, intra flow contention [48], [54] is where nodes within interference range compete with each other in capturing the single channel resources available in order to transmit the frames if they have any. The MAC 802.11 protocol has a mechanism that schedules nodes' transmissions. However, nodes can be greedy and capture the channel for a long period which can lead to packet losses due to timeouts. The previous can happen when the nodes have large packets in their queues that take longer transmission time, in contrast to nodes with fewer packets queuing to be transmitted. Figure 5 above provides an illustration of intra flow contention and the hidden node problem where the packets get dropped in $\mathbf{t}_5$ at nodes 1 and 4 due to the hidden node phenomenon.

2.8.5 TCP-AP

In order to reduce the effect of intra flow contention and the hidden node problem on the performance of the transmission control protocol researches in [48] presented TCP-AP which stands for TCP with adaptive pacing. The authors provided a formula that estimates the time it takes for a packet to travel four hops away from the sender and referred to it as the four hop delay (FHD). The FHD is calculated as per formula presented in (3) in which RTT stands for round trip time and *h* is the number of hops to destination.

$$FHD = 2 * \left(\frac{RTT}{h} + \frac{SizeofPacket - SizeofAcknowledgement}{bandwidth} \right) \quad (3)$$

Furthermore, the researchers scheduled transmission of packets as follow:

1. For each received acknowledgement calculate a new four hop delay which consists of 70% of the previous four hop delay and 30% of the current four hop delay (FHDi) using the formula presented in equation (3) as follow:

   $$FHDnew = 0.7 * FHDold + 0.3 * FHDi$$

2. Calculate the coefficient of variant over the most recent round trip time samples as follows: $COVrtt = \frac{\sqrt{\left(\frac{1}{N-1} \sum_{i=1}^{N} \left(RTTi - \overline{RTT} \right)^2 \right)}}{\overline{RTT}}$

3. Pace TCP packet transmission by the inter packet delay duration calculated as $InterPacketDelay = FHDnew * (1 + 2 * COVrtt)$

This mechanism improves the overall channel utilisation over TCP New Reno and reduces the impact of the hidden node problem channel utilisation and throughput achieved.

2.9 Conclusion

In this chapter, a detailed literature on the previous research conducted in areas related to the challenges ad hoc networks encounter and the proposed solutions have been presented. Previous studies have shown that the performance of TCP in ad hoc networks is very poor in terms of channel utilisation due to intra flow contention, hidden nodes, inability to differentiate congestion from contention packet loss, route failures and cross layer interaction. The various techniques and mechanisms proposed to improve fairness in ad hoc networks did not tackle scenarios consisting of multiple flows with multiple hops which are what this thesis aims to address. Furthermore, prior research proposed electing some nodes to act as access points to regulate traffic and distribute channel resources fairly, however, this does not suit the nature of ad hoc networks as the network topology changes regularly. Providing solutions to improve channel utilisation and fair bandwidth distribution at the transport layer via controlling the maximum transmission unit or the TCP's congestion window does not eliminate the hidden node problem. However, providing mechanisms at the MAC layer to fairly distribute the available bandwidth among competing nodes seems more appropriate as the MAC protocol was designed to provide access to the wireless channel. All the provided solutions suffered from throughput reduction and the effects of the hidden nodes were reduced but not eliminated.

In the next Chapter, simulations will be conducted to illustrate how bad TCP performs in ad hoc networks and a better estimation of the four hop delay will be provided to further improve the performance of TCP in the presence of hidden nodes and intra flow contentions and outperform TCP-AP. Furthermore, in order to improve the unfair bandwidth utilisation among competing nodes sharing the same channel resources various novel approaches have been explored and designed based on penalising greedy nodes according to the collision rate at the MAC layer and through observing and monitoring the transmission rate of the competing neighbouring nodes within transmission range.

Chapter 3. **Improving Utilisation and Fairness in TCP/Ad Hoc Environments**

3.1 Introduction

This research aims to alleviate the effect of hidden nodes and channel contention on TCP performance and fairly distribute the shared medium among competing nodes in ad hoc networks. Researchers have taken different routes and proposed various solutions as described in the previous chapter. The 802.11 MAC protocol used in ad hoc networks is designed to eliminate the hidden node problem through the use of the four-way handshake RTS-CTS-DATA-ACK and to reduce packet drops due to collisions by listening to the channel before transmitting data. The request to send and clear to send control packets are utilised to synchronise nodes transmissions and share the available channel equally. However, even though the MAC protocol performs as required when the nodes are within transmission range of each other, it does not do so when the nodes are outside the transmission range of each other, which leads to unfair and degraded channel utilisation among the same and also between different flows sharing the same channel resources, as some of the nodes may capture the channel for a long period and leave the other nodes to starve. In order to overcome the unfair bandwidth allocation exhibited by the MAC protocol, prior research proposed solutions such as a linear rather than an exponential increase to the back off algorithm at the MAC layer [33], controlling the queue output rate at the network layer as in [37] and [41] and varying the EIFS duration based on the length of the frames [36]. Furthermore, TCP experiences degraded channel utilisation in ad hoc networks due to a number of reasons including high packet loss and high packet error rate. Prior research attempted to alleviate the high packet loss and to increase the throughput that TCP achieves through a number of mechanisms such as the use of the ECN and ICMP messages [17] or ELFN messages [30] to discriminate between congestion and link failure losses.

This chapter proposes an improvement to TCP with an adaptive pacing mechanism [48] to pace packet transmission through introducing a four hop delay, which reduces the effects of hidden nodes and improves channel utilisation. In addition, three novel algorithms are proposed to fairly distribute access to the available medium among the competing flows in identified scenarios where the standard MAC protocol fails to achieve fairness. These algorithms are based on monitoring collision rate that the transmitting nodes experience in scenarios where no competing nodes can be detected and regulating access to the channel in the presence of competing nodes within transmission range based on the transmission rate of the nodes and through information added to the MAC layer acknowledgements about the state of the network.

3.2 TCP New Reno Performance in Wired and Wireless networks

In order to determine the dimension of the problem in terms of how bad TCP New Reno performs in wireless environments, and to be able to compare its performance with implemented techniques, two experiments have been conducted. In all scenarios covered in this research, the wireless nodes are static in order to isolate mobility-induced losses. Previous studies have shown that the performance of New Reno is extremely penalised by its inability to distinguish between error or interference and congestion losses, which led TCP New Reno to assume that all losses are due to congestion losses. In addition, TCP New Reno performance is even worse in wireless multi hop flows when compared to single hop flows.

3.2.1 Performance of TCP New Reno in Wireless and Wired Networks:

In order to determine how TCP New Reno performs in both wired and wireless environments; two simulations have been carried for 200 seconds and in both simulations a ten hop chain topology was designed with a single flow. In the first simulation, nodes were connected to each other via wired links, and in the second one via wireless links. The throughput was calculated for both simulations and the results are shown in the following graph. Many previous studies have illustrated that New Reno performance is severely degraded in wireless networks compared to wired networks. This is logical since New Reno was originally designed for wired networks. The plot presented in figure 6 demonstrates that New Reno suffers about 94% throughput reduction in wireless environment in comparison to the throughput it achieves in wired environments. This is due to factors such as interference induced losses.

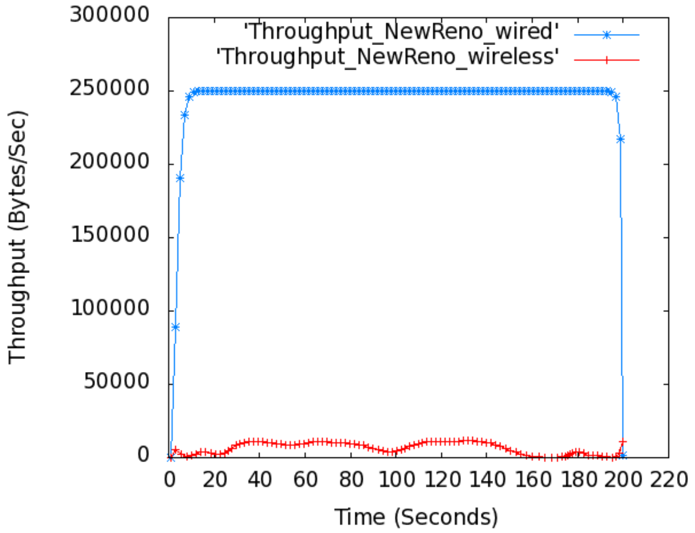


Figure 6: Throughput of TCP New Reno in wired and wireless environments

Figure 6 shows that the performance of TCP New Reno is extremely poor in the wireless environment compared to its counterpart the wired environment. This confirms what researchers have discussed in [55]. The significant reduction in throughput in wireless networks is due to many factors mainly the inability of New Reno to differentiate between congestion losses and wireless losses, interference, hidden nodes, channel errors, and contention. The TCP New Reno performance worsens when multiple flows compete over the shared medium available in wireless environments.

3.2.2 Performance of TCP New Reno in a single hop and four hops wireless environments:

In order to study the effects of a multi hop flow on the performance of TCP New Reno, two simulations have been run for 200 seconds; one with one hop and one with four hops. The throughput versus simulation time has been plotted in figure 7.

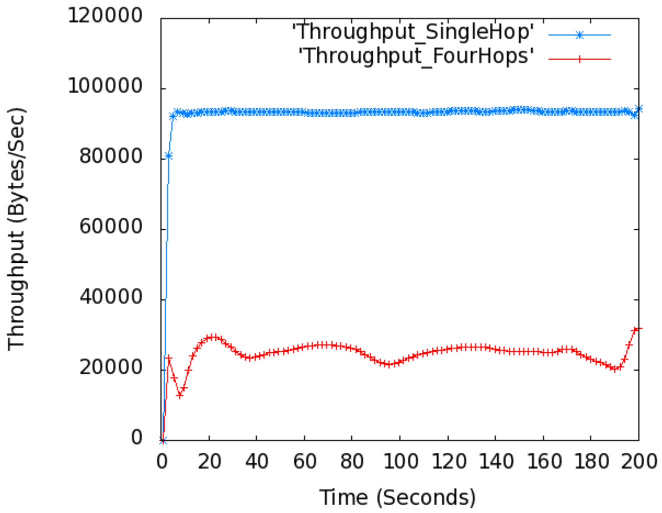


Figure 7: Throughput of TCP New Reno in a single hop and four hops wireless environments

From the graph presented in figure 7, New Reno experiences a remarkable throughput reduction of 76% in the four-hop chain than in the single hop chain. Researchers in [56] and [57] have shown that New Reno experiences even poorer performance in longer wireless chains.

3.3 Pacing TCP transmissions to Improve Channel Utilisation

Researchers in [45], [48] and [49] have shown that the 802.11 protocol severely penalises TCP performance where the chain consists of multiple hops and the situation worsens where there are multiple flows crossing and or sharing the same channel resources as will be investigated further in the following sections. The authors have also shown that delaying packet transmissions does have an impact on the overall network performance and does not underutilise the available medium. The main observation is that there is an optimal delay when pacing packet transmissions depending on the considered topologies and this optimal delay varies depending on the number of hops to destination. Furthermore, the delay duration is not constant during the transmission time i.e. it changes over time depending on the contention experienced on the wireless channel. The 802.11 protocol is designed to sense the channel every time a frame is ready to be sent using the four way RTS-CTS-DATA-ACK handshake. If the channel is free, the frame is sent to the intermediate node which in turn does the same four way handshake with its intermediate node until the frame reaches its destination. If the channel is busy, the MAC protocol backs off its transmission using the exponential back off algorithm and drops the frame if it retransmits the RTS message four times and does not receive a CTS message. The hidden node problem imposes significant reduction in throughput because if two senders are outside the interference range of each other and the receivers are within interference range the intermediate nodes will drop frames depending on which gets access to the channel first even though the source nodes transmitted successfully. However, delaying packets transmission will allow intermediate nodes to forward the first packet before the second one is transmitted. This will relieve contention on the intermediate nodes and improve channel utilisation as it reduces collisions. Researchers in [45] determined the contention on the channel based on how many times the MAC backed off transmission over a certain period of time, and if the contention was higher than a threshold they added one extra transmission time to the back off time in order to further delay transmission and enable other flows to transmit. On the other hand, researchers in [48] calculated the channel’s contention based on the round trip time of a TCP packet and adapted the TCP window size accordingly. So, if the channel is experiencing high contention the size of the congestion window is reduced so that the nodes capture the channel for a shorter period of time and allow other nodes to transmit. Both techniques aim to improve channel utilisation by introducing extra delays to reduce the effect of contention on the performance of TCP.

Consider the topology shown in figure 8 where the interference range is 550 meters as set in the network simulator ns2 and the transmission is 250 meters. The nodes are 200 meters apart. Node 0 can detect when node 2 captures the channel as node 1 would not return the CTS message if node 2 captured the channel. The hidden node problem occurs if node 3 is transmitting to node 4 and node 0 is transmitting to node 1 and the transmitted packets collide. Nodes 0 and 3 are hidden from each other as node 3 is three hops away from node 0 and outside it’s the interference range. The same applies to all nodes which are three hops apart along the same flow. In order to find a solution to the two problems, the intra flow contention and the hidden node problems, which have been discussed in section 2.8; which degrade the performance of TCP wireless networks significantly; introducing a four hop delay between any two successive packets is inevitable in order to relieve contention and allow the packets to be transmitted without collisions. This can be done either at the transport layer or the link layer. If introducing a delay at the link layer some cross layer information will be needed to prevent TCP packet from timing out and eventually dropped. Since the aim is to improve TCP performance in terms of channel utilisation, providing a solution at the transport layer is more feasible. The question that has to be answered is by how much should the packets be delayed? In the case of the chain topology with more than three hops, the optimal throughput that can be achieved is ¼ of the channel bandwidth as shown and discussed in [45], [48] and [54]. So, if the transmission of the second packet is delayed until the first packet has travelled four hops will not only alleviate the effect of packet collisions due to the hidden node problem, but also reduces the intra flow contention and still utilises the bandwidth effectively. Consider the following figure to illustrate what the scheme aims to achieve.

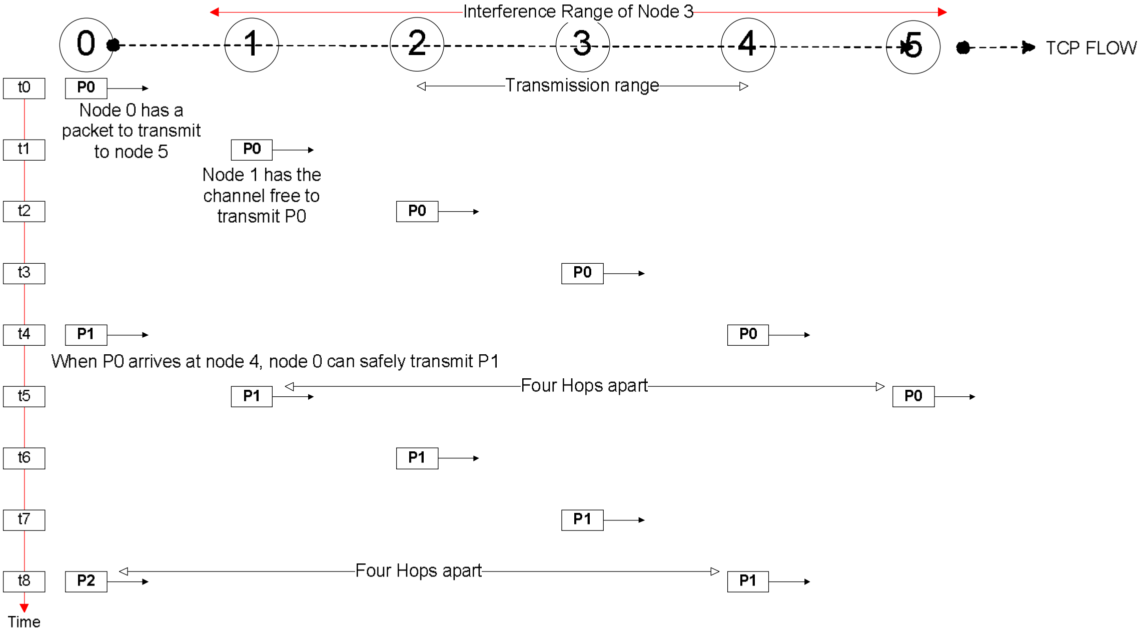

Interference Range of Node 3
0
1
2
3
4
5
TCP FLOW
Transmission range
t0
t1
t2
t3
t4
t5
t6
t7
t8
Time
P0
Node 0 has a packet to transmit to node 5
Node 1 has the channel free to transmit P0
P1
When P0 arrives at node 4, node 0 can safely transmit P1
Four Hops apart
P2
Four Hops apart

Figure 8: Illustration of the Four Hop Delay

In Figure 8, an illustration of an optimal packet pacing during transmission is shown. The packet is not transmitted unless the previous one has reached the fifth node. In addition, nodes within interference range not only do not compete with each other to capture the channel, but also packets will not be dropped due to hidden nodes because nodes 0 and 1 do not send the following packets until the previous packets have been transmitted four hops down the chain. In the previous figure, both nodes 3 and 4 are hidden from 0 and 1 respectively. Now, how can the four-hop delay be calculated? If our scheme is designed to calculate the four hop delay based completely on the round trip time as shown in equation (4).

$$FHD = 4 * \left(\frac{RoundTripTime}{2 * NumberOfHops} \right) = 2 * \frac{RoundTripTime}{NumberOfHops} \qquad (4)$$

This formula does not give a good estimation of the four hop delay because the round trip time includes the transmission time of the acknowledgement packets and we are not interested in delaying acknowledgements. In fact, it will reduce the throughput as TCP packets would time out and the packet gets retransmitted if assumed lost due to not receiving the acknowledgement as expected. However, trace files analysis showed that a better four hop delay estimation is achieved using equation (5); which is derived from equation presented in (3).

$$FHD = (2 + \alpha) * \left(\frac{RTT}{h} + \frac{SizeofPacket - SizeofAcknowledgement}{bandwidth} \right) \qquad (5)$$

In equation (5), RTT denotes the round trip time and h denotes the number of hops to destination. In order to determine the value α, which gives a better four hop delay estimation, a 10 hop chain has been designed and the nodes were spaced by 200 meters and the value α was varied from 0 to 0.9 and the simulations were run until 10000 packets were transmitted. The time it took to transmit the 10000 packet has been recorded and is shown in Table 1. The goodput which is the application layer throughput which does not include any retransmissions or packets overhead has been calculated based on the last 5000 packets transmitted and the first 5000 packets were discarded as initial transient. The reason for calculating the goodput rather than the throughput is simply to make the performance comparison between TCP-AP and FHDMAC feasible.

α :	**Goodput(Kbits/s):**	**Transmission Time Required (s):**
0.0	74.2879	1565.899
0.1	89.9461	1347.999
0.2	91.5083	1314.999
0.3	87.8194	1242.599
0.4	97.6559	1164.200
0.5	112.0992	1149.300
0.6	104.9302	1105.600
0.7	103.6633	1156.200
0.8	101.0566	1232.399
0.9	85.9598	1336.199

Table 1: Goodput variation and Transmission Time Required

Table 1 demonstrate that low alpha values lead to early transmission of subsequent packet and does not eliminate collisions due to hidden nodes and high alpha values lead to excessive delay and suboptimal performance.

Figure 9 shows the goodput obtained from the simulations versus α .

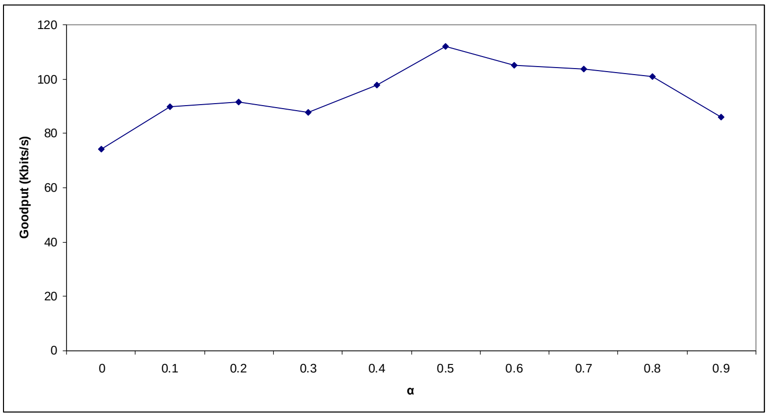


Figure 9: Goodput variation

From table 1 and figure 9, the highest goodput is achieved when alpha is equal to 0.5 which gives the closest estimation of the four hop delay. However, the smallest transmission time required is achieved when alpha equals 0.6 but since we are interested in the highest goodput alpha is chosen to be 0.5 when designing the Four Hop Delay Medium Access Control (FHDMAC) mechanism.

The design of the FHDMAC algorithm is presented in figure 10.

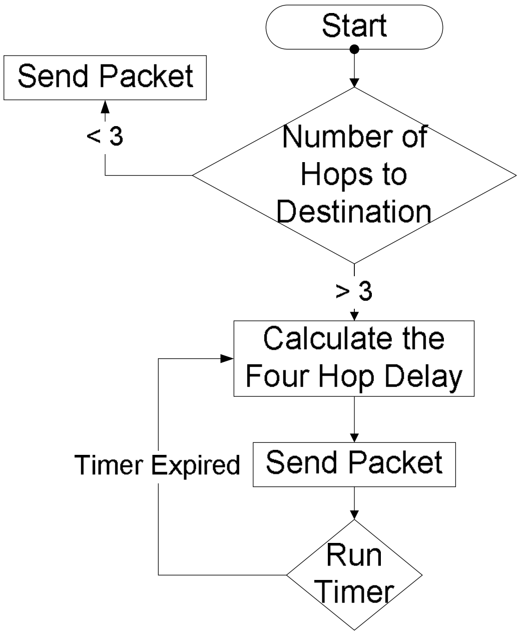


Figure 10: Four Hop Delay MAC Algorithm

The FHDMAC algorithm is different from that of TCP-AP presented in section 2.8.5 in that it preserves the way the congestion window grows as defined in New Reno and uses the most recent round trip time and calculates the four hop delay for each packet to be transmitted. Also, formula (5) gives a better estimation of the four hop delay and designed to achieve better goodput than TCP-AP and New Reno.

The proposed scheme “FHDMAC” improves TCP performance in flows consisting of more than 3 hops through introducing a four hop delay which is calculated for each packet to be transmitted and uses the most recent round trip time to pace TCP packet transmission. In the next two sections, the focus is on improving fairness in ad hoc networks while maintaining or improving the desired throughput and quality of service in the presence of multiple flows sharing the same channel resources. In order to improve fairness in ad hoc networks, various mechanisms will be adopted. It was established from the literature survey that ad hoc networks are unfair because of the way the 802.11 MAC protocol operates. The following sections present the proposed solutions to the unfair channel distribution exhibited by the MAC protocol in the identified topologies.

3.4 Contention Interference through Collision Monitoring

The MAC protocol implements a Distributed Coordination Function (DCF), which employs the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. Each node wishing to access the channel to transmit data starts by sensing the channel to determine whether it is free or another node is already transmitting. If the channel is free, the wireless node captures the channel and transmits data. If the channel is busy, the node backs off transmission by a value calculated using the binary exponential back off algorithm. On the other hand, the MAC protocol has no fairness mechanism to ensure fair access to the channel among nodes where the nodes interfere with each other but are outside the transmission range. A fairness mechanism is usually implemented in access points (APs) where the traffic is routed for all wireless nodes. However, in ad hoc networks, there are no APs, such a mechanism does not exist as the wireless nodes themselves act as traffic routers. If the wireless nodes are within interference range of each other, access to the channel is distributed fairly among nodes. This is because when a node requests access to the channel, it sends a request to send (RTS) frame and the intermediate node replies back with a clear to send (CTS) frame. The nodes which are in the same transmission range (250m) receive the RTS and CTS frames and defer their transmissions for a specified time duration set in the RTS and CTS frames. However, nodes outside the interference range (550m) of the current sending node cannot read the time specified in the RTS/CTS frames and therefore do not update their Network Allocation Vector (NAV). As a result, they may transmit while the initial node is still transmitting leading to packet collisions. When a collision happens, the MAC protocol exponentially increases the contention window which in turn leads to an exponential increase in the back off time. Therefore, the more collisions a node experiences; the less chances it has to capture the channel and transmit data and vice versa. This makes the MAC protocol to always favour the last transmitting node and leave the other nodes to starve.

The IEEE 802.11 MAC protocol is unfair in the presence of multiple flows in the same contention region. Every time a node experiences a collision, its contention window increases exponentially and each time a node successfully transmits data, the contention window decreases exponentially; therefore, the binary exponential back off scheme always favours the latest successful transmission and therefore causes unfair channel utilisation [32]. The nodes transmission is scheduled in the IEEE 802.11 MAC protocol using the Network Allocation Vector (NAV) [53]. According to NAV, if a node senses that the channel is busy by receiving an RTS or CTS that was sent to another node, it defers its transmission by a duration set in the RTS or CTS messages and updates its NAV to schedule the transmission [31]. However, if for example the RTS messages were sent by nodes within transmission range of each other, the nodes within interference range cannot schedule their transmissions and therefore collisions might occur and the duration by which the node should defer its transmission cannot be updated [53], [31]. Therefore, in such scenarios, scheduling the transmissions cannot be possible and packet collision can occur if the nodes transmit at the same time leading to unfair channel utilisation.

In order to improve fairness in scenarios where multiple flows exist, a novel algorithm called Fair Bandwidth Distribution MAC protocol (FBDMAC) is proposed. This algorithm is based on penalising the greedy nodes that cause unfair channel utilisation according to the number of collisions experienced to improve the overall fairness in the presence of multiple flows sharing the same channel resources. The FBDMAC protocol calculates a moving average of the number of MAC collisions for the DATA, ACK, RTS, CTS, AODV, and ARP packets transmitted by the node over a period of time individually. Depending on the types of collisions, the result determines whether the node is likely to be greedy or starving. If the node has high DATA or ACK collision rates, then it is occupying most of the bandwidth, over utilising the channel; if it has high RTS/CTS/AODV/ARP collisions, than it is struggling to access the channel. In terms of the implementation, the algorithm uses two thresholds $\beta_1$and β_2; if the node has a DATA or ACK collisions rate higher than β_1 then it is over utilising the channel and should be penalised. If the collision rates for RTS, CTS, AODV, and ARP packets experienced by the node are higher than β_2**,** then it is struggling to have access to the channel. The FBDMAC protocol algorithm reacts by cancelling the exponential back off algorithm for starving nodes and retransmits the collided packet to make the node more aggressive to gain access to the channel. On the other hand, the FBDMAC penalises the greedy nodes by starting their back off algorithm to force them to stop transmitting and allow the starving nodes to pick up transmission. Also, every time a new collision occurs, the moving average is recalculated in order to have up to date information about the state of the network in terms of MAC layer collisions. The moving averages for each packet type are then compared to $\beta_1$and β_2 to decide whether to penalise the nodes by doubling its contention window and backing off transmission or to cancel the back off algorithm and retransmit collided packets to permit the transmission of the starving nodes. The program diagram for the FBDMAC algorithm is further illustrated in figure 11.

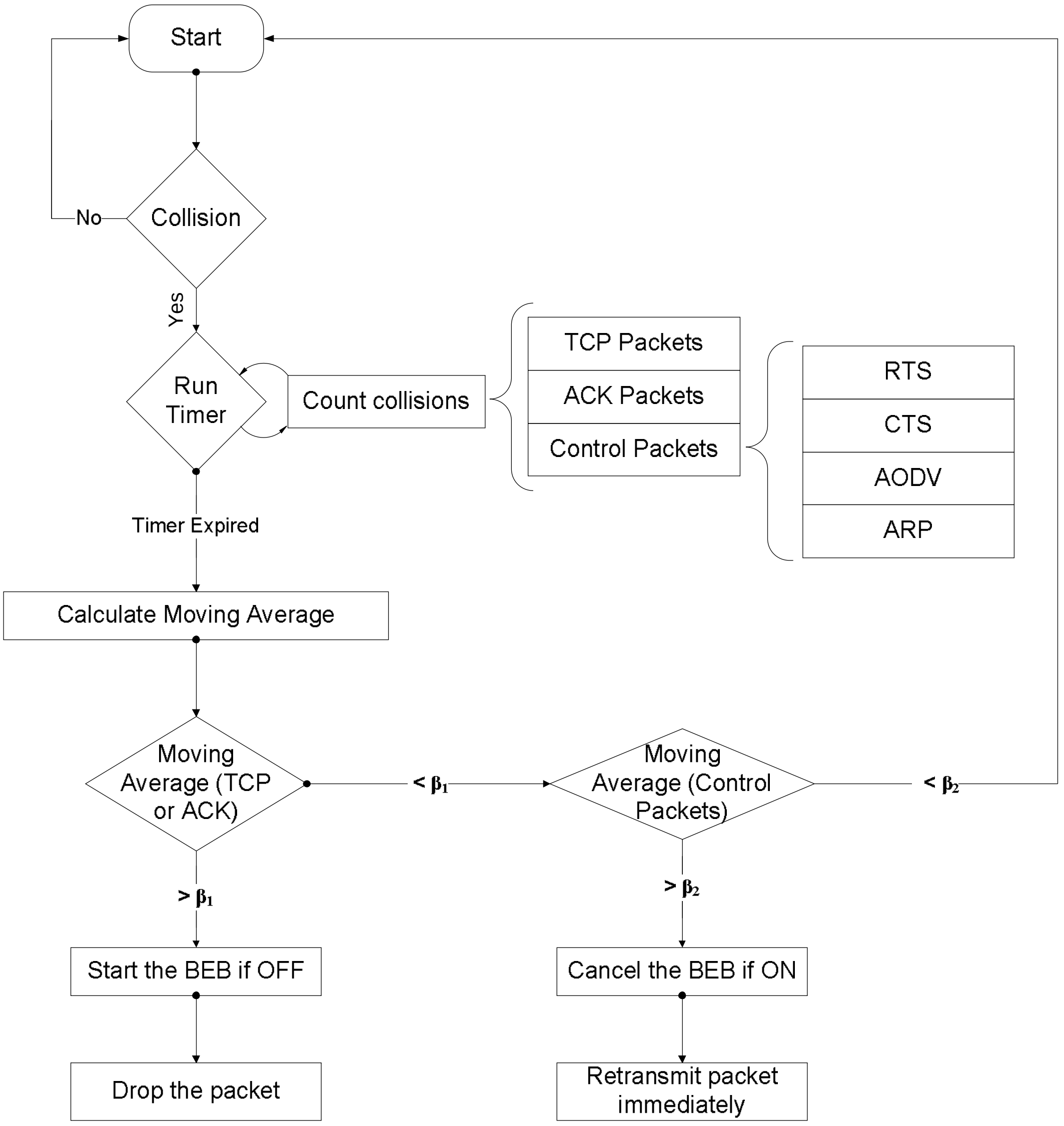


Figure 11: FBDMAC Algorithm Design

The optimal choice of β_1 and β_2 would ensure fairness among the different flows and is dependent on the number of nodes in the contention region and the number of hops to destination. The FBDMAC protocol algorithm would penalise greedy nodes and help nodes that are starving to gain control of the channel, eventually allowing the network to reach a balanced and fair share of the channel utilisation.

3.5 Transmission Policing in Partial Interference Scenarios

Ad hoc networks face great challenges due to the absence of an infrastructure to manage the network, an access to a single channel by multiple nodes, interference and the limited transmission range by the wireless media. Therefore, achieving fairness in ad hoc networks is a difficult task. In this section, two distinct novel algorithms are presented that tackle the unfairness of bandwidth distribution exhibited by the standard 802.11 MAC protocol among the competing nodes in the presence of multiple flows. These algorithms are named Neighbouring Transmission Rate Control (NTRC) and Transmission Rate Control through Acknowledgements Feedback (TRCAF). The Neighbouring Transmission Rate Control (NTRC) algorithm, is based on monitoring the transmission speed of the neighbouring nodes, and aims to optimise network fairness through identifying and helping starving nodes to increase their transmission rates. The Transmission Rate Control through Acknowledgements Feedback (TRCAF) tackles the unfair channel resources distribution through feedback included in MAC layer acknowledgements about the state of the network. This information helps the intermediate nodes on whether to carry on transmitting or halt transmission in order to help the starving nodes to gain access to the channel.

3.5.1 Neighbouring Transmission Rate Control Algorithm

The Neighbouring Transmission Rate Control (NTRC) algorithm tackles the unfair distribution of channel resources in ad hoc networks between competing flows by penalising greedy nodes in order to allow starving nodes to transmit and ensuring that the channel resources are distributed fairly among the competing flows in the same network. The NTRC algorithm is implemented at the MAC layer and allows each node to identify and discover its neighbouring nodes and topology. Every time the node hears a packet not destined for itself, the source address of the packet is extracted and stored in a dynamic array. When the node has a packet to send, the destination address is compared to the addresses stored in the dynamic array. If the address is found in the dynamic array, it is then removed as it is the next hop address to destination. The next hop address is removed from the array in order not to include it in the decision making on whether to stop or carry on with transmission. The dynamic array is then left with addresses of only competing nodes. It is important to differentiate between next hop nodes and competing nodes. The next hop nodes are excluded from the competing nodes array because the NTRC algorithm counts the packets of the nodes that it competes with not the intermediate hops. If the next hop node was included in the decision making, then the intermediate node to destination would always stop transmitting as it assumes that the destination node is starving because the destination node only sends packets back to its intermediate node and which are not counted by the NTRC algorithm. The process of adding nodes to the dynamic array is continuous, as nodes could be mobile and the topology of the network could change at any time and if new nodes are within transmission range and they send packets, they get added to the array. Therefore, the NTRC algorithm takes into consideration the mobility of the nodes in a Mobile Ad Hoc Network and the constant route change. In addition, each time a packet is received a counter of the number of packets heard from the neighbouring nodes is incremented to indicate if they are greedy or starving. Also, nodes that produce no traffic are not assumed to be starving as a node is added to the neighbouring nodes array only when it sends a packet to its next hop, in which case it is competing for the channel resources if it is within the same transmission range to other nodes. The NTRC algorithm also implements a moving average, which re-evaluates the nodes’ behaviour periodically through the use of a timer named waiting time. This moving average is then compared to a threshold to decide whether the node should carry on transmitting or should halt to allow the neighbouring nodes to transmit. The algorithm was designed with the assumption that all sending nodes have a packet ready to be transmitted at all time.

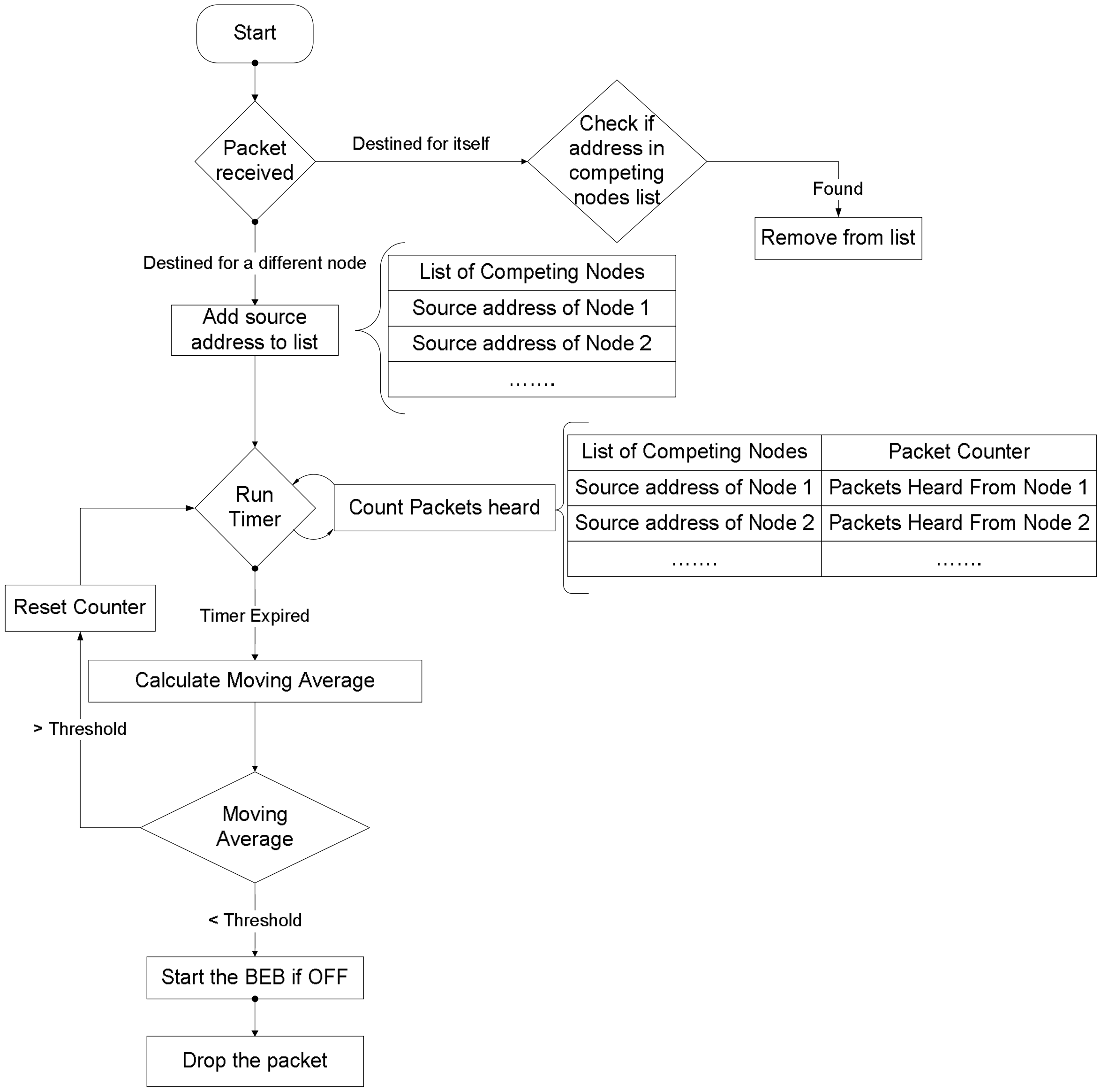
The NTRC algorithm is designed to be efficient and improves the networks fairness dramatically by penalising the greedy nodes through backing off transmission when they detect that their competing node’s transmission is below a threshold which allows the starving nodes to transmit in fair manner.

The NTRC algorithm uses a moving average consisting of the current new average and the previous average value as per equation (6)

$$\mathrm{avg}(i+1) \quad = \quad \mathrm{avg}(i) * \alpha \quad + \quad {\mathrm{count}(i+1)}\big/_{\mathrm{waitingTime}} * \beta \qquad (6)$$

In equation (6), count determines the number of packets transmitted by the competing node over a period of time named waiting time in the equation.

Figure 12 describes the operation of the NTRC algorithm. If a packet is received from a node within transmission range, the destination address is extracted to determine whether the packet was intended for the node that received it. If it is intended for the node that received it, the source address is compared to the list of competing nodes addresses. If it is, then the address is removed as the node that sent the packet is an intermediate hop and should not be competing with it. On the other hand, if the packet is intended for a different node, the source address is extracted and stored in the list of competing nodes. Every time a packet is heard from the nodes stored in the competing nodes list, a packet counter is incremented. Upon the expiry of the waiting time duration, the moving average is calculated and if the moving average for any node in the competing nodes list is below the threshold, then the node is starving and the back off algorithm is started to prevent the greedy node from sending any more packets and allow the starving node to transmit data.



List of Competing Nodes
Source address of Node 1
Source address of Node 2
…….

List of Competing Nodes	Packet Counter
Source address of Node 1	Packets Heard From Node 1
Source address of Node 2	Packets Heard From Node 2
…….	…….

Figure 12: NTRC Algorithm Design

3.5.2 Transmission Rate Control through Acknowledgements Feedback

Ad hoc networks suffer degradation in quality of service in terms of fairness in the presence of multiple flows in the same interference region. As previously described, the lack of fair channel distribution among competing flows is due to the random exponential back off algorithm used by the 802.11 Medium Access Control (MAC) protocol. This section introduces a novel algorithm named Transmission Rate Control through Acknowledgements Feedback.

Figure 13 presents an identified topology for which the standard MAC protocol has no mechanism to ensure fair bandwidth allocation. The topology consists of nine nodes forming three flows competing with each other over a single shared channel at the receiving nodes (n_6; n_7; n_8) which are two hops away from the sending nodes (n_0; n_1; n_2). The source nodes are 574 meters apart. Thus, outside the interference range of each other; this is 550 meters as set in the network simulator ns2. On the other hand, the intermediate nodes (n_3; n_4; n_5) are 387 meters apart which makes them within the interference range but outside the transmission range which is 250 meters according to ns2. Finally, the destination nodes (n_6; n_7; n_8) are 200 meters away from each other.

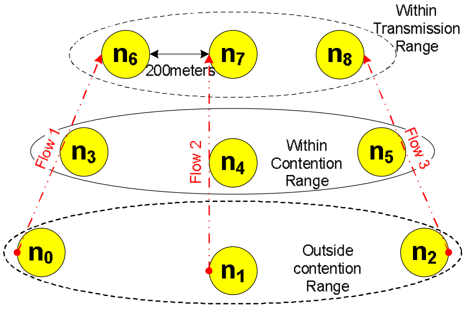


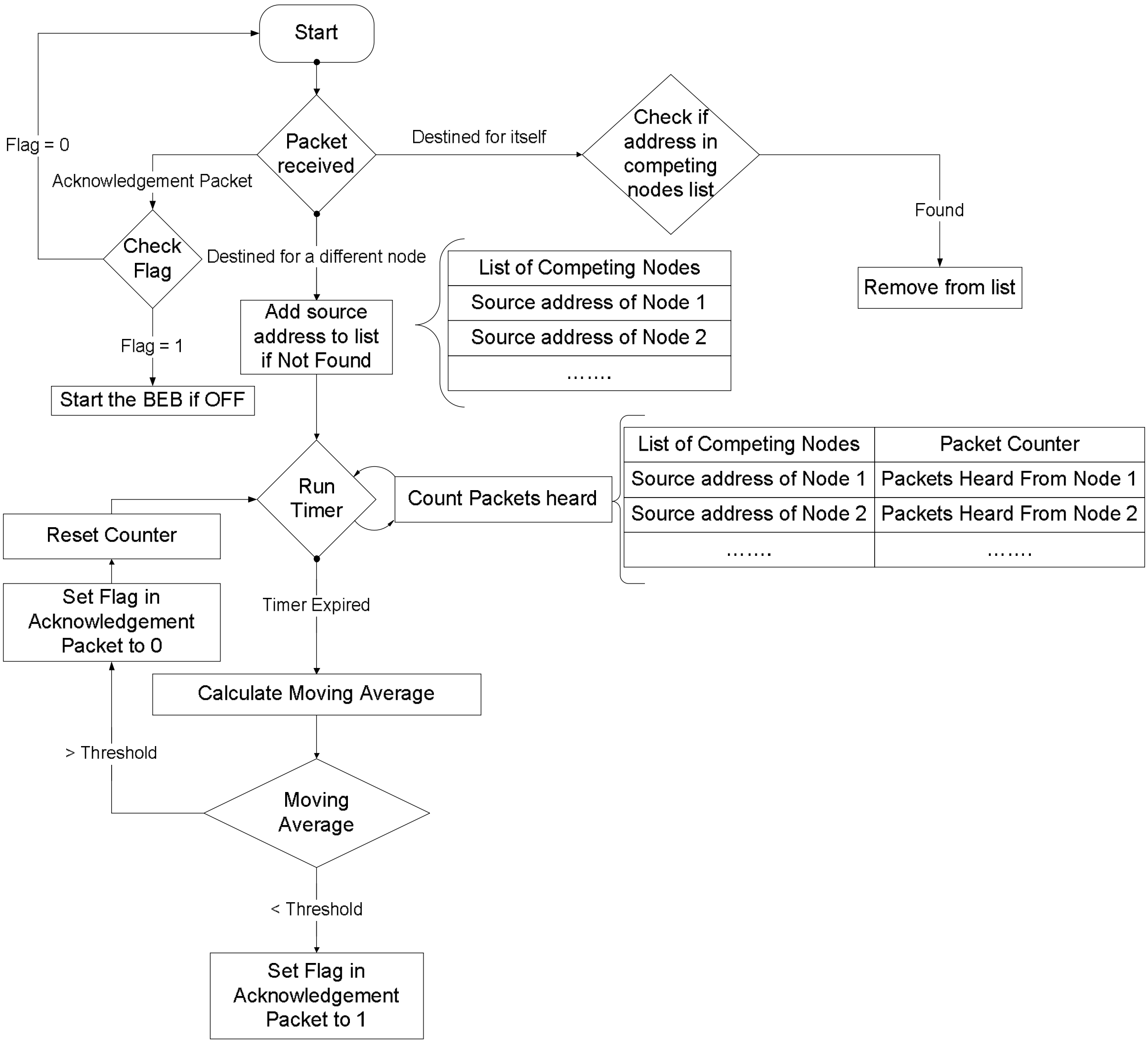
Figure 13: Senders outside interference range

Based on the topology design, if (n_0; n_1; n_2) have data to transmit to their intermediate nodes (n_3; n_4; n_5) respectively, the channel is always idle and the nodes proceed with the transmission leading to collisions at their intermediate nodes. The intermediate nodes would not be able to update their Network Allocation Vector as the time required for the successful transmission is not available as the RTS/ CTS messages cannot be read by the intermediate nodes as they are outside transmission range. In this case, the nodes that suffered collisions would randomly choose a back off time and exponentially increase it every time a collision occurs. On the other hand, the nodes that did not experience collisions would always have higher chances of accessing the channel than the nodes that backed off their transmissions. This leads to severe unfairness among the competing flows as the standard MAC does not have any mechanism that tackles scenarios like the one presented in figure 13.

In order to alleviate the unfair bandwidth distribution among the competing flows the Transmission Rate Control through Acknowledgements Feedback (TRCAF) algorithm is proposed. The Algorithm is implemented as an enhancement to the MAC protocol. Firstly, each time a node hears a transmitted packet that is not destined for it, the source address is extracted and compared to the addresses of its intermediate hops. If the address is not in the list, then it is of a competing node. The competing node's address is stored in an array and every time a packet is heard from that node a counter is incremented and reset upon the expiry of a time interval. According to the topology shown in figure 13, node n_6 would hear the acknowledgement packets sent by nodes n_7 and n_3 but node n_3 is its intermediate node so the address of node n_7 is stored by n_6 and the counter is incremented every time a packet is heard by n_6 from n_7. On the other hand, node n_7 would be monitoring the acknowledgement packets sent by nodes n_6 and n_8 and would have two counters for each node in its list. Secondly, a moving average for each node in the list based on the number of packets counted is calculated every time the interval time expires as per equation (7).

$$avg(i+1) = avg(i)*\alpha + {}^{count(i+1)}\!\!\diagup_{waitingTime} * \beta \qquad (7)$$

Finally, if the moving average is below a threshold, then the competing node that the moving average corresponds to is starving and an acknowledgement is sent with a flag in its header to indicate contention. Upon reception of the acknowledgement packet by the intermediate node, it halts transmission and backs off in order to allow the starving nodes to capture the channel and transmit packets.



List of Competing Nodes
Source address of Node 1
Source address of Node 2
…….

List of Competing Nodes	Packet Counter
Source address of Node 1	Packets Heard From Node 1
Source address of Node 2	Packets Heard From Node 2
…….	…….

Figure 14: TRCAF Algorithm Design

Figure 14 presents a state machine that illustrates the operation of the TRCAF algorithm. Upon reception of a TCP packet, the destination address is extracted and compared to the node’s address to check if it is destined for itself. If it is, then the algorithm ensures that the address is not saved in the competing nodes list to prevent competing with its intermediate hops. Furthermore, if the packet is intended for a different node, then the source address is extracted and stored in the competing node’s list. A counter for each node in the list is incremented every time a packet is received and upon the expiry of the waiting time duration; the moving average is calculated for each node in the list to determine whether any of the competing nodes is starving. If a starving node is identified due to its moving average being below a threshold, then a flag is set in the MAC acknowledgement packet which propagates through the chain to inform the intermediate nodes that the sender is greedy. Upon reception of an acknowledgement with the flag, the intermediate nodes cancel their transmissions and back off transmission to alleviate contention and allow the starving nodes to transmit packets.

3.6 Conclusion

In this chapter, the experiments showed that the throughput that New Reno obtains in wireless networks is reduced by 94% compared to the wired networks. This shows the dimension of the reduction in channel utilisation in wireless networks. In addition, pacing the transmission of packets by introducing a four hop delay between any two successive packets significantly reduces packet loss due to wireless contention and hidden nodes problem in a chain topology consisting of more than three hops long. TCP-AP significantly improves the overall throughput than New Reno by delaying packets transmission by four hops. However, a better estimation of the four hop delay than that of TCP-AP has been introduces to further improve the throughput and goodput in wireless networks where there are four hops or more. Furthermore, fairness in ad hoc networks needs to be addressed and three novel mechanisms to improve fairness in identified scenarios where the standard MAC protocol fails to achieve fair bandwidth allocation have been proposed, in order to improve the channel utilisation while maintaining fair bandwidth allocation among the competing node in both intra and inter flow transmission, and in the presence of multiple flows sharing the same channel resources. In the next chapter, the four mechanisms are evaluated to determine their performance in terms of channel utilisation and fair bandwidth allocation. The FHDMAC algorithm performance is benchmarked against TCP-AP and New Reno. The other three mechanisms “FBDMAC, NTRC and TRCAF” performances are evaluated in comparison to the performance of the standard MAC 802.11 protocol.

Chapter 4. **Evaluation of the Proposed Algorithms**

4.1 Introduction

The task of evaluating novel algorithms in ad hoc networks in comparison to prior solutions by the research community is quite challenging. The challenge augments due to the inaccessibility to the implemented solutions and frameworks and the different environments in which the solutions are provided. Therefore, the proposed frameworks presented in the previous chapter had to be evaluated against standard technologies such as 802.11 MAC protocol except FHDMAC which is evaluated versus the performance of TCP-AP and TCP New Reno. In order to fulfil the task of evaluating the proposed mechanisms, the choice of a simulation based approach is preferred over an implementation based approach as it has not only been the choice of most researchers but also due to the support and flexibility that a simulator offers in terms of node positioning and parameter setting in various scenarios and environments and the ability to idealise the physical conditions in order to accurately evaluate the improvement due to the proposed algorithms.

The following sub-sections describe the simulation experiments carried out in order to test and evaluate the effectiveness and performance of the proposed algorithms in terms of alleviating the effects of hidden nodes, packet loss and distributing the available bandwidth fairly among the competing nodes and flows.

4.2 Simulation Environment

The necessity for a simulation environment is crucial to the successful deployment and testing of novel techniques within the networking society in real scenarios. Although, there is a concern in the research community over the reliability of simulation tools [58] in case of wrong data analysis, as an example, would lead to false results. In this work that would not be the case, as no node mobility is taken into consideration as the nodes in all simulations are static and there is no complex or deep level data analysis required which would lead to misleading results. A number of simulation environments have been developed and thoroughly tested to imitate what would occur in real scenarios. These simulators include and are not limited to: GloMoSim [59] designed for large and scalable network simulation, OPNET [60] which provides the user with graphical interface and used to simulate application and network performance, NetSim [61] which is a commercial network simulator designed for network lab simulation, OMNeT++ which is a simulation framework used to provide the platform for implementing simulations and the network simulator ns2 [62]. Ns2 is a discrete event network simulator developed and aimed for the research community. Ns2 provides support for simulation and development of new and or existing protocols such as TCP, routing and ad hoc networks for both wired and wireless simulations. In this study, the network simulator version 2.34 was used for all simulations carried out to monitor the performance and evaluation of the proposed algorithms. The choice of ns2 lies behind the fact that it is an open source system widely used by the research community and for results comparison purposes. Furthermore, ns2 provides a detailed infrastructure for developing new protocols and applications. It also offers the opportunity to examine and evaluate a large number of protocols and extensions in a managed environment.

4.3 Evaluating the Four Hop Delay Medium Access Control (FHDMAC)

In this section, the Four Hop Delay Medium Access Control algorithm is evaluated in order to determine its weaknesses and strengths in comparison to TCP New Reno and the proposed TCP-AP. The FHDMAC algorithm is implemented in ns2 and the pseudo code is presented as follows:

```
/* Terminology
numPKT = number of packets to be transmitted
numHops = number of hops to destination
RTT = round trip time
FHD = four hop delay
*/
if numPKT > 1
do
   if numHops > 3
       foreach packet
           calculate RTT
           calculate FHD
           Send packet
           timer = FHD
           if (timer expires)
               send packet++
   else
       send packet
```

The FHDMAC algorithm is triggered when the number of hops to destination exceeds three. The round trip time and the four hop delay are calculated every time a packet is awaiting transmission in order to stay tuned with the state of the network. Upon the expiry of the four hop delay duration, the successive packet is transmitted to ensure that no packets contend with each other over the shared medium and eliminate the effects of hidden nodes on the performance of TCP.

4.3.1 Comparison of Four Hop Delay Medium Access Control and New Reno

The FHDMAC algorithm was implemented in ns2 and an inter-packet delay was introduced in order to schedule the transmissions of packets, reduce the effect of intra flow contention and avoid packet loss due to the hidden nodes. It is important to emphasise that the four hop delay is only used when the TCP flow chains consist of more than 3 hops. A comparison between the performance of the Four Hop Delay MAC algorithm and New Reno is presented in figure 15. The topologies consist of a single flow and the number of hops to destination was increased in each simulation. In all simulations, 10000 packets are transmitted, the first 5000 packets are discarded as initial transient in order to achieve consistent comparison and the goodput is calculated based on the remaining 5000 packets. The goodput obtained for both the FBDMAC mechanism and New Reno is plotted in the following graph.

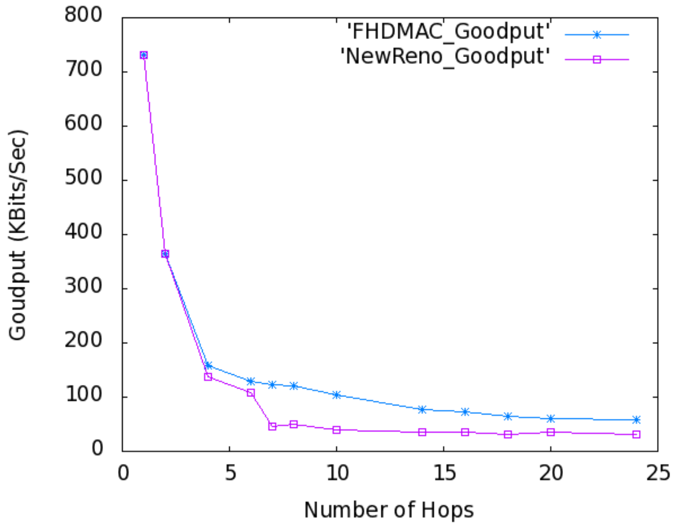


Figure 15: Four Hop Delay Medium Access Control VS New Reno

In the graph presented in figure 15, the FBDMAC scheme achieves the same goodput as New Reno when the distance between the source and destination is three hops or less. If the number of hops is four or more delaying the transmission of the following packets until the previous packet has travelled four hops away achieves better goodput than New Reno as the results in table 2 demonstrate.

Number of hops	FHDMAC (Kbits/s)	TCP New Reno (Kbits/s)	Improvement (%)
1	731.4984	731.4984	0
2	364.2354	364.2354	0
4	157.6784	138.0238	12
6	130.1441	107.6895	17
7	122.597	45.85067	63
8	121.5624	50.15028	59
10	103.2505	38.9593	62
14	76.63612	34.43961	55
16	72.77045	34.82898	52
18	65.51269	31.76965	52
20	60.18394	34.87608	42
24	57.42495	30.58203	47

Table 2: FHDMAC performance VS New Reno

FHDMAC mechanism improves the channel utilisation in comparison to New Reno by at least 12% when the chain consists of four hops and by 63% in seven hops chains. New Reno struggles as the number of hops increases. The goodput that the FHDMAC mechanism achieves also reduces as the chain gets longer but still achieves more than 50% increase goodput than New Reno. The goodput is expected to reduce further if the flow consists of more 24 hops. However, the FHDMAC mechanism would still outperform New Reno in longer chains.

4.3.2 Comparison of Four Hop Delay Medium Access Control and TCP-AP

Similar simulations have been carried as in the previous section and the goodput for both our scheme and TCP-AP [48] has been calculated and a plot showing the performance of the two schemes is presented in the following figure.

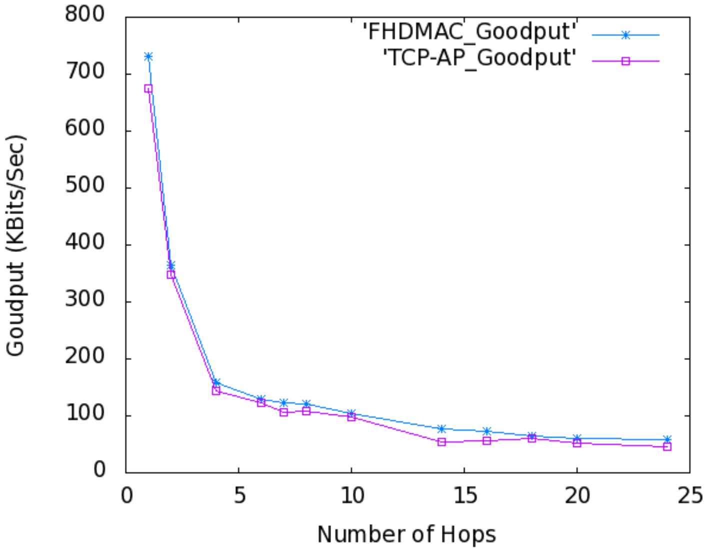


Figure 16: Enhanced Four Hop Delay VS TCP-AP

In figure 16, the FHDMAC scheme achieves marginally higher goodput than TCP-AP for all chain topologies experimented.

Number of hops	FHDMAC (Kbits/s)	TCP-AP (Kbits/s)	Improvement (%)
1	731.4984	675.4146	8
2	364.2354	347.414	5
4	157.6784	142.8158	9
6	130.1441	122.2445	6
7	122.597	107.2091	13
8	121.5624	108.0627	11
10	103.2505	96.96604	6
14	76.63612	54.36173	29

16	72.77045	56.15168	23
18	65.51269	60.65642	7
20	60.18394	51.99202	14
24	57.42495	45.95169	20

Table 3: FHDMAC performance VS TCP-AP

The results in table 3 illustrate that TCP-AP does not take into account flows consisting of less than 4 hops in contrast to the FHDMAC algorithm. In addition, the FHDMAC mechanism provides improvements in terms of channel utilisation of 5% to 23% over TCP-AP.

In figure 17, a comparison between the congestion window sizes (cwnd) in New Reno, TCP-AP and FHDMAC is presented. The simulations have been run on a 10 hop chain for 600 seconds and the congestion window for each scheme has been plotted. It is important to note that the simulation performed in this study range between 450 and 1000 seconds. The lengths of the simulations have been chosen to ensure that the proposed methods are robust and support long transmission durations. Furthermore, long simulations provide precise information on the state of the network used in the evaluation process.

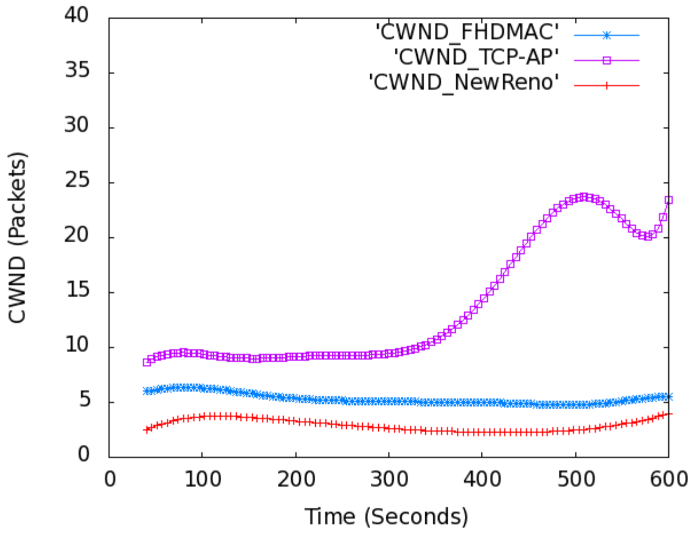


Figure 17: CWND for NewReno, TCP-AP and enhanced FHD

In figure 17, the congestion window in TCP-AP grows very large. This is because in TCP-AP any losses that occur are assumed to be wireless losses rather than buffer overflow. The reason behind this assumption is because there is a delay introduced between the transmissions of any two consecutive packets, it is rare to have buffer overflow. Therefore, the congestion window is not reduced to half as in New Reno when a loss occurs. The congestion window for the proposed FHDMAC scheme is larger than New Reno because there are less packet losses in the proposed FHD algorithm when compared to New Reno. This is because there is less contention, less buffer overflow and most importantly less hidden node collisions.

4.4 Fairness in Non Symmetric Flows

The aim behind designing the topology presented in figure 18 is to create high contention between nodes 12 and 16 in the topology to monitor how the standard MAC protocol distributes the channel resources among the three flows. The first flow runs from node 0 to node 5, the second flow from node 3 to node 4 and the third flow from node 2 to node 1 as show in the figure below. The source nodes of the three flows cannot sense the transmissions of each other as they are outside the interference range.

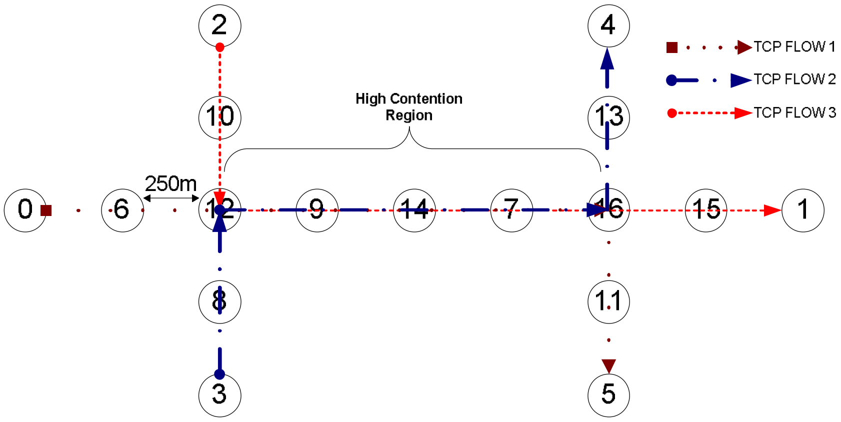


Figure 18: Three crossing Flows

In figure 18, all the packets generated by the source nodes of the three flows have to pass through nodes 12, 9, 14, 7 and 16, where high contention is experienced, in order to reach their destinations. The following figures show the throughput of each flow for New Reno, TCP-AP and the Four Hop Delay MAC mechanism.

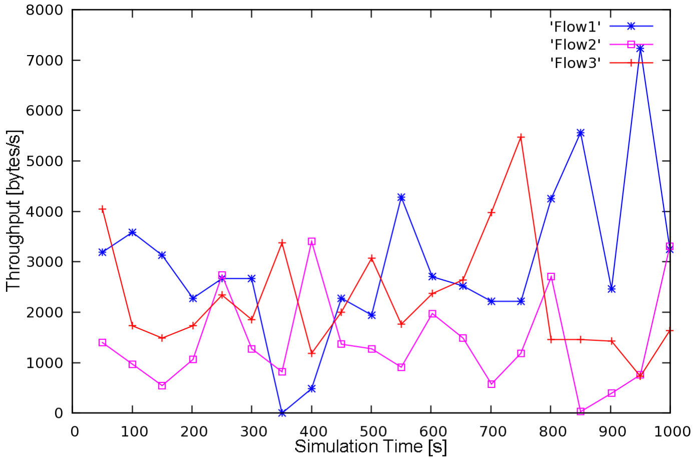


Figure 19: Three Flows New Reno

In figure 19, the average throughput of all flows for New Reno is low in comparison to TCP-AP and FHD shown in figures 20 and 21 respectively. This is because the performance of New Reno is significantly penalised by the intra flow contention and inter-flow contention. Inter-flow contention is where two flows passing in the same interference range compete with each other for the channel [54]. In addition, flow 1 achieves the highest throughput amongst the other two flows. As shown in table 3, flow 1 achieves 31 Kbits/s and flow 2 and 3 achieve 11 Kbits/s and 9 Kbits/s, which is 65% and 71% goodput reduction respectively.

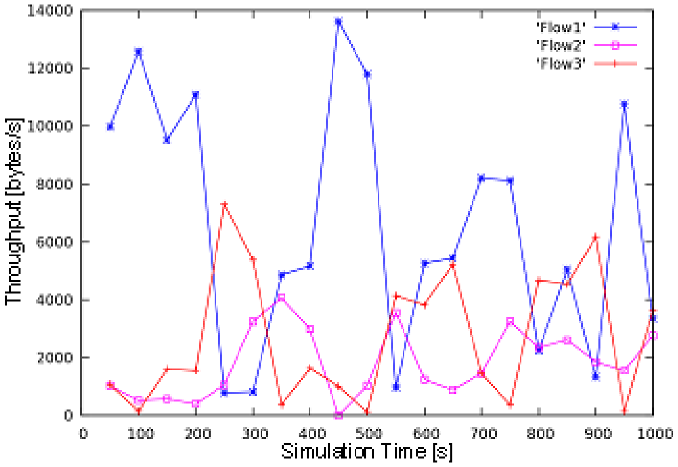


Figure 20: Three Flows TCP-AP

Figure 20 shows the throughput that TCP-AP achieves for the three flows. Flow 1 is again dominating the other two flows in the first 200 seconds of the simulation and the goodput obtained for flow 1 is 61 Kbits/s which is higher than what New Reno obtained. However, flows 2 and 3 obtained 8 Kbits/s and 6 Kbits/s respectively. These results demonstrate that flow 1 dominates access to the channel and leave flows 2 and 3 to starve. This unfair bandwidth distribution is improved through the proposed fairness mechanisms namely FBDMAC, NTRC, and TRCAF evaluated later in this chapter.

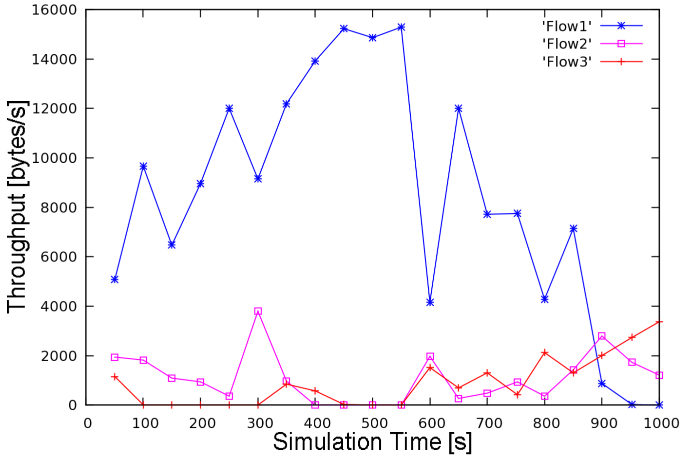


Figure 21: Three Flows FHDMAC

Figure 21 presents the performance of the proposed FHDMAC scheme and yet again flow 1 is dominant with 97 Kbits/s of goodput which is higher than New Reno and TCP-AP but on the other hand flows 2 and 3 obtained only 4 Kbits/s and 4 Kbits/s which is lower than New Reno and TCP-AP. Again, this unfair bandwidth distribution is due to the fact that flow 1 accessed the channel first and prevented the other two flows from accessing it.

Tables 4 below summarises the goodput obtained for the three flows topology presented in figure 18 for each TCP variant, the simulation was run until each flow successfully transmitted 4000 packets, the first 2000 packets where discarded as initial transient and the goodput was calculated based on the remaining 2000 packets.

TCP Variant:	Flows:	Goodput (Kbits/s):
New Reno:	Flow1:	30.862
	Flow2:	10.840
	Flow3:	8.795
TCP-AP:	Flow1:	60.788
	Flow2:	7.503
	Flow3:	6.078
FHDMAC:	Flow1:	96.921
	Flow2:	3.938
	Flow3:	3.891

Table 4: Goodput of NewReno, TCP-AP and FHDMAC in the Three Non Symmetric Flows Topology

After running these simulations, not only the three TCP variants suffer from inter-flow contention but also there is no mechanism that ensures fairness among the flows. As shown in the previous figures, flow 1 dominated the other two flows. The reason for this to happen is the inefficiency of the MAC protocol. The 802.11 MAC protocol implements a Distributed Coordination Function (DCF) that tries to avoid transmission collisions. As mentioned before, prior to a transmission, the MAC sends a RTS message and, if it does not receive a CTS message, it will exponentially back off its retransmission time. If this back off time is large and the channel becomes free before this back off time expires a different node might capture the channel and the previous one will have to back off again. According to the IEEE MAC 802.11 standard [35] the exponential back off time is calculated as follows:

$$\text{Back off time} = \text{Random}\,(0,\ \text{CW} - 1) * \text{SlotTime}$$
$$\text{where : CW is Contention Window, SlotTime} = 20\ \mu\text{s}$$

In case of a collision, the contention window (CW) is increased exponentially as follow:

$$CWnew = min(2 * CWold + 1, CWmax);$$

for this reason the back off time can be large and the larger the back off time is the less chances the node would have in capturing the channel [39].

4.5 Fairness in Symmetric Flows

In figure 22 below, there are three symmetric chains. Each chain consists of 6 hops. There are 400 meters between any two chains. So, flows 1, 2 and 3 are within interference range of each other but outside the transmission range of each other and will not be able to schedule their transmissions as they cannot update their Network Allocation Vector with the transmission time needed which is declared in both the RTS and CTS or in other words collisions will occur.

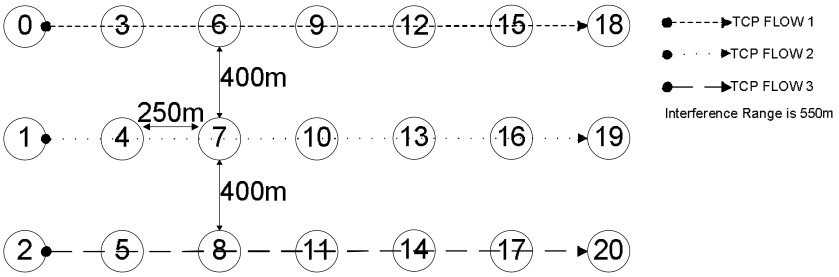


Figure 22: Three Symmetric Flows Topology

In figures 23, 24 and 25, simulations results for the throughput that the three flows obtain for the topology presented in figure 22 for New Reno, TCP-AP and FHD respectively. In all simulations, flow 2 was achieving almost no throughput at all, because it was not able to capture the channel as its transmission was interfered by the other two flows and collisions were constantly happening. Therefore, the back off time was constantly reaching its maximum threshold. The other two flows were reaching reasonable throughput in the following order: FHDMAC aggregate throughput for the two flows was the highest and TCP-AP in second place followed by New Reno in third place.

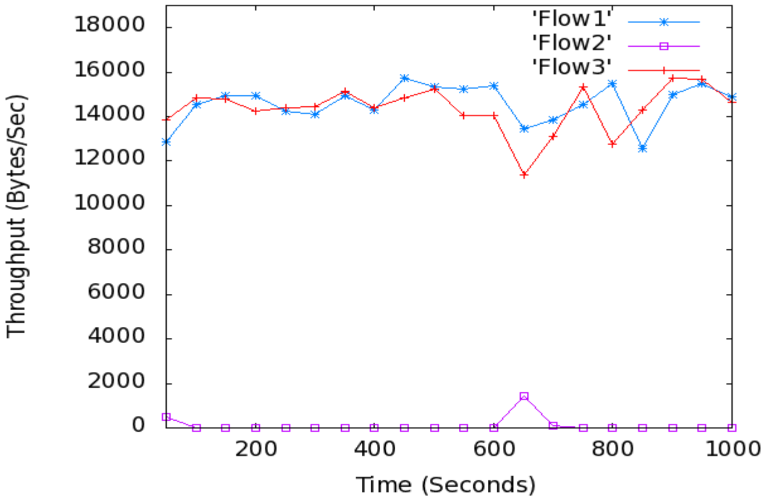


Figure 23: New Reno

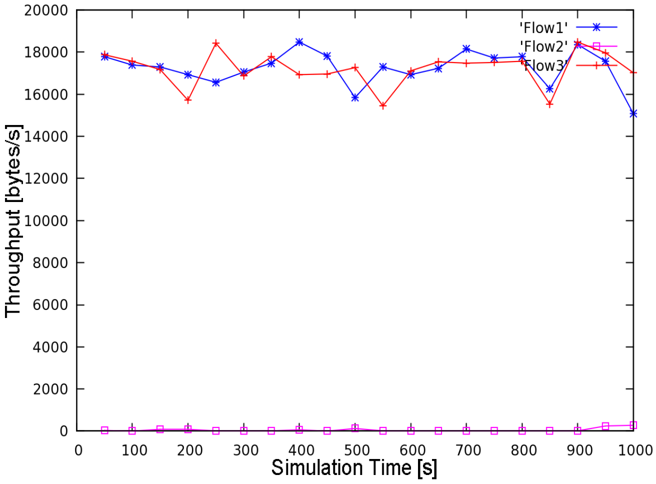


Figure 24: TCP-AP

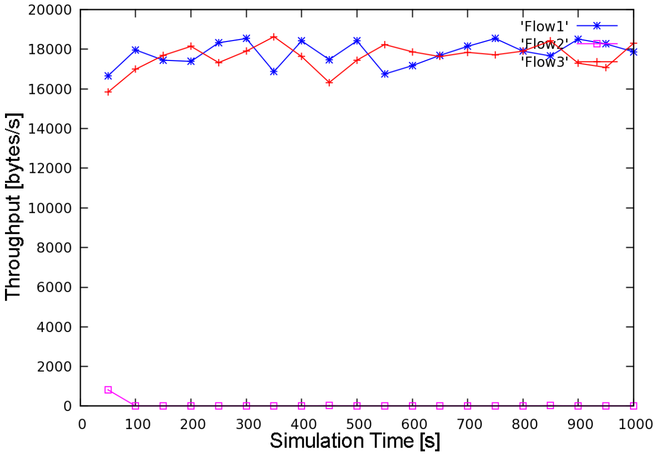


Figure 25: FHDMAC Scheme

4.6 Fairness in a one hop scenario

In order to determine the reasons for such unfairness among flows in wireless networks as shown in sections 4.4 and 4.5, the following simulations have been carried out in ns2. The reason for this simplification is to eliminate the multi hop traffic, any routing issues and see how the MAC protocol behaves in terms of fairness and in the presence of contention and interference. All simulations have been run for 1000 seconds and there are four nodes 200 meters apart and two flows in each simulation both starts at the same time. These topologies are illustrated in the following figures.

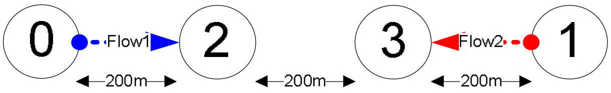


Figure 26: Single hop Fairness 1

In figure 26, the source nodes are outside the interference range of each other and the receiving nodes are within transmission range of each other.

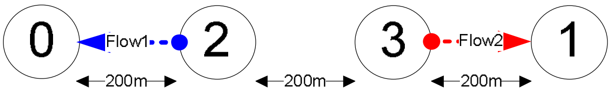


Figure 27: Single hop Fairness 2

In figure 27, the source nodes are within transmission range of each other and the receiving nodes are outside the interference range of each other.

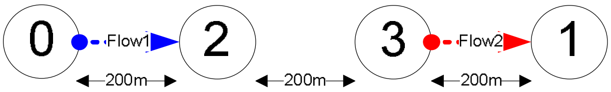


Figure 28: Single hop Fairness 3

In figure 28, the source nodes are within interference range of each other but outside their transmission range and the same apply to the receiving nodes.

The following table shows that TCP in flow 1 achieves 1.827 Kbits/s which is extremely unfair compared to TCP in flow 2, which achieves 91.611 Kbits/s of goodput for the simulation shown in figure 26. Flow 2 dominates access to the channel over flow 1 and TCP in flow 1 starves for the duration of the simulation. In addition, in figure 27, there is a 2 to 1 ratio between the two TCP flows. In figure 28, the performance of TCP is similar to that in figure 26 where flow 2 dominates flow 1.

	Flow:	Goodput (Kbits/s)
Figure 26	Flow 1:	1.827
	Flow 2:	91.611
Figure 27	Flow 1:	33.057
	Flow 2:	60.159
Figure 28	Flow 1:	5.683
	Flow 2:	87.838

Table 5: Unfairness Illustration

In order to find the reasons behind such unfairness among TCP flows as shown in table 5; the trace files for each simulation have been analysed.

In figure 26, the source nodes are outside the interference range so the RTS message can not be heard nor detected which will not allow them to synchronise their transmissions, and if they both transmit at the same time a collision would happen. The receiving nodes are both within transmission range of each other so they can both hear the CTS message if they do send one.

Trace file analysis demonstrates that node 0 sends a broadcast routing message to find a route to node 2. Each time a collision happens, the retransmission is backed off by 2 seconds then by 4 seconds then by 5.5 seconds. After the fourth collision, the MAC layer at the source node “0” reports a link breakage, all packets in the queue get dropped and a route discovery is initiated. This happened because node 1 captured the channel first by sending the broadcast message first at 1.0003 whereas Node 0 sent the broadcast message at 1.0006 second and the Contention Window (CW) at the link layer grows exponentially after every collision which means node 0 will not try to access the channel for longer periods. On the other hand, node 1 has not experienced any collisions and its CW will be very small and hence will have greater chances in capturing the channel.

The other remark from analysing the trace file is that the link layer reports a link failure even though it is not and the routing layer initiates a route discovery process, which does not help in such a scenario because the route is there but other greedy nodes would not release it.

Node 2 did not successfully receive the broadcast message for the route until 400.003 seconds of simulation time. Node 0 managed to capture the channel and send the frame to node 2 because a packet collision happened at node 3 at 400.002 seconds. The collision happened because node 0 happens to send a frame to node 2 at 400.0028 seconds just before node 1 sent the frame to node 3 at 400.0029 seconds.

Apart from the simulation shown in figure 27, where the throughput achieved by flow 1 was 50% lower than the throughput achieved by flow 2, because the source nodes for the two TCP flows were within transmission range of each other hence they were able to hear the RTS messages they sent, but they could not hear the CTS messages the receiving nodes sent. As a result of this, they were able to detect when the channel was busy but could not schedule their transmissions effectively. On the other hand, in both simulations shown in figures 26 and 27, there was extreme unfairness where one TCP flow was completely shut down by the second flow. Analysis of the trace files revealed that most TCP packet drops were due to collisions because it is not possible for the MAC layer to schedule frames transmissions in such topologies. The RTS frame is very small compared to the data packet in which case if a collision happens the RTS frame gets dropped which in turn means that the node requesting for the channel will not win over a node that already captured the channel. The no route found (NRTE) message is generated a node tries to contact its intermediate node but with no success. A time out message is generated (TOUT) if the TCP packets timed out in the interface queue (IFQ) waiting to be forwarded and gets dropped after the timer expires and finally the large back off time. Trace file analysis revealed that the back off time can reach 48 seconds before the node tries to capture the channel again.

The next two sections present a performance evaluation of the three techniques described in sections 3.4 and 3.5, which deal with the unfair channel resources distribution while maintaining the desired throughput.

4.7 Fair Bandwidth Distribution MAC Algorithm (FBDMAC) Evaluation

The FBDMAC algorithm presented in the previous chapter has been implemented in the network simulator ns2 and tested on identified topologies in which the standard MAC protocol does not distribute the channel resources fairly among the flows within the same interference range. The pseudo code of the FBDMAC algorithm is presented as follows:

```
/* Terminology
PktType = DATA || ACK || AODV || RTS || CTS
Δt = 1 second
cw = Contention Window
BK = Back Off Algorithm
α = 100
 */
while(simulation)
do
avg[PktType]=0
if(collision)
   collisions[PktType]=0
  timer=0
  while(timer <= Δt)
           collisions[PktType] ++
end if
avg[PktType]=(avg[PktType]+(collisions[PktType]/Δt)*α)/(α+1)
if (avg[(DATA||ACK)]>β1)
```

```
  // Node is greedy
  //Penalise – increase cwnd, start back off algorithm
  Drop(packet)
  cw=cw*2
  BK.start(cw)
end if
else if (avg[(RTS||CTS||AODV||ARP)]>β2)
  //Node is starving
  //Reward – cancel exponential back off algorithm
  if(BK.on)
      BK.stop
  else
      send (packet)
  end if
end else if
done
```

Upon experiencing a collision, a timer is started and the number of collisions experienced for each packet type is counted. The moving average is calculated for each packet during a time interval. If the moving average for data packet is high then the node is greedy and if the moving average for control packet is higher then the threshold then the node is starving. The greedy nodes are penalised by dropping their packets and starting the back off algorithm. On the other hand, the back off algorithm for the starving nodes gets cancelled to allow them to access the channel. The thresholds $\beta_1$and β_2 have been set to 1.0 and 0.2 respectively and these values have been chosen based on monitoring the collision rate for each packet type during the simulation of the standard MAC protocol.

4.7.1 Symmetric topology evaluation

In figure 29, there are three pairs of communicating nodes: (n_0; n_2), (n_1; n_3), and (n_4; n_5), where nodes n_0, n_1, n_2 are the senders and n_3, n_4, n_5 are receivers respectively. The distance between the senders and the receivers is 200 meters and the distance between the pairs is 400 meters. The positioning of the nodes is important hence the choices of 200 and 400 meters, because the transmission range is set to be 250 meters and the carrier sense range is set to be 550 meters in ns2. Therefore, the two pairs (n_0; n_3) and (n_2; n_5) do not interfere with each others transmission. However, the middle pair (n_1; n_4) suffers interference from both pairs (n_0; n_3) and (n_2; n_5) and competes for the channel resources with both of them.

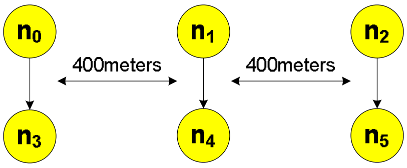


Figure 29: Three symmetric pairs

The simulation has been run for 500 seconds, the flows start at the same time and the results for the standard 802.11 MAC protocol are shown in the graph presented in figure 30 for comparison purposes with the performance of the FBDMAC shown in figure 31. The throughput that the two pairs (n_0; n_3) and (n_2; n_5) achieved is very high *186086 and 185943 B/s* and almost identical, which is expected since the two flows do not interfere nor compete for the channel resources. However, the throughput that the middle pair (n_1; n_4) achieves is extremely low and only *467 B/s*. This is because pair (n_1; n_4) competes for the channel resources with both other pairs as the pair (n_1; n_4) is in the contention region of both pairs (n_0; n_3) and (n_2; n_5) but gets penalised due to the behaviour of the standard MAC in such conditions to favour the last successful transmission.

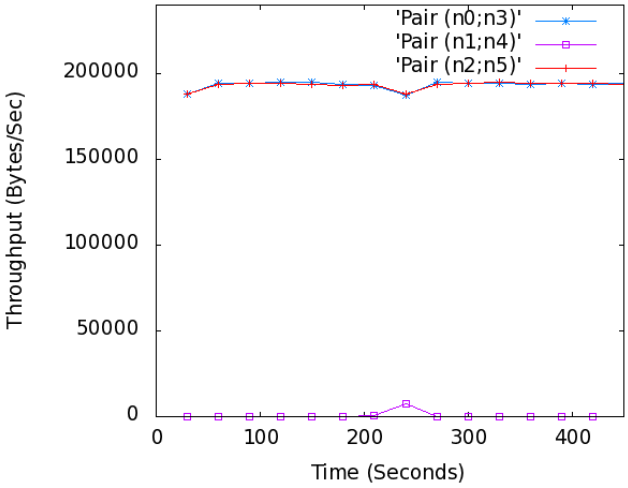


Figure 30: Performance of high contended flows using standard MAC

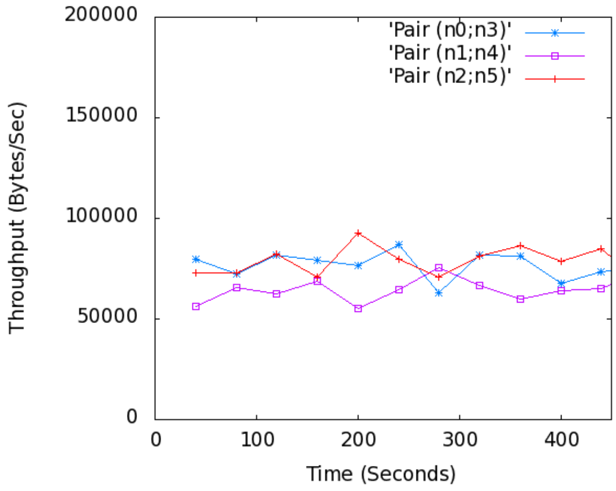


Figure 31: Performance of FBDMAC algorithm

In the case of FBDMAC algorithm, the pair (n_1; n_4) does compete with the other two pairs and achieves very satisfying throughput of *62019 B/s*. Furthermore, pairs (n_0; n_3) and (n_2; n_5) achieve *73237* and *75477 B/s* respectively. In addition, by comparing figures 30 and 31, the channel bandwidth is well distributed by all the different flows fairly. However, the overall channel utilisation drops by 43% in case of FBDMAC at the expense of fair bandwidth distribution in comparison with the standard 802.11 MAC protocol. However, if the fact that flows (n_0; n_3) and (n_2; n_5) were using two channels simultaneously into account and therefore not sharing the single channel as imposed by FBDMAC algorithm, then the FBDMAC does not actually reduce the channel utilisation.

There are a number of fairness measurements available such as Jain's fairness index, max-min fairness and fairly shared spectrum efficiency. In this thesis, the fairness of the proposed algorithm was evaluated using Jain's fairness equation in which *n* is the number of flows and *x* is the throughput that each flow achieves. Also, *1/n* is worst case fairness and *1* is perfect or total fairness [63]. In addition, Jain's fairness index is robust, simple and provides accurate details about the fairness of the system. Jain' fairness equation is presented as follows:

$$\text{Fairness} = {(\Sigma\, x_i)^2} \Big/ {\left(n \cdot \Sigma\, x_i{}^2\right)}$$

The average throughput achieved by each of the three flows in case of standard MAC protocol is: *(x_1; x_2; x_3) = (186086; 467; 185943) B/s*. Applying Jain’s equation, the fairness achieved by the standard 802.11 protocol is 0.66. This value shows that the standard 802.11 MAC protocol provide poor fairness, as the worst case for our topology is 0.33 (whereby one flow would have full control and the other two would not be able to transmit at all) and perfect fairness is 1 (if the three flows would equally share the available bandwidth).

When applying the proposed algorithm for changing the MAC protocol on the same simulation scenario, the average throughput achieved by each of the three flows is: *(x_1; x_2; x_3) = (73237; 62019; 75477) B/s*. Applying Jain’s equation using these, the networks fairness is 0.99. This value shows that the FBDMAC protocol does indeed provide almost total fairness among the different flows.

From the graphs above and by applying Jain’s equation it is clear that the FBDMAC protocol achieves with 99% fair bandwidth distribution, which is a great fairness improvement from that of the standard MAC protocol.

The FBDMAC protocol is designed to be adaptable to different network topologies. In the next subsection, the FBDMAC protocol will be tested on a different scenario to prove that it does adapt and improve the network fairness when compared to the standard MAC protocol.

4.7.2 Non symmetric topology evaluation

In figure 32 below, nodes n_0, n_1 are senders and n_2, n_3 are receivers respectively. If n_0 and n_1 start transmitting at the same time and if n_0 captures the channel first, the packets sent by n_1 would be dropped due to collisions with n_0 packets. The contention window for n_1 would increase exponentially and also n_0 and n_1 would not be able to schedule their transmissions as they cannot update their NAV tables because they cannot read nor understand the RTS and CTS messages exchanged as n_2 and n_3 are outside the interference range of each other. In addition, as mentioned earlier, the MAC 802.11 favours the latest successful node as its contention window would always be small, making it even harder for n_1 to capture and utilise the channel.

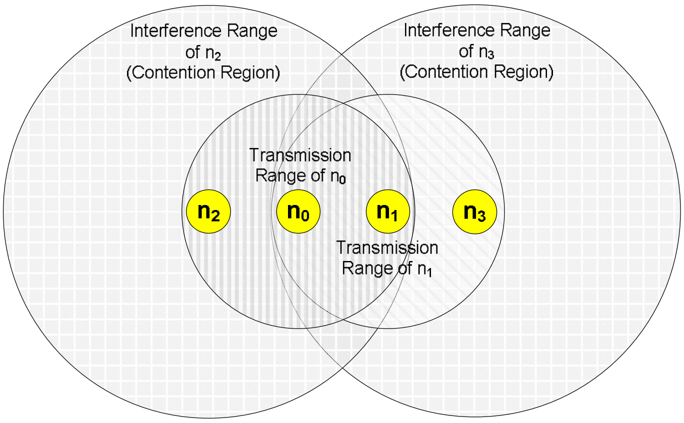


Figure 32: Two pairs Topology

The simulation ran for 500 seconds, the nodes were positioned 200 meters apart and the results for the standard 802.11 protocol show that the pairs (n_0;n_2) and (n_1;n_3) achieve 75091 B/s and 107062 B/s throughput respectively; whereas the simulation results for the FBDMAC protocol show that the pairs (n_0;n_2) and (n_1;n_3) achieve 81300 B/s and 83751 B/s throughput respectively. Applying Jain's fairness index equation, the standard MAC protocol achieves 97.01% fairness and on the other hand, the FBDMAC protocol achieves 99.99% fairness. This shows that the FBDMAC algorithm provides a marginal improvement to the fair distribution of bandwidth.

Testing the FBDMAC protocol under the two different scenarios, as shown in figures 29 and 32, has shown that it does provide fair access to the channel and equal bandwidth share between the competing flows. However, the overall network throughput has dropped by 43% for the scenario shown in figure 29 and by about 9% for the scenario shown in figure 32. This is due to factors such as penalising the greedy nodes to allow starving nodes to have access to the channel and to computation overhead. The 802.11 MAC protocol works well in scenarios where the flows are in the same interference range of each other. However, if the flows are outside the transmission range but within carrier sense range the MAC 802.11 would not be able to ensure fair bandwidth distribution among the competing flows and the unfairness problems arise. The FBDMAC protocol version has been shown to greatly improve fairness in such scenarios. In addition, good choices of the thresholds β_1 and $\beta_2$further improve the bandwidth distribution depending on the network's topology.

4.8 Neighborhood Transmission Rate Control (NTRC) Evaluation

The NTRC algorithm is designed to tackle the unfair bandwidth distribution exhibited by the standard MAC protocol in scenarios where the sources nodes are within transmission range. The NTRC algorithm has been implemented in the mac-802.11 class of the ns2 simulator and the pseudo code is presented as follows.

```
while(simulation)
do
if(receive packet not destined for itself)
   extract source address from packet header
  //source address is my neighbour address or my next hop
address
  add source address to myArray
end if
if(send packet)
  get packet destination address
  if(destination address == myArray[i])
  // the address is my next hop address so remove from myArray
  // myArray must not include next hop address so that it does
 // not affect on decision making
  Remove next hop address from myArray if found
  end if
end if
  while(timer <= waiting time)
      if(relay packet)
          myArray[i].counter++
      end if
  end while
update moving_average
if (moving_average[i] < threshold)
  // Node is greedy
  //Penalise –start back off algorithm to allow its neighbours
to transmit
  Backoff.start(Contention Window)
end if
done
```

First the node differentiates between its intermediate competing nodes. Then number of packets that the competing nodes have transmitted is counted over waiting time duration. Then, the moving averages are calculated for each node in the competing nodes’ list. If the moving average of any of the nodes in the list is below the threshold, then the node is deemed to be starving. In this case, the NTRC algorithm obliges the transmitting node to stop transmitting by starting its back off algorithm to permit the starving nodes to transmit data. Initially, the parameters were set to α = 0.9 and β = 0.1 but following from the preliminary tests, α was set to 0.5 and β set to 0.5 to avoid sharp changes to the moving average. The waiting time was set to 0.2 seconds, the threshold was set to 1 and the contention window was set to 744 following from experimental results.

In order to demonstrate the performance of the NTRC algorithm a number of simulations have been run. A radial fixed structure to cover a circular area has been chosen where the performance of the standard MAC 802.11 protocol is extremely unfair. This scenario is presented in figure 33 below.

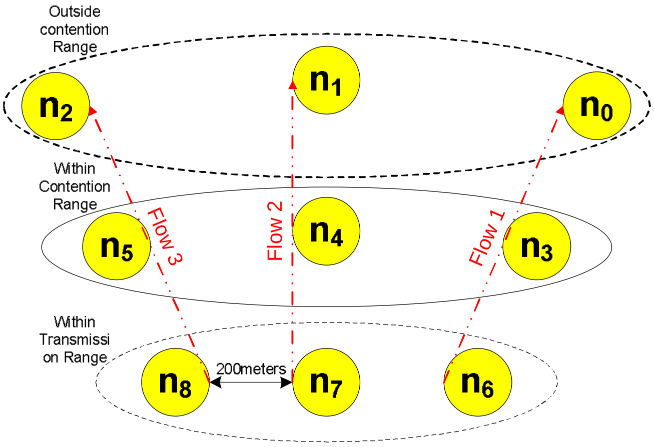


Figure 33: Fixed structure to cover a circular area

Figure 33 presents three flows competing against each other for the wireless channel. Each flow consists of two hops from source to destination as follows: nodes (n_6, n_3 and n_0) make up flow 1; (n_7, n_4 and n_1) construct flow 2 and flow 3 consists of nodes (n_8, n_5 and n_2). In ns2, the transmission range is configured to be 250 meters and the interference or contention range is set to be 550 meters. Taking this into consideration, nodes n_6, n_7 and n_8 are 200 meters apart hence within transmission range, nodes n_3, n_4 and n_5 are 387 meters apart so outside transmission range of each other but within interference range and finally, nodes n_0, n_1 and n_2 are separated by 574 meters which makes them outside the interference range of each other.

In figure 34 below, the throughput that each flow achieves is plotted in order to show how unfair the 802.11 protocol behaves in such a scenario. The simulation was run for 450 seconds. The pair (n_7; n_1) dominates access to the channel for the duration of the simulation. All transmissions were started at the same time and node n_7 gained access to the channel first because when its packet collided with the packets of the neighbouring nodes n_6 and n_8, it backed off for a shorter period than nodes n_6 and n_8 which allowed node n_7 to request access to the channel while nodes n_6 and n_8 were still waiting for the back off time to expire. The other two nodes n_6 and n_8 deferred their transmissions based on the information included in the RTS message which node n_7 sent to its next hop. However, when the packet reached its intermediate node n_4; nodes n_3 and n_5 were unable to schedule their transmissions as they are outside the transmission range of each other. Therefore, they could not schedule when to request access to the channel which led to collisions. For this reason, nodes n_6 and n_8 backed off and kept exponentially increasing their contention window which meant their probability to access the channel reduced too.

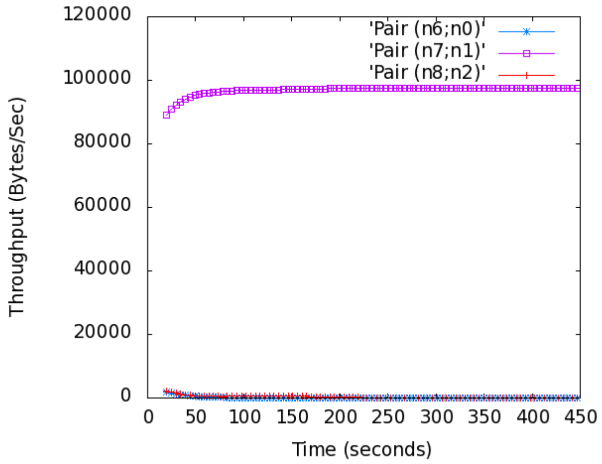


Figure 34: Standard MAC 802.11 performance

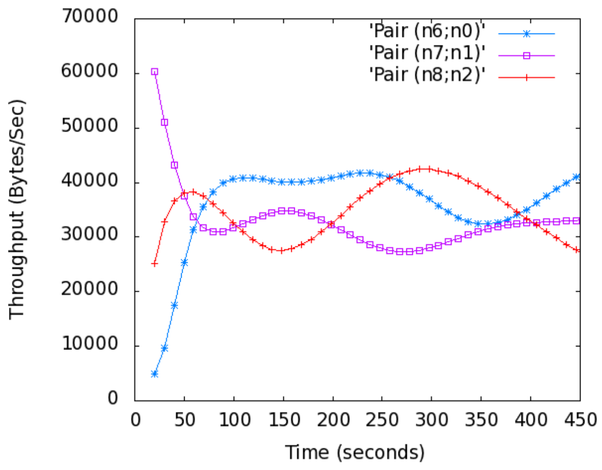


Figure 35: The performance of NTRC Algorithm in terms of Fairness

Figure 35 above demonstrates the performance of the NTRC algorithm in terms of throughput and channel utilisation. The three competing flows do share the bandwidth available to them fairly. At the beginning of the simulation, the nodes in the network are not aware of their competing nodes or neighbours. However, after about 10 seconds of simulation time, the competing nodes were detected and the starving nodes where identified and start to channel resources started to be shared fairly among the competing nodes. The transmission speed of pair (n_7;n_1) starts from about 60000 B/s and is gradually reduced to about 32000 B/s. The transmission speed of pair (n_6; n_0) at the beginning of the simulation is about 5000 B/s and gradually increased to about 40000 B/s. Pair (n_8;n_2) does increase its transmission speed from about 25000 B/s to about 37000 B/s. The graph shows that it took approximately 30 seconds to evaluate each other's transmission speed, after which the nodes share the bandwidth fairly between them as required by the NTRC algorithm.

In order to evaluate the fairness of the NTRC algorithm and the standard MAC protocol, Jain's fairness equation as illustrated in the previous section was used to scale the NTRC fairness in comparison to the MAC protocol.

The aggregate throughput that each flow achieves with the standard MAC protocol is *(144.32; 97011.16; 288.32) B/s* for pair (n_6; n_0), pair (n_7; n_1) and pair (n_8; n_2) respectively. Applying Jain's equation to these values, the resulting fairness index is *0.336*, which shows that the MAC 802.11 is unfair. On the other hand, the aggregate throughput that pair (n_6;n_0), pair (n_7;n_1) and pair (n_8;n_2) achieves with the NTRC algorithm is *(38289.32; 31806.16; 34041.32) B/s* respectively and applying Jain's equation to these values the outcome is 0.994 which shows that the NTRC algorithm does achieve fairness between the competing nodes.

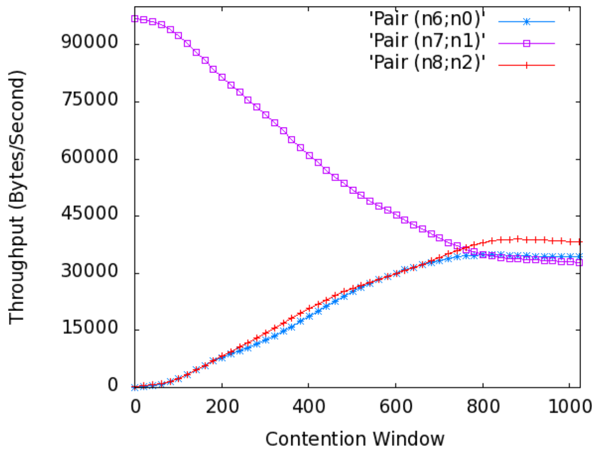


Figure 36: The impact of CW on Fairness

Figure 36 shows the throughput that each flow achieves for each value of the contention window. It illustrates that the three competing flows achieve approximately the same transmission speed of about 37000 B/s when the contention window size is greater than 700 and up to 1024.

Figure 37 below shows a 3-dimensional plot of Jain's fairness index versus threshold values and contention window values; in order to determine the impact of threshold and contention window on the performance of the NTRC algorithm in terms of fair bandwidth distribution. The results were obtained by setting the waiting time value to 0.2 and varying the contention window and the threshold values.

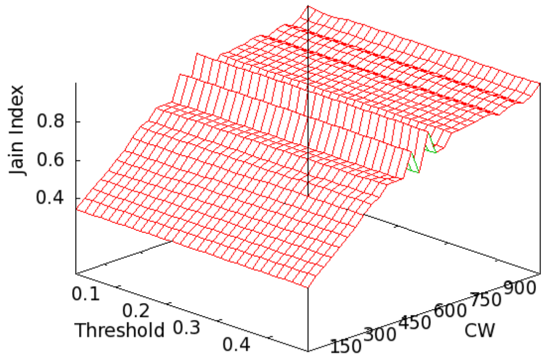


Figure 37: The impact of CW and Threshold on Fairness

The experiments show that varying the threshold values does not have an impact on fairness because the channel can only be accessed and utilised by one node at a time. However, varying the contention window does influence fairness. The network was fair when the contention window was set to in the range 700 to 1024. At these values, the fairness index is very close to 1.

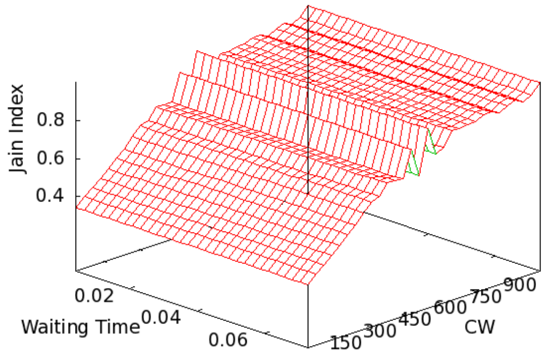


Figure 38: The impact of CW and Waiting Time on Fairness

In figure 38, the same observations are made as with figure 37. The results were obtained through fixing the threshold value to 1 and varying the contention window and waiting time values in the NTRC algorithm. Then, Jain's Fairness equation was applied to the aggregate throughput achieved from each flow and the results were plotted as shown in figure 38 above. Again, varying the waiting time values did not have an impact on fairness. The fairness index is very close to 1 when the contention window is chosen in the range 700 to the maximum value 1024. The fact that the fairness increases continuously up to the maximum value of the contention window indicates that there is room for improvement. In addition, the overall throughput does decrease by about 5% when the contention window was set to 1024 in comparison to contention window size 744.

4.9 Transmission Rate Control through Acknowledgement Feedback (TRCAF)

The Transmission Rate Control through Acknowledgement Feedback has been implemented in ns2 as an enhancement to the functionality of the standard MAC protocol. It is designed to improve fairness in topologies where the receivers are within transmission range of each other and the source nodes are outside the interference range. The pseudo code for the TRCAF algorithm is presented as follows:

```
while(simulation)
 do
  if(receive packet not destined for itself)
    extract source address from packet header
 //source address is my competing node’s address or
    //my next hop address
 add source address to myArray
  end if
  if(send packet)
    get packet destination address
   if(destination address == myArray[i])
     //the address is my next hop address so remove
        //from myArray
     //myArray must not include next hop address so
        //that it does not affect on decision making
     Remove next hop address from myArray
     end if
  end if
  while(timer <= waiting time)
   if(relay packet)
      myArray[i].counter++
     end if
  end while
  update moving_average
  if (moving_average[i] < threshold)
 // Competing node is starving
 set flag in the Acknowledgement packet to 1
  else
 set flag in the Acknowledgement packet to 0
  end if else
  if (receive Acknowledgement)
  if(Flag == 1)
 Backoff.start(Contention Window)
   end if
done
```

Similarly to the NTRC algorithm, the TRCAF algorithm differentiates between intermediate and competing nodes through comparing the destination address and the addresses stored in the competing nodes list. The transmitted packets by the competing nodes are counted and the moving average is calculated upon the expiry of the waiting time set to 0.2 seconds. Initially, the parameters were set to α = β = 0.5 but following from the preliminary tests, α is set to 0.7 and β is set to 0.3 to smooth the moving average. Furthermore, if the moving average is lower than the threshold which is set to 1, then the node that the moving average corresponds to is starving and an acknowledgement is sent with a flag set to 1 to inform the intermediate nodes that the sender is greedy and to start the back off algorithm to defer transmissions by a contention window value set to 831 in order to allow the starving nodes to pick transmission and share the available bandwidth fairly with competing nodes.

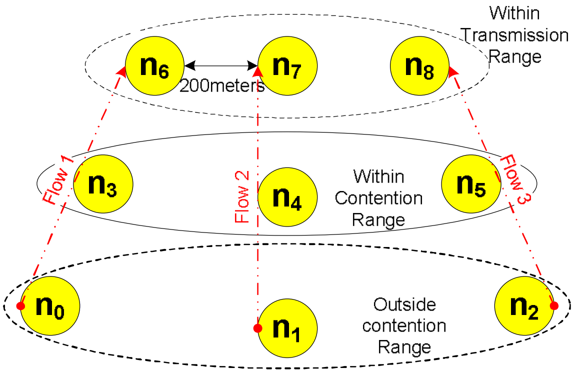


Figure 39: A half radial topology

A radial fixed structure to cover half a circular area has been selected as shown in figure 39 where the MAC 802.11 protocol is extremely unfair. The results of the simulation are shown in the plots of figure 40 below.

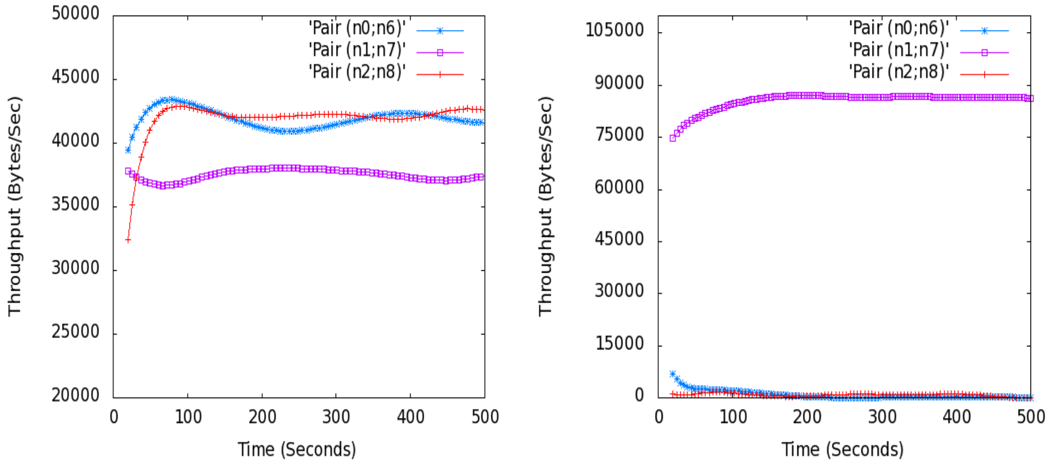


Figure 40: Performance comparison between TRCAF (left) and standard MAC protocol (right) for the half radial topology

According to the plot on the right of figure 40, the standard MAC protocol is extremely unfair as the pair (n_1; n_7) dominates access to the channel and maintain a transmission speed of 85656 B/s and leave the other two pairs to completely starve and only achieve 834 B/s even though the three flows started transmission at the same time. The situation does not change for the duration of the simulation, which is 500 seconds. The source nodes transmitted their packets assuming that the channel is free as they were outside the contention range of each other. When the packets reached the intermediate nodes, collisions occurred and the intermediate node could not synchronise their transmissions, as they were outside the transmission range of each other and randomly backed off. Pair (n_1; n_7) did not experience collisions; hence it did not back off transmission and gained access to the channel. The other two pairs (n_0; n_6) and (n_2; n_8) were unable to access the channel as whenever their intermediate nodes have packets to transmit they find that pair (n_1; n_7) is occupying the channel.

However, the TRCAF shown on the left of figure 40 fairly distributes the channel resources among the three competing flows. As the destination nodes (n_6; n_7; n_8) are within transmission range. According to the TRCAF algorithm whenever a node notices that its competing nodes are starving, it sends a MAC acknowledgement with a flag to indicate contention and inform the intermediate nodes to back off transmission and allow the starving nodes to pick up transmission. The pairs (n_0; n_6) and (n_2; n_8) achieve 42 KB/s throughput and the pair (n_1; n_7) achieve 38 KB/s which is about 10% less than the other two nodes due to node n_7 having two nodes to compete with and therefore having to back off in order to allow the other two nodes to increase their transmission.

Simulations have been run in order to determine the optimal value of the contention window where the algorithm achieves its best performance in terms of fairness. The contention window (CW) size has been incremented from 31 to 1024. The results of the simulations are shown in figure 41 below.

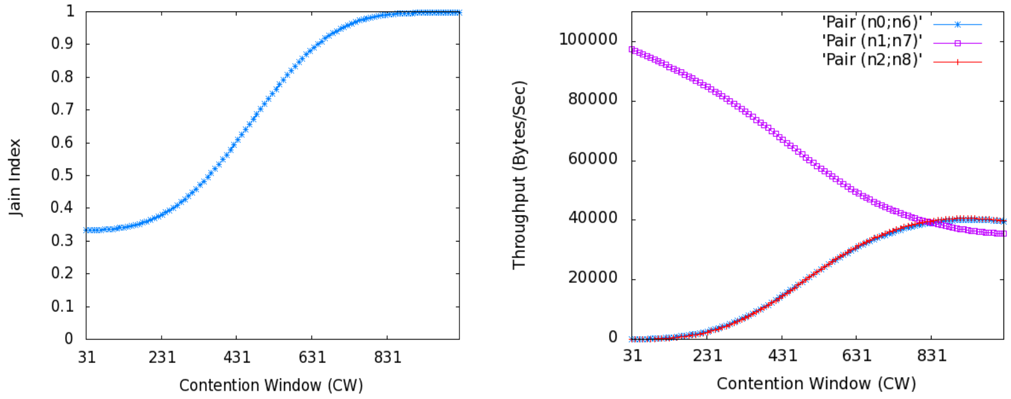


Figure 41: Fairness Index (left), Impact of Contention Window on Fairness (right)

The graph on the left of figure 41 illustrates that, as the contention window is increased, the competing nodes start to fairly share the medium. Fair bandwidth allocation among the competing nodes is achieved when the contention window size is in the range 831 and 1024 as the graph on the right of figure 41 confirms.

Figure 41 and the fairness index based on Jain's equation as a function of the set CW show that the network is fair when the contention window size is equal or higher than 831 at which the fairness index is 0.9973. Therefore, the contention window size in the TRCAF algorithm is set to 831, at which point the three pairs (n_0; n_6), (n_1; n_7) and (n_2; n_8) achieve (41916 B/s), (37482 B/s) and (41970 B/s) respectively. In contrast, the standard MAC protocol for the pairs (n_0; n_6), (n_1; n_7) and (n_2; n_8) achieved (834 B/s), (85656 B/s) and (834 B/s) respectively. Thus, the standard MAC protocol achieves a total throughput of 87324 B/s whereas the TRCAF algorithm achieves total throughput of 121368 B/s. Therefore, not only the TRCAF algorithm fairly distributes the channel resources among the three flows but also achieves 28% higher aggregate throughput than the standard MAC protocol.

In order to evaluate the performance of the TRCAF algorithm, a three flow topology has been designed, with four hops in each flow and the network is contented in the centre. This topology is illustrated in figure 42 below.

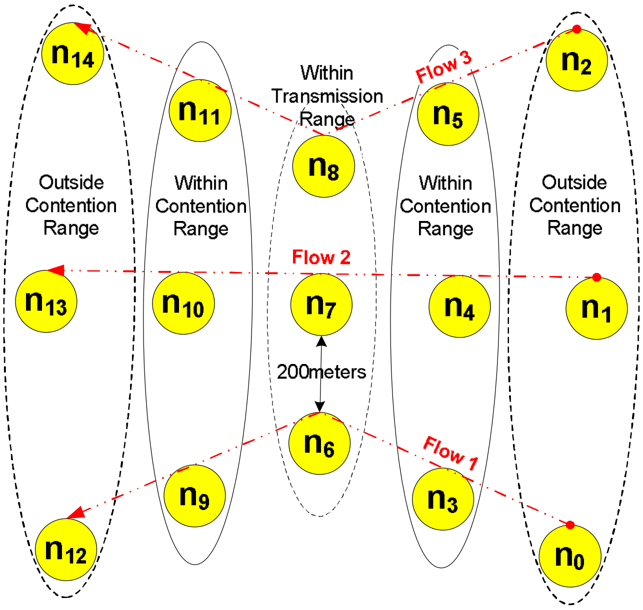


Figure 42: A fixed radial topology

There are three competing flows in the topology shown in figure 42 where the distance between each two consecutive nodes in each flow is 200 meters. The source and destination nodes of the three flows are outside the interference range, the intermediate nodes from the senders and receivers are within the interference range, but outside the transmission range of each other, and the nodes n_6, n_7 and n_8 are within transmission range.

In order to evaluate the performance of the TRCAF algorithm in comparison to the standard MAC, simulations have been run for 500 seconds in ns2 and the output is presented in figure 43 below.

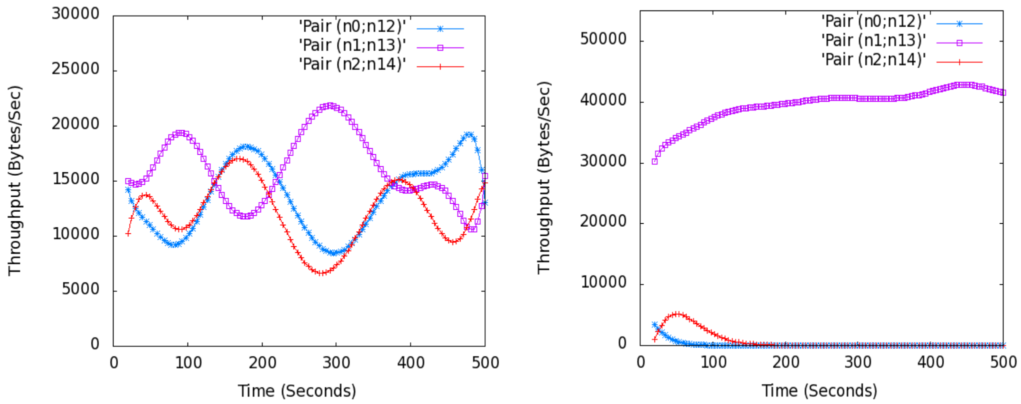


Figure 43: Performance comparison between TRCAF (left) and standard MAC protocol (right) for the radial topology

The plot on the right hand side of figure 43 shows that the MAC protocol does not fairly share the bandwidth between the three competing flows as the pairs (n_0; n_{12}), (n_1; n_{13}) and (n_2; n_{14}) achieve (138 *B/s*), (39606 *B/s*) and (738 *B/s*) respectively. However, the plot on the left of the same figure illustrates that the TRCAF algorithm fairly shares the bandwidth between the three competing flows and the pairs (n_0; n_{12}), (n_1; n_{13}) and (n_2; n_{14}) achieve (13554 *B/s*), (16056 *B/s*) and (12048 *B/s*) respectively. Applying these results to Jain's fairness equation, the fairness index is 0.34 for the MAC protocol and 0.98 for the TRCAF algorithm. This demonstrates that the TRCAF algorithm does achieve fairness. In addition, the TRCAF algorithm achieves 41658 *B/s* total throughputs, whereas, standard MAC protocol achieves 40482 *B/s* total throughputs. Therefore, not only does the TRCAF algorithm fairly distributes the channel resources fairly but also increases the channel utilisation by 3%.

4.10 Conclusion

In this chapter, an evaluation of the four proposed algorithms has been presented using the network simulator ns2. With the exception of the Four Hop Delay algorithm which is an improvement on a previously proposed algorithm TCP-AP, the other algorithms namely FBDMAC, NTRC and TRCAF are novel algorithms and their performance was evaluated against the performance of standard IEEE 802.11 MAC protocol in various scenarios. This is due the lack of the implementations of prior research studies in the field of fairness in ad hoc networks and also due the proposed algorithms designed to improve fairness in certain specific topologies where the standard MAC protocol proves to be inefficient and distribute the channel resources unfairly among the competing flows. The Four Hop Delay MAC algorithm was shown to achieve higher throughput and goodput then TCP-AP and also TCP New Reno and delays packet transmissions based on the usage of most recent round trip time and also via a better estimation of the four hop delay. The mechanism alleviates the effect of the hidden nodes and was shown to be efficient in presence of inter and intra flow contentions. In addition, evaluations of the novel mechanisms presented to overcome the unfair bandwidth distribution exhibited by the standard MAC protocol were demonstrated. The FBDMAC mechanism was evaluated against single hop chains where the nodes are outside the transmission range of each other, but within contention range in the first scenario which consisted of three competing flows over the single medium. Furthermore, the FBDMAC algorithm was also evaluated in a scenario where the senders were within transmission range of each and the receivers were outside the transmission range. In both scenarios the FBDMAC algorithm maintained fair channel resources distribution over the duration of the simulation. The proposed method provides a balanced distribution of network resources amongst the competing nodes and achieves 99% fairness according to Jain's fairness equation as opposed to 33% for the standard MAC protocol. The NTRC and TRCAF mechanisms were evaluated and shown to achieve 99% fair bandwidth distribution using Jain's fairness index in a number of identified scenarios, in which the standard MAC protocol favours one flow and leading the others to starve, including a radial fixed structure that covers half a circular area with two and four hops between the sender and receiver.

Chapter 5. **Fair Channel Resources Distribution Framework**

5.1 Introduction

Improving quality of service in wireless ad hoc networks has been a topic of extensive research for many years; despite their advantages in terms of infrastructure setup, ad hoc networks suffer from unfair bandwidth distribution among competing flows due to the single channel access, interference, hidden nodes and the absence of a central station to manage the network. To solve the unfairness problem, the IEEE802.11 [35] MAC protocol uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [9] protocol to reduce collisions by sensing the channel before each transmission. If the channel is not idle, the node backs off transmission by a period specified by the Exponential Back off Algorithm [10] implemented in the Distributed Coordination Function (DCF) [11] and schedules its transmission by updating its Network Allocation Vector (NAV) with the transmission time required specified in the RTS/CTS messages when the competing nodes are within the same transmission range of each other, hence, the nodes can hear each other's transmission and schedules them accordingly. On the other hand, if the nodes are in the interference range of each other, updating NAV is not possible as the RTS/CTS message cannot be read and the nodes may transmit at the same time, leading to collisions which exponentially increase the back off time. This leads to unfair channel distribution as the MAC protocol would always favour the last successful transmission, while the other nodes are backing off.

Many researchers have examined the unfair bandwidth allocation and proposed schemes to improve fairness in ad hoc networks. Researchers in [33] investigated the limitations of the exponential back off algorithm in the presence of high contention and proposed to increase the contention window size (CW) linearly as opposed to exponentially based on the node's transmission and reception rate over a time interval. However, researchers in [14] calculated the successive back off time as the product of the previous back off time, its log and a time slot in order to prevent increasing the contention window size exponentially. Researchers in [37] presented a fair bandwidth distribution scheme among TCP flows by varying the queue output rate at the network layer according to the severity of the contention experienced by the network every time a packet is passed from the network layer to the MAC layer; although it provided an improvement in terms of fairness, the proposed scheme experienced more than 11% of throughput loss. The concept of authority and ordinary nodes is introduced in [40] in addition to a Contention Window Based Fairness Back off algorithm, which favours the authority nodes to access the channel while limiting the ordinary nodes from accessing the channel. The ordinary nodes only gain access to the channel after a number of successive transmission failures. The proposed scheme also ensures that no node would be left to starve. In [41], a neighbourhood Random Early Detection (RED) algorithm was implemented at the network layer, requiring no modification to the MAC protocol, in order to detect contention and improve fairness. The approach required each node to monitor its queue size and broadcast it through network congestion notification (NCN) control packets to the nodes within its transmission range to decide whether to drop packets from the queue and ease contention. In [42], the researchers proposed a fair share estimation algorithm of the channel resources between nodes sharing the same channel. Each node in the network estimates the amount of channel resources that is being assigned to other nodes based on how many packets they have transmitted and then modifies their contention window size according to a predefined fairness metrics in order to achieve the desired fairness. The researchers also noticed that the proposed algorithm does sacrifice some throughput in order to achieve an acceptable level of fairness.

In this chapter, the proposed FHDMAC, FBDMAC, NTRC and TRCAF algorithms are integrated together to form the Fair Channel Resources Distribution (FCRD) framework to alleviate packet loss due to intra flow contention and hidden nodes, and improve the channel utilisation in chain topologies consisting of more than three hops to destination. Furthermore, in scenarios where the standard MAC protocol fails to ensure fair access to the medium, the FCRD framework dynamically adapt the suitable solution to the chosen topology in order to fairly share the available channel resources among the competing nodes.

5.2 Integration of the Proposed Algorithms

The MAC 802.11 distributes the channel resources fairly among multiple flows that compete over the same channel resources only in topologies where all nodes i.e. senders and receivers are within the same transmission range of each other. However, in scenarios where nodes compete with each other over the medium and are within interference range the MAC protocol fails to distribute the channel resources fairly among the nodes leading to greedy nodes that capture the channel and leave the others to starve. As shown in the simulations in the previous chapter, the starving nodes may not transmit any packets at all. Scheduling transmissions is not possible in scenarios where the competing nodes cannot communicate or hear each other's transmissions, leading to collision and unfair bandwidth distribution. In chapter 3, the solutions presented tackle the unfair channel resources distribution in specific scenarios where the MAC protocol extremely penalises the starving nodes in favour to the greedy nodes. However, this chapter presents a framework where all the proposed solutions to the aforementioned problems are integrated to form what is named Fair Channel Resources Distribution (FCRD) framework. The FCRD framework permits the nodes to dynamically adapt to the scenario they are faced with and ensures that the channel resources are distributed fairly among the competing nodes regardless of the topology and specifically where the standard MAC protocol fails to achieve fair bandwidth distribution. After careful analysis, the situations that the competing nodes face can be summarised as follow:

1. Competing nodes are outside the transmission range and within interference range of each other.

2. Source nodes are within transmission range of each other and the receivers are within interference range of each other.

3. Destination nodes are within transmission range each other and the source nodes are within interference range of each other.

4. All nodes are within transmission of each other.

The nodes can schedule access to the channel if they are all within transmission range through updating their Network Allocation Vector with the correct transmission duration required that are specified in the RTS/CTS control packets. However, in the other situations, the MAC 802.11 protocol fails to ensure fair access to the available bandwidth.

The FCRD framework is implemented as an enhancement to the MAC protocol and designed to overcome the weaknesses of the MAC protocol in terms of ensuring fairness in the scenarios described earlier as well increasing the channel utilisation. The FCRD framework is triggered upon the failure of the Distributed Coordination Function of the MAC protocol to fairly distribute the channel resources among competing flows. The FCRD framework integrates the four algorithms presented in chapter 3 as shown in figure 44.

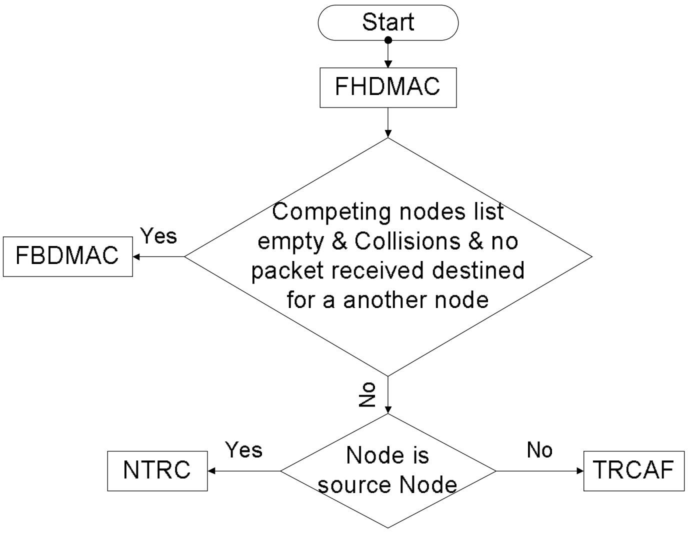

Start
FHDMAC
Competing nodes list empty & Collisions & no packet received destined for a another node
Yes
FBDMAC
No
Node is source Node
Yes
NTRC
No
TRCAF

Figure 44: Overview of FCRD framework

Figure 44 illustrates the proposed solutions integrated together to form the FCRD framework. In addition, the figure shows how the FCRD framework triggers the appropriate algorithm based on the positioning of the nodes in the topology. If the node experiences collisions and no competing nodes are detected within transmission range, the framework triggers the FBDMAC mechanism to ensure fairness in such scenarios. If the competing nodes are source nodes within transmission range of each other and the moving average for a competing node is below the threshold, then the NTRC mechanism is called to regulate the distribution of the channel resources. Finally, if the competing nodes are destination nodes and the moving average for a competing node is below the threshold, then the TRCAF scheme is called to maintain fair channel resources distribution. In scenarios where the number of hops from source to destination is more than three hops, the framework utilises the FHDMAC mechanism to pace TCP packet transmissions, to allow the packet to travel four hops away before sending a new packet. Thus, avoiding hidden node collisions and intra flow contention.

Firstly, the FCRD paces the transmission of TCP packets when the chain consists of more than three hops in order to eliminate the hidden node phenomenon, which results in interflow collisions. This is achieved by sending a TCP packet every time the four hop delay duration expires which is calculated as per the following equation every time a packet is ready for transmission:

$$FHD = (2+\alpha)*\left(\frac{RTT}{h} + \frac{PktSize - ACKSize}{bandwidth}\right)$$

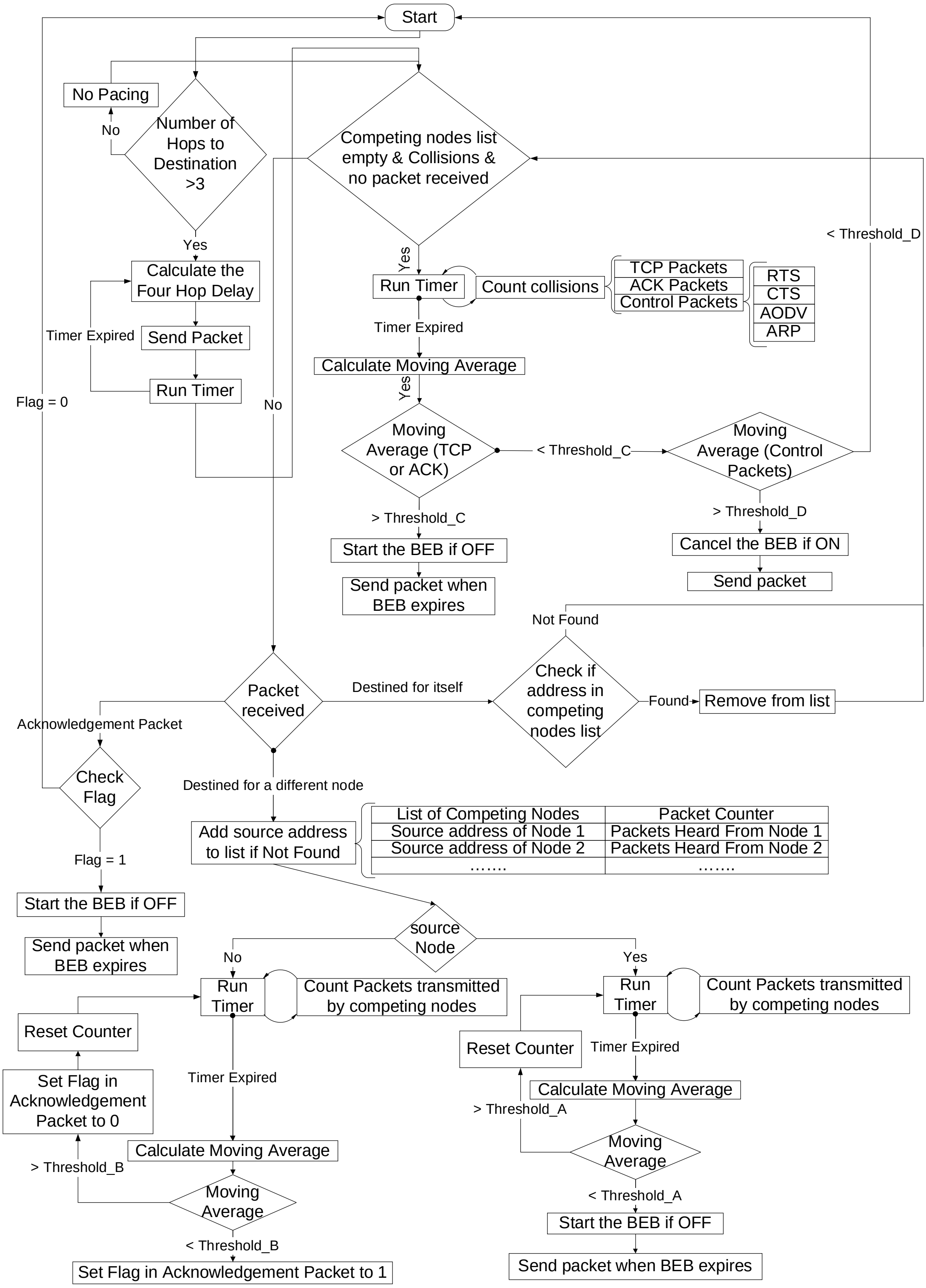
In addition, the FCRD framework employs a mechanism that tackles unfairness in scenarios where there are no competing nodes within the transmission range and the transmitted packets encounter collisions. If the RTS messages collision rate is higher than a predefined threshold then this implies that there are nodes competing for the channel, even though they cannot be detected as they are outside the transmission range but within interference range. In such case, the FCRD framework counts the number of collisions for each packet type such as control packets (RTS/CTS/ARP/AODV) and Data (TCP/ACK) over a time interval and calculates a moving average for each packet type as per the following equation.

$$\mathrm{avg}_{i+1} = \mathrm{avg}_i * \alpha + \frac{count_{i+1}}{\mathrm{waitingTim\ e}} * \beta$$

Initially, the parameters were set to α = β = 0.5 but following from the preliminary tests in which α values were increased from 0.1 to 0.9 while decreasing β values from 0.9 to 0.1 and monitoring the throughput obtained for each set of simulation, α was set to 0.65 and β was set to 0.45 to avoid sharp changes to the moving average. If the moving average for data packets is high then a predefined threshold then the node is greedy and is penalised by starting its back off timer. If the moving average for control packets is high then another predefined threshold then it is an indication that the node is starving and its back off timer is disabled to make the node more aggressive in requesting access to the channel.

Furthermore, each time a node hears a transmitted packet that is not intended for itself, the source address is extracted and compared to the addresses of its intermediate hops. If the address is not in the list, then it is of a competing node. The competing node’s list is stored and every time a packet is transmitted from that node a counter is incremented and reset upon the expiry of a time interval. Next, a moving average for each node in the list based on the number of packets counted is calculated every time the interval time expires as per the equation shown earlier.

If the node is a receiver node and the moving average is below a threshold established from observing the moving averages of the nodes during simulations, then the competing node that the moving average corresponds to is starving and when the node acknowledges the reception of the data frame a MAC layer acknowledgement is sent with a flag in its header to indicate contention. Upon the reception of the acknowledgement packet by the intermediate nodes, the intermediate nodes backs off transmission in order to allow the starving nodes to capture the channel, transmit packets and achieve fairness. Finally, if the node is a source node and the moving average is below a predefined threshold the node that the moving average corresponds to is starving and the initial node has to back off its transmission in order to allow the starving node to capture the channel and transmit packets too. The operation and design of the FCRD framework is further illustrated in figure 45.



List of Competing Nodes	Packet Counter
Source address of Node 1	Packets Heard From Node 1
Source address of Node 2	Packets Heard From Node 2
…….	…….

Figure 45: FCRD Framework Design

The FCRD framework is designed to enhance the performance of the MAC 802.11 protocol. In scenarios where the MAC 802.11 fails to distribute the available bandwidth fairly among the competing nodes, the FCRD framework is triggered to tackle the problem and distribute the channel resources fairly and improve the channel utilisation in ad hoc networks. The FCRD framework is implemented in ns2 and the pseudo code is provided in the following.

```
/* Terminology
PktType = DATA || ACK || AODV || RTS || CTS
Δt1 = 0.2 second
Δt2 = 0.2 second
cw = Contention Window
α = 100
pkt_rcv = packet received
BK = Back Off Algorithm
coll = collisions
*/
while(simulation)
 do
 for node i=1 to N
 if(hopsToDestination > 3)
   calculate FourHopDelay
   Packet.send
   runTimer
   if(timer.expire)
    packet.send
  if(competingArray.empty && coll)
   avg[PktType]=0
   if(collision)
    coll[PktType]=0
  timer=0
  while(timer <= Δt1)
     coll[PktType] ++
   end if
   avg[PktType]=(avg[PktType]+(coll[PktType]/Δt1)*α)/(α+1)
   if (avg[(DATA||ACK)]>β1)
  // Node is greedy
  //Penalise – increase cw, start back off algorithm
  Drop(packet)
  cw=cw*2
  BK.start(cw)
   end if
   else if (avg[(RTS||CTS||AODV||ARP)]>β2)
  //Node is starving
  //Reward – cancel exponential back off algorithm
  if(BK.on)
```

```
     BK.stop
     send (packet)
  end if
case received packet:
  if(pkt(dst) != node_i(addr))
competingArray[j].addr = pkt_rcv(src)
    // Source address is my competing node’s
    // address or my next hop address
  end if
 case send:
 if(pkt(dst) == CompetingArray[j])
     // The address is my next hop address
     // so remove from myArray in order not
     // to influence on decisions
 delete CompetingArray[j]
 end if
 while (timer <= Δt2)
 if(relay packet)
     CompetingArray[j].counter++
 end if
 end while
 update moving_average
 if(node == sourceNode)
  if (moving_average[i] < β3)
  // Node is greedy
  //Penalise –start back off algorithm to allow its neighbours
to transmit
  Backoff.start(Contention Window)
  end if
 if(node == destinationNode)
 if (moving_average[i] < β4)
  // Competing node is starving
  MAC_Acknowledgement.flag = 1
 else
  MAC_Acknowledgement.flag = 0
 end if else
 if (MAC_Acknowledgement.flag)
   BK.start(cwnd)
 end if
done
```

The FCRD framework employs the FHDMAC algorithm in order to pace packet transmissions by introducing a four hop delay. The FCRD framework differentiates between intermediate and competing nodes and calculates the transmission rate of the competing nodes to determine if any competing node is starving. Furthermore, if the node is a source node and competes with other nodes over the available medium, the FCRD engages the NTRC mechanism. If the node is a receiving node the FCRD framework engages the TRCAF mechanism to ensure fair bandwidth allocation among the competing nodes. The FCRD framework dynamically adapts and uses the appropriate mechanism according to the topology used. In addition, the FCRD framework dynamically determines the type of the node i.e. source or receiver, in order to use the appropriate mechanism and solution. In the implementation of the pseudo code, the thresholds β1, β3 and β4 were set to 1, and β2 was set to 0.2 and the contention window was set to 831 as the preliminary experiments carried to determine these values showed that the framework fairly distributes the channel resources at these values.

5.3 Summary

The FCRD framework aims to provide fair bandwidth sharing among the competing flows over the same channel as well as improving the channel utilisation through reducing the collision rate and eliminating the effects of the inter and intra flow contention by signalling contention to senders who, in turn, self-police to allow starving nodes to transmit. The framework relies on each node being able to dynamically differentiate between its communicating or intermediate nodes and competing nodes. The FCRD framework does not create any packets overhead. In addition, the FCRD framework is designed not to reduce throughput on the expense of fair channel distribution as opposed to previous solutions in the field; each node monitors how many packets its competing nodes have received and sent and, if the competing nodes are starving, the nodes set a flag in the MAC acknowledgement to instruct its communicating node to stop transmitting and allow the starving nodes to receive and send packets. In addition, where it is not possible to receive the packets that the competing nodes send, the FCRD framework switches to monitoring the collision rate that each node experiences. If the collision rate is greater than a predefined threshold for data packets, then the node becomes greedy and cancels its exponential back off algorithm. Also, if the collision rate for control packets is higher than a predefined threshold then the node establishes that it has to halt transmission and allow the competing node to transmit. Furthermore, the FCRD framework paces packet transmission to eliminate the effects of the hidden nodes. Due to the fact that the proposed NTRC and TRCAF mechanisms share common features, such as monitoring the transmission rate of the competing nodes and use the same moving average formula, making their integration feasible with FHDMAC as they all aim to impose fair bandwidth allocation. In the next Chapter, the FCRD framework is evaluated to determine how fair it is in comparison to the standard MAC protocol as well as the channel utilisation in a number of identified scenarios.

Chapter 6. **Evaluation of the FCRD Framework**

6.1 Introduction

This chapter evaluates the strengths and weaknesses of the Fair Channel Resource Distribution framework presented in chapter 5, through simulations in ns2, in a number of scenarios, varying in terms of the number of hops from source to destination and the positioning of the nodes, in a static topology, where the mobility of the nodes has been omitted. The FCRD framework is not a replacement of the standard MAC 802.11 protocol but an enhancement to the 802.11 MAC protocol and is intended to overcome the weaknesses of 802.11 MAC protocol in scenarios where it fails to distribute the single channel resources fairly among the competing flows. The FCRD framework could not be evaluated against prior research for two reasons: the lack of the source code for prior research enhancements and limited prior research in the area of fair bandwidth distribution for multi hops and flows sharing the same channel resources.

The evaluation of the performance of the FCRD framework in real network topologies was not possible due to the fact that the topologies for which the 802.11 MAC protocol fails to achieve fair channel resources distribution can span over 1 km radius.

6.2 Environment

Similar to the individual protocols' evaluations, the FCRD framework was implemented in ns2. The resulting trace files were analysed to determine the throughput that the FCRD framework and the standard MAC protocol achieved as well as packet collisions and other crucial information to examine the network performance. The transmission range is set to 250 meters and the interference range is set to 550 meters in ns2. The nodes have been carefully positioned in the topologies in which the standard MAC 802.11 protocol exhibits unfair bandwidth distribution leaving some nodes to starve. The FCRD framework is then switched on and the results plotted for comparison with the performance of the 802.11 MAC protocol. The topologies used consisted of three flows sharing the same channel resources. If the number of flows exceeds three; where the distance between each two flow is 300 meters as an example the fourth flow would be sharing different channel with flow three as the interference range within the same channel is 550 meters. The number of hops in the topologies also ranges between one hop and six hops. The later is to test all the functionalities of the FCRD framework in comparison to the standard MAC 802.11 protocol. In all the topologies experimented, the nodes were static and no node was mobile throughout the duration of the simulation. As a result, no mobility-induced losses affected the throughput achieved while testing performance of the proposed framework and that of the standard MAC protocol.

6.2.1 Topology 1 (Senders outside interference range)

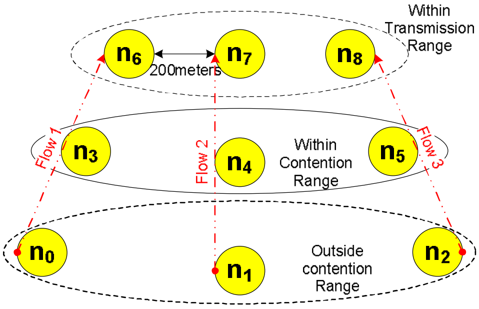


Figure 46: Receivers within transmission range

Figure 46 presents a radial fixed structure, which is one of the most common topologies, to cover half a circular area that was used as the basic topology, given its potential for the MAC 802.11 protocol to perform extremely unfair. There are 9 nodes forming three flows competing with each other over a single shared channel at the receiving nodes (n_6; n_7; n_8) which are two hops away from the sending nodes (n_0; n_1; n_2). The source nodes are outside the interference range of each other and the intermediate nodes (n_3; n_4; n_5) are within interference range but outside the transmission range. Finally, the destination nodes (n_6; n_7; n_8) within transmission range of each other. Based on the topology design, if (n_0; n_1; n_2) have data to transmit to their intermediate nodes (n_3; n_4; n_5) respectively, the channel is always idle and the nodes proceed with the transmission leading to collisions at their intermediate nodes. The intermediate nodes, which are outside transmission range, would not be able to update their Network Allocation Vector, as the RTS/CTS messages cannot be read by the intermediate nodes. In this case, the nodes that suffer collisions would randomly choose a back off time and exponentially increase it every time a collision occurs. On the other hand, the nodes that did not experience collisions would always have higher chances of accessing the channel than the nodes that backed off their transmissions. This leads to severe unfairness among the competing flows as the standard MAC protocol does not have any mechanism that tackles the unfair bandwidth sharing in scenarios such as the scenario presented in figure 46. In order to investigate the performance of the FCRD framework as well as the standard MAC 802.11 protocol in such a topology; simulations have been run for 500 seconds in which the competing flows start transmitting at the same time. The simulation results for both the standard MAC protocol and the FCRD framework have been plotted in the following figures to demonstrate their performances.

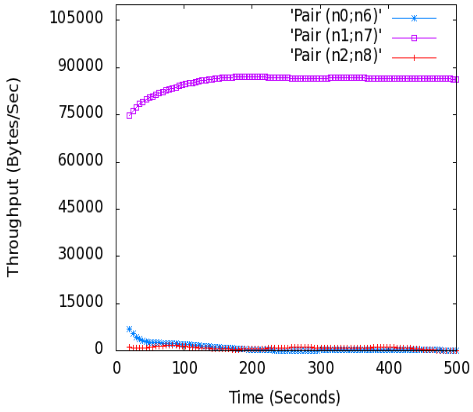


Figure 47: Standard MAC Protocol Performance

Figure 47 shows that the standard MAC protocol is unfair as the pair (n_1; n_7) dominates access to the channel and maintain a transmission speed of 85 KB/s leaving the other two pairs to starve and only achieve 834 B/s, even though the three flows started transmission at the same time. The source nodes transmitted their packets at the same time assuming that the channel is free as they were outside the contention range of each other. When the packets reached the intermediate nodes, collisions occurred and the intermediate nodes could not synchronize their transmissions as they were outside the transmission range of each other and randomly backed off. Pair (n_1; n_7) did not experience collisions, hence, did not back off transmission and gained access to the channel. The other two pairs (n_0; n_6) and (n_2; n_8) were unable to access the channel as whenever their intermediate nodes have packets to transmit they find that pair (n_1; n_7) is occupying the channel.

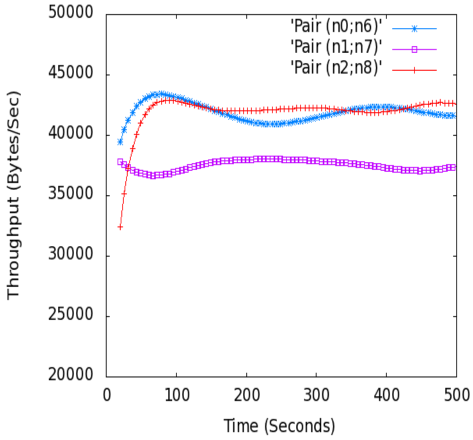


Figure 48: FCRD Framework Performance

In contrast, the performance of the FCRD, as shown in figure 48, fairly distributes the channel resources among the three competing flows. In such a scenario, the FCRD framework relies on the TRCAF algorithm to distribute the channel resources fairly as the destination nodes (n_6; n_7; n_8) are within transmission range, they can hear each other's packets. According to the FCRD framework, whenever a destination node notices that its competing nodes are starving, it sends a MAC acknowledgement with a flag to indicate contention. This flag instructs the intermediate nodes to back off transmission and allow the starving nodes to pick up transmission. The pairs (n_0; n_6) and (n_2; n_8) achieve 42 KB/s throughput and the pair (n_1; n_7) achieve 38 KB/s which is about 10% less than the other two pairs due to node n_7 having two nodes to compete with and therefore having to back off further in order to allow the other two nodes to increase their transmission and achieve fair bandwidth distribution among the competing nodes.

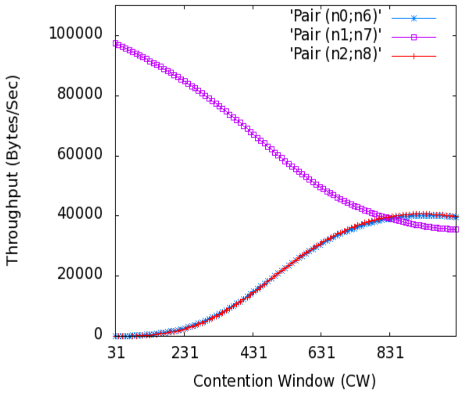


Figure 49: Impact of Contention Window size on Throughput achieved by FCRD Framework

In order to establish the correct back off time, the contention window size (CW) was incremented from 31 to 1024. Figure 49 shows that, as the contention window is increased, the difference between the throughputs achieved by each flow gets narrower and a fair bandwidth distribution is achieved when the contention window size is in the range 700 to 1024, at contention window size of 831 the three pairs (n_0; n_6), (n_1; n_7) and (n_2; n_8) achieve throughput speeds of 41916 B/s, 37482 B/s and, 41970 B/s respectively. This contrasts significantly with the standard MAC protocol where the pairs (n_0; n_6), (n_1; n_7) and (n_2; n_8) achieved 834 B/s, 85656 B/s and 834 B/s respectively as obtained from the simulation results shown in figure 47. Furthermore, the standard MAC protocol achieves 87324 B/s total throughput, whereas, the FCRD framework achieves a total throughput of 121368 B/s. Therefore, not only the FCRD framework fairly distributes the channel resources among the three flows but also achieves 28% higher aggregate throughput than the standard MAC protocol.

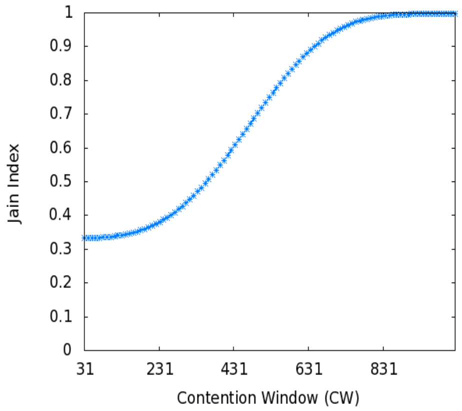


Figure 50: Fairness Index

The plot in figure 50 the fairness index calculated using Jain's equation and shows that the network is fair when the contention window size is equal or higher than 831 at which the fairness index is 0.9973.

6.2.2 Topology 2 (A radial fixed structure to cover a circular area)

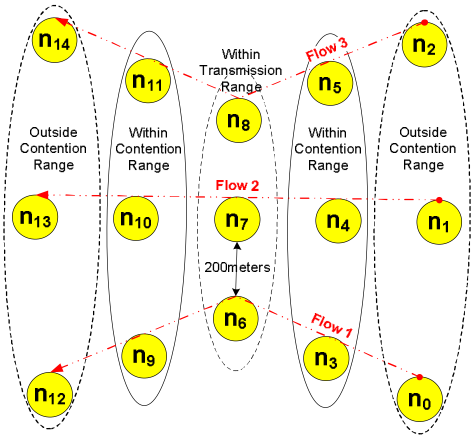


Figure 51: A radial fixed topology to cover a circular area

In order to evaluate the strength of the FCRD framework, a radial fixed topology has been designed with four hops in each flow and the network contention is at its most in the centre of the network. There are three competing flows in the topology shown in figure 51 where the distance between each two consecutive nodes in each flow is 200 meters as well as nodes n_6, n_7 and n_8 where the network is contented. The source and destination nodes of the three flows are outside the interference range and the intermediate nodes from the senders and receivers are within the interference range but outside the transmission range of each other.

In order to evaluate the performance of the FCRD framework in comparison to the standard MAC, simulations have been run for 500 seconds in ns2 and the results are plotted in the following figures.

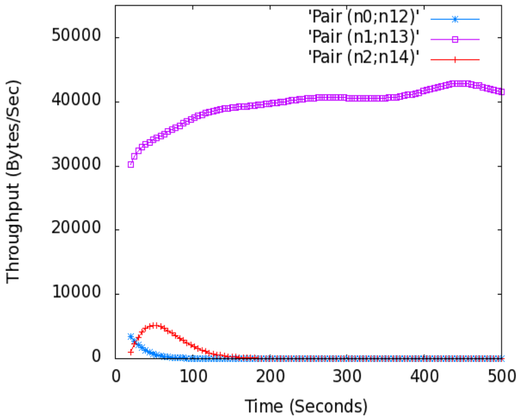


Figure 52: Standard MAC Protocol Performance for the radial topology

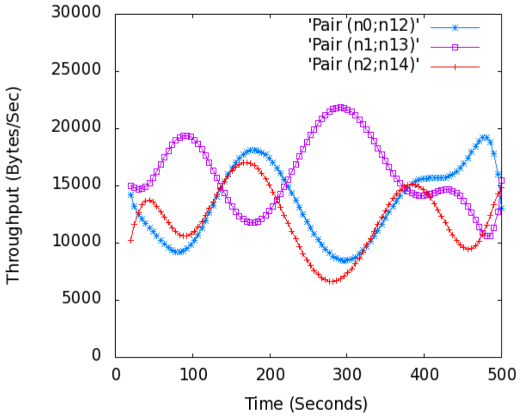


Figure 53: FCRD Framework Performance for the radial topology

The results of the simulation are presented in figures 52 and 53. Figure 52 shows that the MAC protocol does not fairly share the bandwidth between the three competing flows as the pairs (n_0; n_{12}), (n_1; n_{13}) and (n_2; n_{14}) achieved 138 B/s, 39606 B/s and 738 B/s respectively. However, figure 53 illustrates that the FCRD framework fairly shares the bandwidth between the three competing flows (n_0; n_{12}), (n_1; n_{13}) and (n_2; n_{14}) which achieve 13554 B/s, 16056 B/s and 12048 B/s respectively. Applying these results to Jain's fairness equation, the fairness index is 0.34 for the MAC protocol and 0.98 for the FCRD framework. This demonstrates that the FCRD framework does achieve fairness and improve the channel utilisation by 2.9%.

The FCRD framework is designed to be adaptable to a variety of network topologies. In the next set of simulations, the FCRD framework is tested on a further three topologies which have been selected in order to demonstrate the performance of the FCRD framework in comparison to the standard MAC 802.11 protocol.

6.2.3 Topology 3 (Competing nodes outside the transmission range)

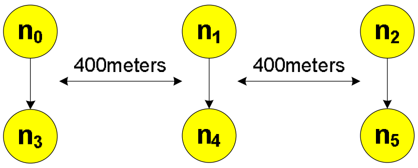


Figure 54: Symmetric topology

Figure 54 presents three competing flows where there is a single hop to destination. The distance between the three flows is 400 meters; hence, the flows are within contention range of each other and the distance between communicating nodes is 200 meters.

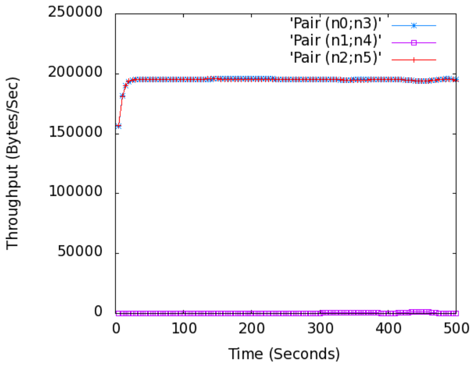


Figure 55: Standard MAC Protocol for the Symmetric Topology

The graph in figure 55 shows how the MAC 802.11 protocol performs for the scenario presented in figure 54. The pair (n_1; n_4) achieve very low transmission rate of 467 B/s over the 500 seconds of the simulation due to having to compete with two flows at each side. On the other hand, as the pair (n_1; n_4) is starving, the pairs (n_0; n_3) and (n_2; n_5) are outside the interference range of each and do not compete against each other. Therefore, the pairs (n_0; n_3) and (n_2; n_5) make the most of the channel resources and achieve 186086 B/s and 185943 B/s respectively. This is an extreme scenario where the MAC 802.11 protocol cannot distribute the channel resources fairly.

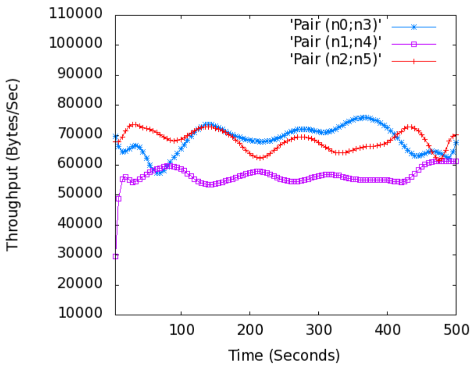


Figure 56: FCRD framework Performance for the Symmetric Topology

The graph presented in figure 56 demonstrates the performance of the FCRD framework in the symmetric topology presented in figure 54. Since, the nodes are outside the transmission range, the FCRD framework relies on the FBDMAC mechanism to impose fairness. The nodes start calculating their collision rates in order to determine if they are greedy and stop transmitting or starving and request access to the channel aggressively by cancelling their back off algorithm. In contrast to the standard MAC protocol, the FCRD framework fairly shares the bandwidth among the three flows and the pairs (n_0; n_3), (n_1; n_3) and (n_2; n_5) achieve 68592 B/s, 56242 B/s and 68262 B/s respectively which is 48% lower aggregate throughput than that achieved by the MAC protocol. This high aggregate throughput achieved by the MAC protocol is due to the fact that the two greedy flows are actually using two channel resources as they are outside the interference range and not sharing the same channel as in the case of the FCRD framework. The aggregate throughput that the three flows achieve in case of FCRD framework is 193096 B/s which is actually higher than that achieved by flow (n_0; n_3) which is 186086 B/s in case of the standard MAC protocol in which the flow (n_0; n_3) captured the channel throughout the simulation which makes the comparison valid.

6.2.4 Topology 4 (Competing nodes outside and inside transmission range)

The competing flows in figure 54 are outside the transmission range, but inside the interference range of each other. The topology presented in figure 57 includes competing flows consisting of four hops each, but with competing source nodes within interference range and competing receivers within transmission range. Some of the nodes in the competing flows can be detected and their transmitted packets monitored. Some of the nodes are within interference range, hence, they have to rely on monitoring their collisions rate in order to establish if they are greedy and halt transmission in order to alleviate contention and achieve fair bandwidth distribution. At the same time, the starving nodes would be more aggressive by not backing off transmission after a collision as would the MAC 802.l1 protocol when acquiring access to the channel.

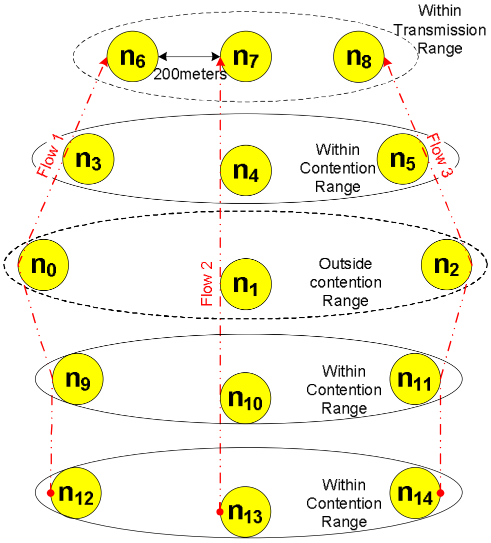


Figure 57: Competing nodes outside and inside transmission range

The difference between this topology and the previous topologies is that it includes two situations in the same scenario. The nodes can hear each others transmissions as in topology presented in figure 46 and where nodes have to rely on their collisions rate as in figure 54. There are three competing flows where each flow consists of four hops from the source to destination. The senders are within interference distance of each other and the receivers are within transmission range.

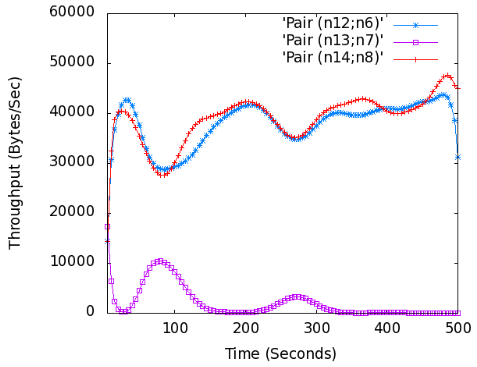


Figure 58: Standard MAC Performance

Figure 58 confirms that the MAC 802.11 is unfair as the pairs (n_{12}; n_6) and (n_{14}; n_8) obtain 37962 and 39012 B/s respectively and the pair (n_{13}; n_7) obtain 1998 B/s. These results demonstrate that the behaviour of the MAC 802.11 is similar to that presented for the topology in figure 54 as the middle pair (n_{13}; n_7) suffers from having to compete with the other two pairs at the same time.

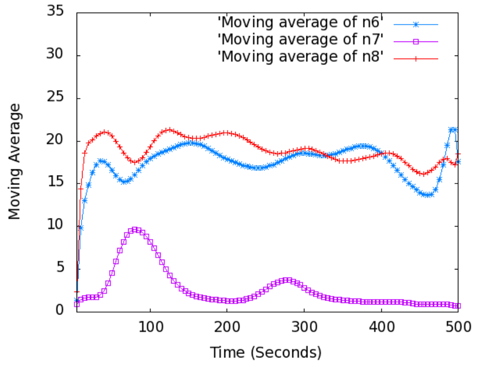


Figure 59: Packets Counted by Competing Nodes for Standard MAC Protocol

Figure 59 shows a graph of the packets that the competing receivers (n_6; n_7 and n_8) monitor about each other in the case of standard MAC protocol where the moving average of n_6 and n_8 shows the packets that n_7 monitors and which are for nodes n_6 and n_8 and the moving average of n_7 is the for the packets counted by nodes n_6 and n_8. The graph also confirms that nodes n_6 and n_8 can tell from the moving average of n_7 that it is starving, as shown in figure 58.

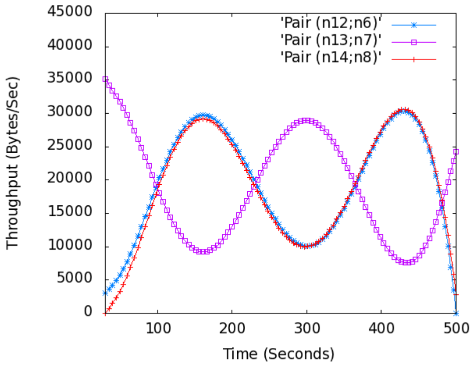


Figure 60: FCRD framework Performance

The performance of the FCRD framework for the same topology is shown in figure 60. The pairs (n_{12}; n_6), (n_{14}; n_8) and (n_{13}; n_7) achieve 19620 B/s, 18966 B/s and 19308 B/s respectively. Therefore, as opposed to the standard MAC protocol, the FCRD framework distributes the channel resources fairly among the competing flows even in complex scenarios. However, the FCRD framework slowly responds to the changes and decisions being made due to the decision having to propagate four hops along the chain in order to reach the senders.

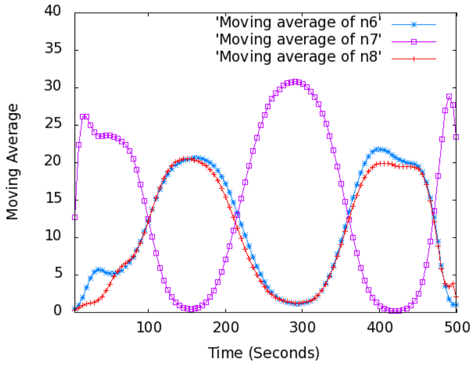


Figure 61: Packets Counted by Competing Nodes for FCRD framework

Figure 61 shows how the moving average calculated by the competing nodes matches the throughput achieved by each node. When node n_7 is starving the moving average of the competing nodes n_6 and n_8 is high and as the node n_7 picks up transmission the moving average of its competing nodes n_6 and n_8 drops. However, the FCRD framework struggles to maintain the same moving average of the number of packets heard by the competing nodes throughout the duration of the simulation.

6.2.5 Topology 5

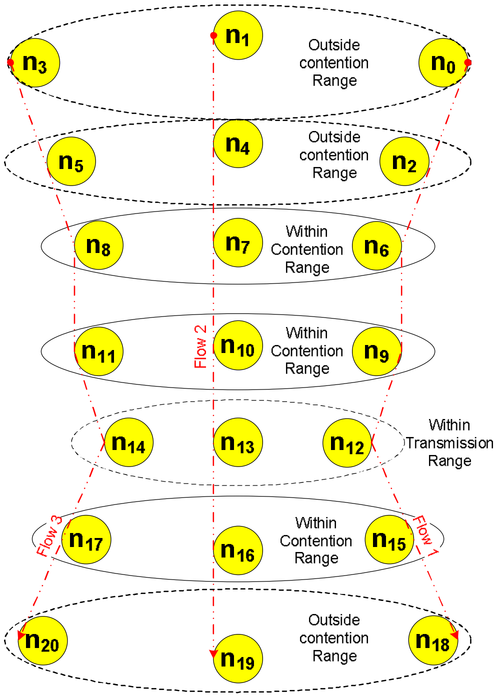


Figure 62: Topology 5

The most challenging topology yet to evaluate the strengths of the FCRD framework in comparison to the performance of the standard MAC protocol is presented in figure 62. Each flow consists of six hops from source to destination. The source nodes as well as their intermediate nodes are outside the interference range of each other. The source nodes do not compete with each other over the available medium. Since the distance from source to destination is six hops, the FCRD framework triggers the FHDMAC mechanism to pace TCP packet transmission in order to avoid intra flow contention and eliminate the occurrence of the hidden node problem. Therefore, the source node would not send the successive packet until the previous one has travelled four hops down the chain. When the packets reach nodes (n_6; n_7 and n_8) collisions occur as these nodes lie within interference range of each other.

These nodes are unable to synchronise their transmission as they are not within transmission range and transmit randomly leading to collisions. At which point the FCRD framework switches on the FBDMAC mechanism to penalise the greedy nodes and reward the starving nodes based on the collision rate the nodes experience. Furthermore, as the packets reach nodes n_{12}, n_{13} and n_{14}, which are within transmission range of each other, hence, can both hear each other transmission and count the packets they each send. In case of FCRD framework, the nodes monitor each others transmission and if one or more are starving since their moving average for packets transmitted is below the threshold. The non-starving node would back off its transmission by starting its binary exponential back off algorithm and allow the staving node to transmit packets as well as setting the flag in the acknowledgement packet to instruct the previous nodes in its chain to refrain from transmitting packets.

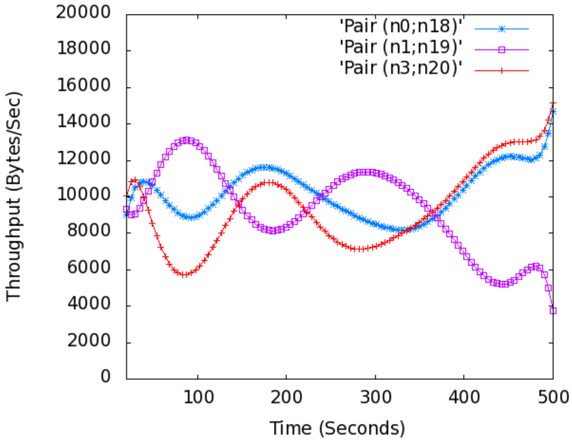


Figure 63: FCRD framework performance

Figure 63 presents the performance of the FCRD framework on the topology presented in figure 62. The three pairs (n_0;n_{18}), (n_1;n_{19}) and (n_3;n_{20}) fairly share the available medium throughout the simulation time of 500 seconds and achieve 10236.16 B/s, 9276.16 B/s and 9408.16 B/s respectively making a total aggregate throughput of 28920.48 B/s.

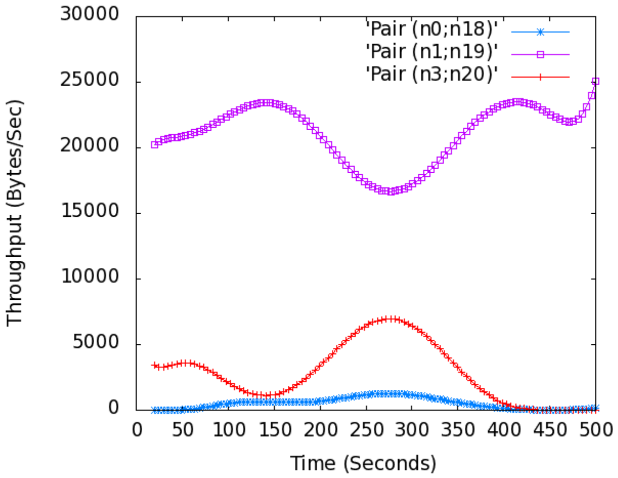


Figure 64: Standard 802.11 MAC performance

Figure 64 shows the performance of the standard MAC protocol. The pair (n_1; n_{19}) dominates access to the channel throughout the simulation time of 500 seconds; leaving the other two pairs (n_0; n_{18}) and (n_3; n_{20}) to starve. Since, all three flows start transmitting at the same time, the source nodes transmit their packets to their intermediate nodes that in turn forward their packets to their intermediate nodes with success given that the first two hops in each flow are outside the interference range of each other and do not interfere with each others transmission. However, as the packet travel along the six hops chains and reach the third nodes of each flow, collisions occur since the third nodes from source in each flow reside within interference range of each other. These nodes are unable to schedule their transmission accurately as they cannot receive the RTS messages. The trace files from the ns2 simulator show that node n_7 captures the channel and nodes n_6 and n_8 back off their transmission by a random value set by the binary exponential back off algorithm which is exponentially increased every time a collision occurs leaving the nodes n_6 and n_8 to starve which in turn impact the throughput achieved by the flows they belong to. The standard MAC 802.11 favours the flow formed by the pair (n_1; n_{19}) as its back off window is small throughout the duration of the simulation as it is the least flow to suffer from collisions. The throughput that the three pairs (n_0; n_{18}), (n_1; n_{19}) and (n_3; n_{20}) achieved from analysing the simulation trace file is 522.16 B/s, 21057.16 B/s and 2832.16 B/s respectively making a total aggregate throughput of 24411.48 B/s. These results show that not only the standard MAC protocol unfairly distribute the channel resources among the competing nodes, but also experience a throughput reduction of 15.59% in comparison to the throughput achieved by the FCRD framework.

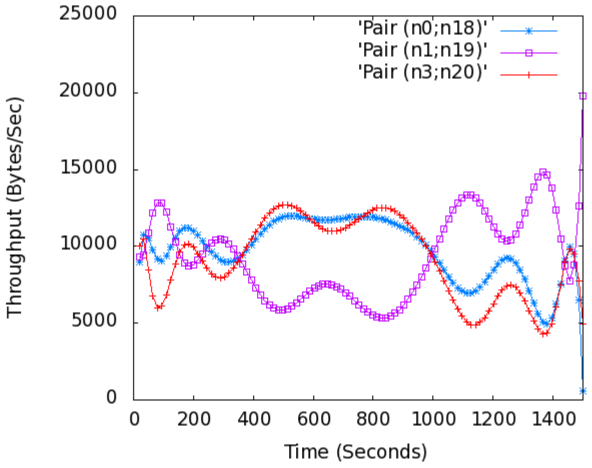


Figure 65: FCRD performance over 1500 seconds of simulation time

Figure 65 presents the throughput achieved by the three competing flows formed by the pairs (n_0; n_{18}), (n_1; n_{19}) and (n_3; n_{20}) in case of the FCRD framework. The bandwidth is fairly distributed among the competing flows throughout the simulation time of 1500 seconds and the pairs (n_0; n_{18}), (n_1; n_{19}) and (n_3; n_{20}) achieve 9780.05 B/s, 9316.05 B/s and 9174.05 B/s respectively. This demonstrates that the FCRD framework maintain its fair medium distribution even for longer simulation time.

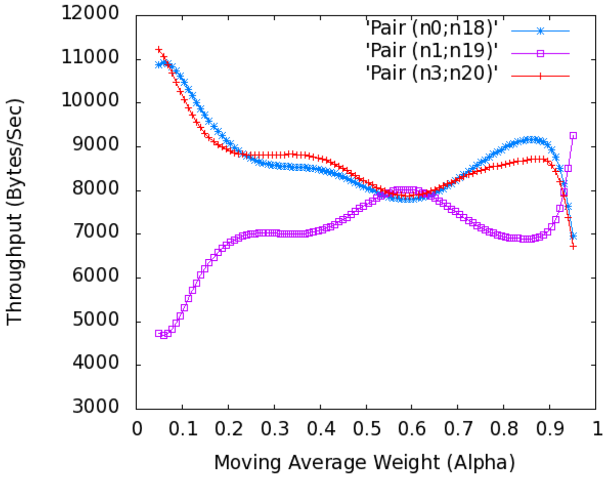


Figure 66: The impact of the moving average weightings on fairness

The graph in figure 66 shows how the alpha and beta weights in the moving average formula impact the performance of the FCRD framework in terms of overall throughput and fairness. The graph demonstrates that the three pairs (n_0; n_{18}), (n_1; n_{19}) and (n_3; n_{20}) fairly share medium between them when the α weighting for the moving average is 65% for the previous moving average and β is 45% for the new calculated average and for which three pairs (n_0; n_{18}), (n_1; n_{19}) and (n_3; n_{20}) achieve 8823.08B/s, 9480.08B/s and 9996.08 B/s respectively.

6.3 Summary

In this chapter, an evaluation of the performance of the Fair Channel Resources Distribution framework has been presented. The framework has been tested on a number of topologies varying in terms of the number hops in each chain and also in the positioning of the nodes to cover the possible scenarios in which the standard MAC protocol fails to share the available medium among the competing flows. The FCRD framework was evaluated on topologies consisting of 6, 8, 14 and 20 nodes. The framework is designed to be scalable since it is an enhancement to the MAC protocol. However, the throughput achieved in large topologies is lower than that in smaller topologies. This is inherited from the nature of wireless networks as the delay and channel errors increase in longer chains. In addition, The FCRD framework adapted to all the topologies presented dynamically and managed not only to fairly distribute the channel resources but also achieve higher throughput than the standard MAC protocol. The FCRD framework imposes fair medium distribution in situations where there is no information about the status of the network through monitoring the severity of packet collisions that nodes experience. In addition, the FCRD framework controls the channel resources distribution through feedback, included in MAC layer acknowledgements, about the status of the network. The signalling determines whether the intermediate nodes carry on transmitting or stop in order to help the starving nodes to gain access to the channel. Furthermore, besides pacing the transmission of the TCP packets to overcome the interflow collisions caused by the interflow contention due to the hidden nodes, the FCRD framework monitors the competing nodes transmission rate at source nodes to discover starving nodes and back off transmission in order to allow the competing starving nodes to pick up transmission and share the available resources in a fair and acceptable level while maintaining good transmission throughput that exceeds that achieved by the standard MAC protocol. Simulation results showed that the FCRD framework achieved 99% fairness according to Jain's fairness index in all topologies utilised, in contrast to the standard MAC protocol that achieved on average 33% under comparable overall utilization of the channel resources. In the next chapter, the limitations of the proposed mechanisms and framework are described as well as a conclusion and how to further improve the performance of the proposed solutions.

Chapter 7. **Conclusions and Future Work**

7.1 Achievements of the Research

The thesis aims to improve quality of service by proposing novel solutions to the problem of channel utilisation due to intra flow contention and hidden nodes, and the unfair channel resources distribution due to the behaviour of the standard MAC protocol in identified scenarios.

In chapter 2, a detailed literature review on the research challenges in wireless ad hoc networks were presented emphasizing on the areas where such networks under perform in comparison to their counterpart the wired networks. The state of the art research provided a good understanding and an insight into the challenges faced in wireless ad hoc networks in addition to prior research and the proposed solutions to overcome these challenges and improve the performance of the existing standards to accommodate the different environments that wireless networks are to be deployed in. Ad Hoc Networks have proven to be unable to differentiate between interference packet losses, congestion losses and route failures. If a packet fails be to be transmitted seven times it gets dropped and the route to destination is assumed invalid. This assumption is not always correct and rather than changing the protocol completely researchers provided solutions to prevent having to reach the seven times retransmission limit by providing more information about the state of the network that leads to take different action when retransmitting the packet. In addition, TCP retransmissions and MAC 802.11 retransmissions often lead to increased overhead and inconsistency between the transport and link layers. This led the research community to design solutions to this problem such as the Snoop Protocol in which TCP is aware of the Link layer retransmissions. The hidden node problem caused tremendous concern among researchers and is unique to wireless networks. On a separate stream, the studies showed that fair bandwidth allocation among competing nodes in different flows has not been exhaustively researched. Prior research focused on intra flow fairness or in other words fair bandwidth allocation among competing nodes within the same flow “from source to destination” as well as fair bandwidth distribution among competing flows but for single hop chains. Whereas, in this thesis, solutions were proposed to cover longer chains of up to six hops in each chain.

In chapter 3, novel solutions were proposed to enhance the performance of ad hoc networks in areas where the behaviour of the standard 802.11 MAC protocol is unacceptable in terms of providing quality of service such as throughput and fair bandwidth distribution. The research led to the proposal of four mechanisms, three of which are novel and one based on an improvement from a previously proposed solution named the TCP-AP. The FHDMAC mechanism outperforms TCP-AP and TCP New Reno as it paces packet transmissions based on the most recent round trip time and also via a better estimation of the time it requires for a packet to travel four hops before the successive packet is sent. The mechanism also alleviates the effect of the hidden nodes and improves throughput and was shown to be efficient in presence of intra flow contention. The remaining three mechanisms address improving fair bandwidth allocation in ad hoc networks through observing the state of the network and take certain decisions to maintain equal share of the channel resources available. The FBDMAC mechanism monitors the collisions that the node experiences in order to decide on whether to penalise greedy nodes or reward starving nodes and is designed to be adopted in scenarios where the competing nodes cannot communicate but, interfere with each others transmissions.

The FBDMAC mechanism has been shown to greatly improve fairness in scenarios where all flows are within interference. In addition, good choices of the threshold values improve the fairness even better depending on the network's topology. The NTRC was proposed to improve fairness in situations where the source nodes from each flow are within transmission range whereas the receiving nodes do not interfere with each others transmission. A study case was taken into consideration in order to demonstrate its performance in comparison to the standard MAC 802.11 protocol. The NTRC algorithm dynamically detects the node's neighbours and distinguishes them from its next hop nodes. It also fairly distributes the channel access between the competing nodes by monitoring if a node's neighbours have transmitted any packets during a time interval by calculating a moving average periodically to stay tuned with the state of the network. If the moving average is below a threshold then the node stops transmitting to allow its starving neighbours to transmit. The NTRC algorithm achieves 99.4% fairness based on Jain's fairness equation where the standard MAC protocol only achieves 33.6%. The unfair channel distribution of the MAC 802.11, in a scenario where the sending nodes are outside the interference range and receivers are within transmission, is improved via the TRCAF mechanism. This is done through feedback from the receiving nodes to the sending nodes in order to learn about the state of the network and whether to stop transmitting in order to alleviate the unfair distribution of channel resources. The TRCAF algorithm proves to impose fair bandwidth sharing among the competing flows over the same channel. In the TRCAF algorithm each node dynamically differentiates between its communicating and competing nodes. Each node monitors how many packets its competing nodes have received and if the competing nodes are starving then it sets a flag in the MAC acknowledgement to instruct its communicating nodes to stop transmitting in order to allow the starving nodes to receive and send packets. These novel mechanisms improved the performance of the MAC protocol in relation to fair bandwidth allocation among competing nodes sharing the same channel resources. Furthermore, the mechanisms also improved channel utilisation through increasing the throughput achievable. According to Jain's fairness index, the proposed mechanisms achieved over 99%. This percentage shows that the mechanisms distribute the channel resources equally among the competing flows. Also, the aggregate throughput was increased; hence, less bandwidth is wasted and the channel was utilised more effectively as opposed to the standard MAC protocol.

The performance of the proposed mechanisms was evaluated in chapter 4 using the ns2 simulator in various topologies. The proposed mechanisms were shown to not only distribute the available medium fairly as they all achieved over 99% which is total fairness, but also improved the channel utilisation in the scenarios they are designed for.

In chapter 5, the proposed mechanisms "FHDMAC, FBDMAC, NTRC and TRCAF" were integrated together to form the FCRD framework. The design and operation of the framework is described in detail in chapter 5. As opposed to the individual proposed mechanisms which are designed for specific topologies, the FCRD framework improves fairness in any topology where the standard MAC protocol was deemed to fail. The improvement is aimed at distributing the available bandwidth equally among the competing nodes. In addition, the framework maintained higher channel utilisation than that achieved by the MAC protocol. Depending on the positioning of the node in terms of outside or inside interference or transmission range, the FCRD framework selects the appropriate mechanism to adopt in order to overcome the unfair channel utilisation and fair bandwidth distribution which the flow experiences. The FCRD framework enhances the performance of the MAC protocol in ad hoc networks which help towards the successful deployment of ad hoc networks in the commercial and public sectors.

The FCRD framework has been validated through simulations in the network simulator ns2, in chapter 6, to demonstrate its effectiveness in comparison to the MAC protocol. The FCRD framework was shown to outperform the MAC protocol in the tested topologies in terms of channel utilisation and fair channel resources distribution. The testing comprised of various simulations of increasing complexity in terms of the number of hops from source to destination and the positioning of the competing nodes. In all simulations, the FCRD framework outperformed the standard MAC protocol. The FCRD framework achieved total fairness and distributed the channel resources equally among the competing nodes. Furthermore, the aggregate throughput achieved by the FCRD framework was higher than that achieved by the standard MAC protocol. This shows that the FCRD framework improves channel utilisation in comparison to the MAC protocol.

7.2 Limitations of the Research

The proposed mechanisms and framework to increase throughput and maintain fair bandwidth allocation among competing flows encountered a number of limitations, as summarised below:

- The validation of the proposed mechanism and the framework did not consider scenarios where the competing flows start transmitting at different times. Furthermore, the performances of the proposed schemes and framework were not tested in topologies with nodes of different transmission speeds and tests only included nodes transmitting data continuously.

- The design of the proposed mechanisms and framework did not address the fact that a node may stop transmitting data while the competing nodes have not.

- Although, the NTRC algorithm does take into account new neighbours that have come within transmission range when they transmit a packet, it does not take into consideration if the neighbours disconnect or stop transmitting.

- Although, over the simulation time fairness was achieved, TRCAF algorithm struggled to maintain it for each moment in time and was slow in responding to the unfair bandwidth distribution due to the fact that the acknowledgements have to propagate to the source node to instruct them to stop transmitting.

- The thresholds at which the algorithms penalise the greedy nodes and make the starving nodes more aggressive have been chosen based on running multiple simulations for which the proposed schemes achieve high throughput and fairness. These thresholds differ from topology and one another. Thus, having to pre-estimate the threshold values to be used cannot be done in a real network. The threshold choices impact on the performance of all mechanisms proposed. If the node was to be penalised thinking it is greedy when it is not due to a low threshold lead to unfair distribution of the channel resources. Furthermore, the time duration that the greedy nodes have to be penalised for is also crucial and should be as small as possible so that the node does not exercise a decision longer than necessary. If the node was to be penalised for a longer duration than it should be, unfair channel resources distribution would arise.

- The mobility of the nodes has not been taken into account due to the fact that it would raise further challenges related to routing protocol optimisation. The behaviour the proposed mechanisms has not been tested on a mobile environment due to the fact that the unfair bandwidth distribution exhibited by the MAC protocol could be influenced by mobility issues, therefore, by omitting mobility, the unfair bandwidth allocation would be solely due to the inefficiency of the MAC protocol do handle certain scenarios.

- The FCRD framework was evaluated in various predefined topologies but not in a random topology. This would have evaluated the performance of the framework in scenarios imitating what happens in reality. Furthermore, the FCRD framework was not evaluated in topologies in which flows cross each other “nodes route more than one flow” because none of the mechanisms that it integrates were designed to handle such a situation as a node would either back off transmission or not regardless of the number of the flows that it routes.

- In certain topologies the FCRD framework struggled to maintain fairness for each moment in time and was slow in responding to the decisions being taken. This was due to having long flows each consisting of four hops and leading to the decision having to propagate for few hops to reach the senders in case of the acknowledgement packets. In future work, the FCRD framework would be improved to react quickly to the decisions taken in the presence of flows consisting of more than two hops.

- The NTRC and TRCAF mechanisms adopted in the FCRD framework dynamically detect if a new node has been added through hearing their transmissions when within transmission range. However, if a node is silent, these mechanisms have no way of detecting this and the nodes monitoring the silent nodes would keep assuming that they are starving as they have not transmitted any packet. One way to overcome this problem is to periodically probe the silent nodes to detect their presence which in turn has consequences in terms of packet overhead and reduced throughput. Improvements and or alternatives must be proposed to overcome this assumption and take into account that the competing nodes may be starving or have stopped competing for the channel.

- On the exception of the FHDMAC mechanism, the proposed mechanisms were not evaluated in comparison to prior research solutions in the field of the fair bandwidth distribution due to the lack of the implementation of such solutions and the type of challenge they were proposed to tackle such as fairness within the same flow whereas the proposed solutions in this thesis tackle intra as well as inter flow fair channel resources distribution.

7.3 Future Work Suggestions

- Evaluation of the proposed schemes and framework in a generalised scenario, with a random topology and the nodes starting transmission at random time.

- Even though the FHDMAC mechanism achieves better throughput than TCP-AP, it is believed that a better estimation of the four hop delay could be achieved or even a different approach to overcome the hidden node problem could be proposed. Future work would look into alternative novel approaches to eliminate the effects that the hidden node problem imposes on channel utilisation.

- The NTRC, TRCAF and framework implementation need to take into account idle and disconnected nodes. A solution would be to probe the node from time to time.

- Dynamically selecting the corresponding thresholds to the contention experienced by the nodes will be investigated as the topology of an ad hoc network changes unpredictably over time. An algorithm that would learn about the network's topology and update the threshold values accordingly would be favoured. Also, how frequent should the average of the number of collision be calculated and what impact that has on fairness on one hand and on the overall channel utilisation on the other hand would be further investigated.

- The NTRC, TRCAF and FCRD will be improved to include scenarios where next hop nodes do route traffic for other flows or in other words where next hop nodes are also competing nodes for the channel resources at certain times.

- Introduce a mathematical model to determine the maximum theoretical throughput that the node could achieve based on the number of hops, flows in the scenario. If the nodes have access to transmission speeds, the number of nodes competing for the channel and the moving averages, they may dynamically decide on the correct threshold and waiting time values to use in order to equally distribute the channel resources. However, knowing how many competing nodes exist is not always possible as with the case of the FBDMAC mechanism which is designed to tackle unfairness when the nodes are outside the transmission range of each other, hence, they cannot determine the number of their competing nodes as they are out of reach.

- Evaluate the performance of the FCRD framework in real network topologies. In order to achieve this, a number of factors have to be taken into consideration. These factors include a flat area which spans to 1 km^2. The nodes have to be within line of sight of each other to eliminate packet drops caused by buildings or obstacles. Mobile nodes equipped with the same wireless transmission speed. The nodes have to be positioned carefully in order to imitate the topologies simulated and in which the standard MAC protocol suffers degraded performance. On the basis that the same conditions as in ns2 are met such as 2 Mb/s transmission speed, the type of antenna, the queue output rate and size, TCP Newreno as the transmission protocol and the routing protocol AODV. It is anticipated that the same results achieved using ns2 in terms of channel utilisation and fair bandwidth distribution for both the FCRD framework and the standard MAC protocol. Furthermore, the implementation of the framework resides in the MAC layer. In order for the framework to be applied in a wireless node, the driver for the wireless card has to be modified and the FCRD framework integrated in the MAC protocol. This is easier achieved under the Linux environment which provides full access and control to the user.

References

1. Bakht, H., *History of mobile ad-hoc networks*, in *the research seminar*. 2005, School of Computing and Mathematical Sciences: Liverpool.

2. Bakht, H., *Applications of mobile ad-hoc networks*, in *the research seminar*. 2005, School of Computing and Mathematical Sciences: Liverpool.

3. James, A.F. and L. Barry, *A DoD perspective on mobile Ad hoc networks*, in *Ad hoc networking*. 2001, Addison-Wesley Longman Publishing Co., Inc. p. 29-51.

4. Haas, Z.J., et al., *Wireless Ad Hoc Networks*, in *Wiley Encyclopedia of Telecommunications*, J.G. Proakis, Editor. John Wiley & Sons, 2002: New York.

5. Fifer, W. and F. Bruno, *The Low-Cost Packet Radio*, Proceedings of the IEEE, January 1987. 75(1): p. 33-42.

6. *Intel Technology Journal for WiMax http://www.intel.com/technology/itj/2004/volume08issue03*.

7. Chlamtac, I.A., M.B. Conti, and J. Liu, *Mobile ad hoc networking: imperatives and challenges*. Ad Hoc Networks 2003. 1(1): p. 13-64.

8. IEEE, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE Standard 802.11, June 1999.

9. Xin, W. and K. Kar. *Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks*. in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. 2005.

10. Vukovic, I.N. and N. Smavatkul, Saturation Throughput Analysis of Different Backoff Algorithms *in IEEE 802.11*. IEEE802.11[A]. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. Vol. 3, p. 1870-1875

11. Eshghi, F., A.K. Elhakeem, and Y.R. Shayan, *Performance evaluation of multihop ad hoc WLANs*. Communications Magazine, IEEE, 2005. 43(3): p. 107-115.

12. Joseph, D., et al., *A practical cross-layer mechanism for fairness in 802.11 networks*. Mob. Netw. Appl., 2006. 11(1): p. 37-45.

13. Hu, C., et al., *An Analysis of the Binary Exponential Backoff Algorithm in Distributed MAC Protocols*. http://lion.cs.uiuc.edu/~chunyuhu/publications/chunyuhu-UIUCDCS-R-2005-2599-corr.pdf, 2005.

14. Manaseer, S.S., M. Ould-Khaoua, and L.M. Mackenzie. *Logarithmic based backoff algorithm for MAC protocol in MANETs*. in *Proceedings of CSIT2006*. 2006. Jordan.

15. Hari, B., et al., *Improving TCP/IP performance over wireless networks*, in *Proceedings of the 1st annual international conference on Mobile computing and networking*. 1995, ACM: Berkeley, California, United States.

16. Rahman, A.H.A. and Z.A. Zukarnain, *Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad Hoc Networks*. European Journal of Scientific Research, 2009. **31**(4): p. 566-576.

17. Liu, J. and S. Singh, *ATCP: TCP for mobile ad hoc networks*. Selected Areas in Communications, IEEE Journal on, 2001. **19**(7): p. 1300-1315.

18. Abolhasan, M., T. Wysocki, and E. Dutkiewicz, *A review of routing protocols for mobile ad hoc networks.* Ad Hoc Networks, 2004. **2**(1): p. 1-22.

19. Perkins, C., E. Belding-Royer, and S. Das, *RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing*. 2003. http://rfc.sunsite.dk/rfc/rfc3561.html.

20. Paul, K. and D. Westhoff, *Context Aware Detection of Selfish Node in DSR based Ad-hoc Network*, in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*. 2002: New Orleans, LA. p. 90-100.

21. Albers, P., et al., *Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches*, in *Proceedings of the First International Workshop on Wireless Information Systems (WIS-2002)*. 2002. p. 1-12.

22. Rafique, K., *A Survey of Mobile Ad Hoc Networks*. 2002, Columbia University Student Project Report.

23. Clausen, T.H., *MANET Autoconfiguration Standardization*. http://www.lix.polytechnique.fr/hipercom/index.php?option=com_content&task=view&id=126&Itemid=90.

24. Davida, G.I., *Forward error correction with decision feedback*. 1972. p. 117–133.

25. DeSimone, A., C. Mooi Choo, and Y. On-Ching. *Throughput performance of transport-layer protocols over wireless LANs*. in *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM '93., IEEE*. 1993.

26. Papanastasiou, S., M. Ould-Khaoua, and L.M. Mackenzie. *On the evaluation of TCP in MANETs*. in *International Workshop on Wireless Ad-hoc Networks*. 2005. London.

27. Zhu, J. and Z. Niu. *A reliable TCP-aware link layer retransmission for wireless networks*. in *Communication Technology Proceedings, 2000. WCC - ICCT 2000. International Conference on*. 2000.

28. Christina, P. and J.J. Garcia-Luna-Aceves, *Improving TCP performance over wireless networks at the link layer.* Mob. Netw. Appl., 2000. 5(1): p. 57-71.

29. Kartik, C., et al., *A Feedback Based Scheme for Improving TCP Performance in Ad-Hoc Wireless Networks*, in *Proceedings of the The 18th International Conference on Distributed Computing Systems*. 1998, IEEE Computer Society.

30. Gavin, H. and V. Nitin, *Analysis of TCP performance over mobile ad hoc networks,* in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. 1999, ACM: Seattle, Washington, United States.

31. Yoo, J. and C. Kim. *On the hidden terminal problem in multi-rate ad hoc wireless networks*. in *ICOIN'05*. 2005. Springer, Heidelberg.

32. Xu, S. and T. Saadawi, *Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?* Communications Magazine, IEEE, 2001. 39(6): p. 130-137.

33. Berqia, A., et al. *Fairness and QoS in Ad-Hoc Networks*. in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. 2008.

34. Dhoutaut, D., *Etude du standard IEEE 802.11 dans le cadre des reseaux ad hoc: de la simulation a l'expérimentation*. 2003: PhD thesis, Institut National des Sciences Appliquées de Lyon (INSA de Lyon), France.

35. *IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput.* IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), 2009: p. c1-502.

36. Li, Z., et al. *Study of IEEE 802.11 Fairness and its Interaction with Routing Mechanism*. in *IFIP MWCN*. 2003. Singapore.

37. Luqing, Y., K.G.S. Winston, and Y. Qinghe, *Improving fairness among TCP flows crossing wireless ad hoc and wired networks*, in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing*. 2003, ACM: Annapolis, Maryland, USA.

38. Thyagarajan, N., et al., *Achieving MAC layer fairness in wireless packet networks*, in *Proceedings of the 6th annual international conference on Mobile computing and networking*. 2000, ACM: Boston, Massachusetts, United States.

39. Xiao Long, H. and B. Brahim, *On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation*, in *Proceedings of the 2nd ACM*

international symposium on Mobile ad hoc networking & computing. 2001, ACM: Long Beach, CA, USA.

40. Zhang, D., B. Zhang, and D. Lu, *Contention Window Based Fairness Backoff Algorithm in Ad Hoc Networks*. IEEE Int. Conference Neural Networks & Signal Processing, Zhenjiang, China, June 8-10, 2008.

41. Kaixin, X., et al., *Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED*, in *Proceedings of the 9th annual international conference on Mobile computing and networking*. 2003, ACM: San Diego, CA, USA.

42. Bensaou, B., W. Yu, and K. Chi Chung. *Fair medium access in 802.11 based wireless ad-hoc networks*. in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*. 2000.

43. Vaidyanathan, A., *ATP: A Reliable Transport Protocol for Ad Hoc Networks*. IEEE Transactions on Mobile Computing, 2005. 4(6): p. 588-603.

44. Xin, Y., *Improving TCP performance over mobile ad hoc networks by exploiting cross-layer information awareness*, in *Proceedings of the 10th annual international conference on Mobile computing and networking*. 2004, ACM: Philadelphia, PA, USA.

45. Fu, Z., et al. *The impact of multihop wireless channel on TCP throughput and loss*. in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*. 2003.

46. Cordeiro, C.D.A., S.R. Das, and D.P. Agrawal. *COPAS: dynamic contention-balancing to enhance the performance of TCP over multi-hop wireless networks*. in *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*. 2002.

47. Fu, Z., et al., *The Impact of Multihop Wireless Channel on TCP Performance*. IEEE Transactions on Mobile Computing, 2005. 4(2): p. 209-221.

48. Sherif, M.E., K. Alexander, and L. Christoph, *TCP with adaptive pacing for multihop wireless networks*, in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. 2005, ACM: Urbana-Champaign, IL, USA.

49. Altman, E. and T. Jiménez. *Novel delayed ACK techniques for improving TCP performance in multihop wireless networks*. in *Proceedings of the IFIP International Conference on Personal Wireless Communications*. 2003. Venice, Italy.

50. Braden, R., *Requirements for Internet hosts - communication layers*. STD 3, RFC 1122, October 1989.

51. Asis, N., et al., *Performance of multipath routing for on-demand protocols in mobile ad hoc networks*. Mob. Netw. Appl., 2001. 6(4): p. 339-349.

52. Marina, M.K. and S.R. Das. *On-demand multipath distance vector routing in ad hoc networks*. in *Network Protocols, 2001. Ninth International Conference on*. 2001.

53. Kaixin, X., M. Gerla, and B. Sang. *How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks*. in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*. 2002.

54. Zhai, H. and Y. Fang, *Distributed Flow Control and Medium Access in Multihop Ad Hoc Networks*. IEEE Transactions on Mobile Computing, 2006. 5(11): p. 1503-1514.

55. Hari, B., et al., *A comparison of mechanisms for improving TCP performance over wireless links*. IEEE/ACM Trans. Netw., 1997. 5(6): p. 756-769.

56. Anastasi, G., et al., *Experimental analysis of TCP performance in static multi-hop ad hoc networks.* Nova Science Publisher, 2007.

57. Shugong, X., S. Tarek, and L. Myung. *On TCP over wireless multi-hop networks*. in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*. 2001.

58. Andel, T.R. and A. Yasinac, *On the Credibility of Manet Simulations*. IEEE Computer Magazine, 2006. 39(7): p. 48-54.

59. Bajaj, L., et al., *GloMoSim: A scalable network simulation environment.* Technical Report 990027. UCLA Computer Science Department., 1999.

60. OPNET, *OPNET Modeler*. http://www.opnet.com/products/modeler/home.html, 2006., 2006.

61. *http://www.tetcos.com.*

62. *http://www.isi.edu/nsnam/ns/.*

63. Jain, R., D.M. Chiu, and W. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems*. DEC Research Report TR-301, 1984.

Appendix A – Publications