THE UNIVERSITY OF CALGARY

Channel Selection Strategies for Multi-Channel MAC Protocols in Wireless Ad-Hoc Networks

by

Ian B. Wormsbecker

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

November, 2005

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Channel Selection Strategies for Multi-Channel MAC Protocols in Wireless Ad-Hoc Networks" submitted by Ian B. Wormsbecker in partial fulfillment of the requirements for the degree of Master of Science.

Supervisor, Dr. Carey Williamson
Department of Computer Science

Dr. Zongpeng Li
Department of Computer Science

Dr. Abraham Fapojuwo
Department of Electrical and Computer Engineering

Nov 16/05

Date

Abstract

Wireless devices using the IEEE 802.11 protocol can create on demand multi-hop ad-hoc networks for information collection or dissemination. Due to the ubiquity of the Internet, the Transmission Control Protocol (TCP) and Internet Protocol (IP) often provide transport and network layer services in these networks. New Medium Access Control (MAC) protocols, such as multi-channel MAC protocols (MCMAC), have been proposed to alleviate the inefficient interactions that can occur between IEEE 802.11 and TCP.

The IEEE 802.11 protocol suffers from several problems when used in conjunction with TCP. The hidden and exposed terminal problems result in significant TCP throughput degradation. Furthermore, contention at the link layer results in retransmissions that increase overhead and degrade performance. MCMAC protocols diminish the effect of the hidden terminal problem and eliminate the exposed terminal problem. Bi-directional MCMAC has been proposed to further reduce contention at the link layer. MCMAC protocols, however, suffer from the multi-channel hidden terminal problem.

This thesis uses *ns-2* network simulations to explore the impact of channel selection techniques in the Bi-Directional Multi-Channel MAC protocol (Bi-MCMAC). Four major channel selection techniques are evaluated: Soft Channel Reservation, Soft Channel Reservation with Randomization, Lowest Channel First, and Random Channel Selection. In general, the Soft Channel Reservation techniques provide higher TCP throughput due to fewer link layer losses. In particular, the Soft Channel Reservation techniques reduce link-layer data frame losses by alleviating the Missed Reservation Problem.

Contents

List of Figures

List of Acronyms

ACK	Acknowledgement
AODV	Ad-hoc On-demand Distance Vector
AP	Access Point
Bi-MCMAC	Bi-directional Multi-Channel MAC
CBR	Constant Bit Rate
CRN	Channel Reservation Notification
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CTS	Clear to Send
DCF	Distributed Coordination Function
DSDV	Destination Sequenced Distance-Vector
DSR	Dynamic Source Routing
FTP	File Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LRED	Link-layer Random Early Detection
MAC	Medium Access Control
MCMAC	Multi-Channel MAC
NAV	Network Allocation Vector
RTS	Request to Send
TCP	Transmission Control Protocol

Chapter 1

Introduction

1.1 Motivation

With the advent of new portable devices and an increasing desire for "any-time anywhere" communication, wireless technology has seen increased deployment. The ability to construct on demand networks for information collection or dissemination has given rise to different wireless technologies that are usable in diverse settings. Educators, research labs, corporate installations, disaster recovery and security services industry, as well as Internet providers, are looking to wireless technology to spur growth and advancements in their own fields.

Wireless technologies have become prevalent in many areas of society. Cellular phone use has become a predominant method of communication, even in third world countries. Satellite television has become a mainstream alternative to cable television. Advancements in radio technologies such as Digital Audio Broadcast have brought CD quality music to portable radios, vehicles, and homes. The wireless technology Bluetooth, specifically designed for short range wireless communication, is now used to provide wireless keyboards and mice for computers, speakers for home entertainment systems, and communication between robotic devices in manufacturing systems. Researchers have looked to wireless sensor networks to monitor and collect data on a wide range of topics such as animal migration, water quality, and tracking forest fires. With the development of extremely small computing devices in the field of nanotechnology, the ability for sensor networks to monitor the human body are even possible. Internet providers are looking to a mixture of high speed wired and wireless networks in order to extend their coverage area. Mesh networks make use of wireless devices with the capability to forward data on behalf of other devices to connect both wired and wireless nodes to high speed networks for access to the Internet.

The IEEE 802.11 standard has become an increasingly popular solution for providing wireless communication capabilities to a wide range of devices. This standard defines a common communication scheme for devices to interoperate at the physical and link layers.

Due to the prominence of the Internet, the Internet Protocol (IP) and Transmission Control Protocol (TCP) are likely to be used as the network and transport layer protocols above the IEEE 802.11 link-layer protocol. Given the broad range of applications available for personal computers that utilize the TCP and IP protocols, as well as the desire for handheld devices to integrate into existing networks, the IEEE 802.11 wireless technology must work favourably with current architectures. Due to the prevalence of TCP on the Internet and the difficulty associated with making drastic changes to it, the IEEE 802.11 protocol must not hinder TCP performance.

TCP performs well when it is used in conjunction with a highly reliable link-layer protocol such as that provided by most wired networks [1]. However, wireless networks, especially ad-hoc wireless networks, are more prone to link-layer errors due to the rapidly changing conditions of the wireless medium. Multi-hop networks, those networks that use intermediate nodes to forward data between a source and a destination, introduce added complexities and interactions between the IEEE 802.11 protocol and TCP [5].

TCP can function poorly when the underlying network uses the IEEE 802.11 protocol [7]. False link-layer failures, excessive retransmission attempts, and the inability of the MAC layer to provide fairness across competing TCP connections can result in poor network performance [2] [4] [11] [33]. TCP's congestion control and retransmission mechanisms do not work well with IEEE 802.11, especially in multi-hop network topologies.

Although TCP / IEEE 802.11 interaction is not optimal, many researchers are working on the protocols in order to improve performance. Some researchers have focused on rebuilding TCP with the knowledge that wireless technology may be used at the physical layer [29]. However, due to the prominence of TCP and the barriers to changing it, much of the work has been on the physical and link layer mechanisms of IEEE 802.11. Pacing mechanisms, attempting to spread out a window of TCP packets in time to alleviate the typical bursty packet transmission of TCP, have been proposed to reduce contention at the link layer. The benefit of this strategy is the ability to implement pacing at either the link layer [4], or transport layer [2]. Other researchers have proposed modifying the wireless physical layer to use a more reliable protocol, as well as power control to reduce interference with other wireless users [13].

In an effort to improve the performance of TCP over IEEE 802.11 multi-hop networks, researchers have examined *multi-channel protocols*. These protocols use many features of the IEEE 802.11 protocol, but augment them to support more than one underlying physical layer channel. The use of multiple channels at the physical layer allows multiple users to transmit and receive data concurrently without interfering with each other.

Multi-channel protocols can improve TCP performance, however their methods differ greatly. Some researchers advocate using multiple wireless network cards per device, therefore allowing a single device to access many of the physical layer channels concurrently [23]. Other researchers, however, have attempted to solve the problems by using a single wireless network card [10] [14] [22] [31].

1.2 Thesis Objectives

Regardless of the technique used, some fundamental problems must be solved in order to improve TCP performance. First, the hidden terminal problem must be alleviated. Second, the exposed terminal problem, which adversely affects TCP throughput and fairness, must be counteracted. Third, channel contention must be reduced. A fourth problem, specific to the multi-channel protocols, is the *multi-channel hidden terminal problem*, also known as the *missed reservation problem*. This thesis will analyze this problem in detail.

These fundamental problems will be examined in the context of multi-channel multi-hop wireless networks. The following research questions are the focus of this thesis:

- Can multi-channel MAC protocols solve the hidden terminal and exposed node problems?

- How does the number of physical layer channels affect TCP performance?
- How significant is the impact of the missed reservation problem on TCP performance?
- Can the missed reservation problem be alleviated by careful channel selection?
- Do channel selection strategies have a significant impact on TCP performance?
- How do TCP parameters affect network performance?

1.3 Thesis Outline

The remaining chapters of this thesis are organized as follows. Background and related work are discussed in Chapter 2. An introduction to the IEEE 802.11 MAC layer and multi-channel MAC layer protocols are described in detail in this chapter. Chapter 3 discusses the simulation methodology, including modifications to the *ns-2* simulator, protocol models, and channel selection techniques used. It also describes the experimental design and performance metrics used in the rest of the thesis. Chapter 4 discusses simulation validation. Chapter 5 provides a detailed analysis of simulation results for three MAC layer protocols and the four channel selection techniques. The effects of the number of channels and TCP parameters are also explored. Chapter 6 summarizes the conclusions and contributions of this thesis.

Chapter 2

Background and Related Work

Wireless networking, in particular 802.11, is a growing technology with deployments in many scenarios. Installations in homes and small offices, as well as larger corporate deployments, are becoming commonplace. Wireless technology is being utilized in a wide variety of applications such as communications equipment for disaster recovery, sensor and mesh networks, and as a "last mile" access technology. Regardless of the use, users expect the same performance characteristics as in wired networks. This chapter introduces the IEEE 802.11 wireless technology and explores some of the methods used to improve its performance.

2.1 Multi-hop Wireless Ad-Hoc Networks

Wireless networks can operate in two different modes: infrastructure mode and ad-hoc mode.

Infrastructure based wireless networks rely on a single gateway to connect the wireless users to a wired network, or with each other. This scenario is typically deployed for home and corporate users. Any user wishing to participate on these networks must associate with a wireless access point (AP). All traffic between users must be processed by this access point. If multiple locations need wireless access, a series of access points can be provided through which all users access other network resources. If a user decides to change locations, it must find a new wireless access point with which to associate itself in order to continue utilizing these network resources. Figure 2.1 shows an example of an infrastructure based wireless network where Node A communicates with Node B via the AP.

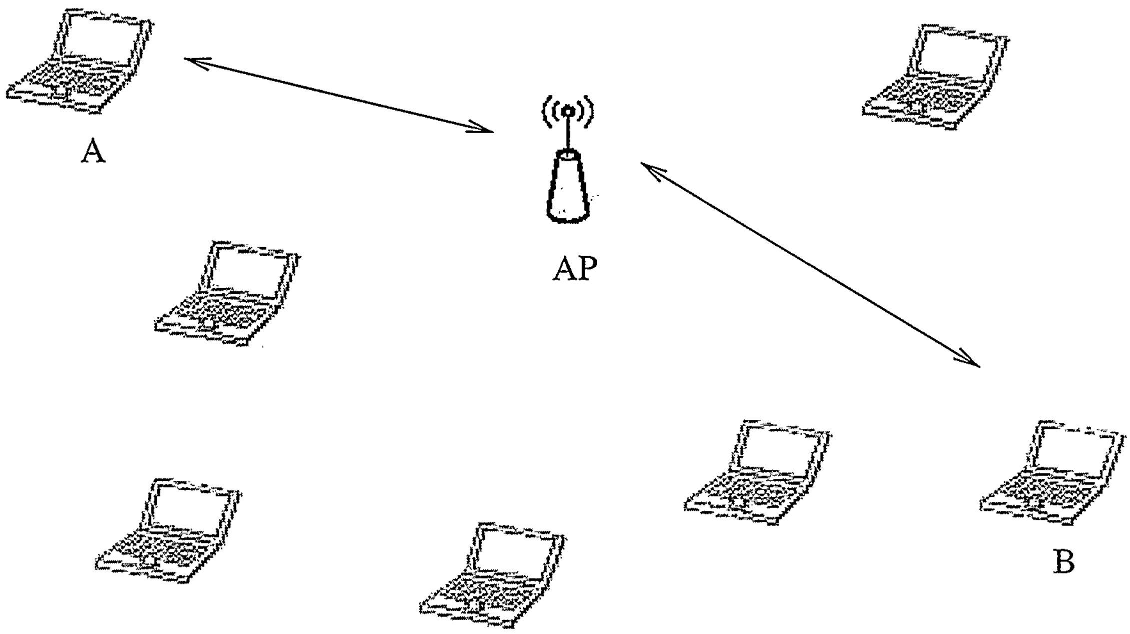


Figure 2.1: Infrastructure mode wireless network - Communication path between node A and B via a Wireless Access Point.

Ad-hoc wireless networks are a decentralized system whereby each wireless user can communicate directly with other wireless users. Furthermore, in order for a user at one location to talk to a user that is located outside of its radio range, it may require intermediate nodes to forward data on its behalf. Scenarios where this is done are known as multi-hop ad hoc networks. As well, ad hoc networks allow user mobility through the use of specialized routing protocols. The dynamic nature of these networks makes them ideal for military deployments, disaster recovery, and sensor networks where the ability to set up a complex infrastructure does not exist or the network may be short-lived. Figure 2.2 shows a multi-hop wireless ad-hoc network where Node A communicates with Node B via intermediate nodes.

Regardless of the mode of operation, wireless networks have other important physical layer characteristics related to wireless signal propagation. A wireless signal, when transmitted, can be successfully received if its signal strength is strong enough at the receiver. Although many factors contribute to the signal strength at a receiver, each transmitter basically has a finite *transmission range*. Any nodes

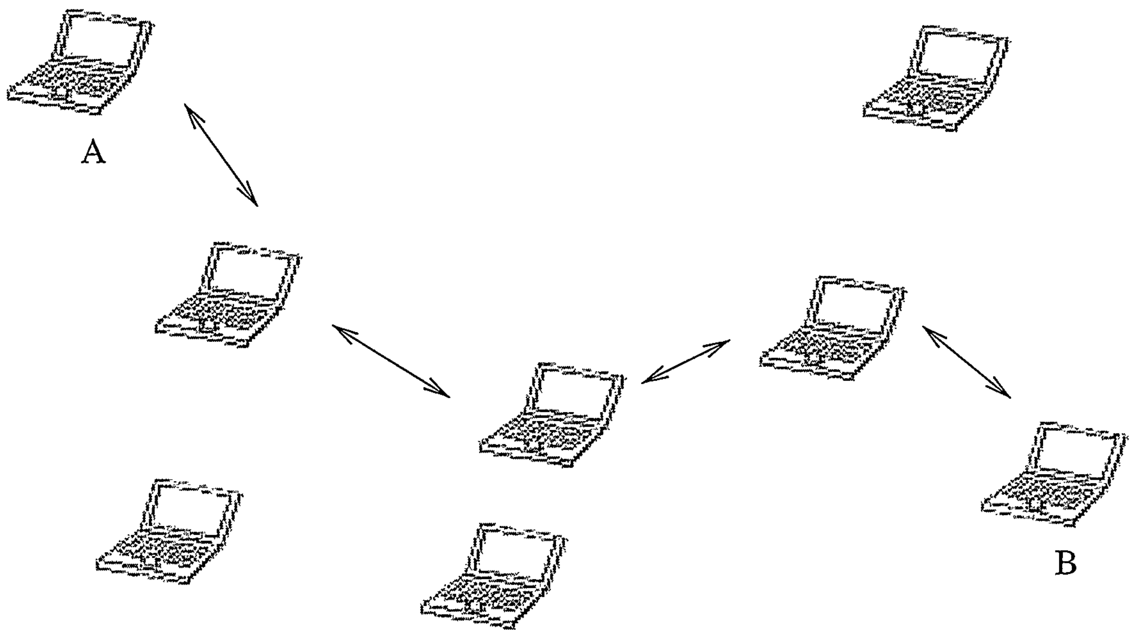


Figure 2.2: Ad-Hoc mode wireless network - Communication path between node A and B via intermediate nodes.

within this transmission range are likely to receive a wireless signal successfully, whereas any nodes outside the range are unable to do so.

Another important property is the *carrier sensing range.* Although a node may be outside of the transmission range and unable to receive data successfully, it may be able to detect that some other node is using the wireless channel. This is known as the *carrier sensing range.* As will be described later, the carrier sensing range is important when attempting to transmit data on a wireless channel.

Lastly, *interference range* is the range at which a node causes interference when transmitting. If a node is attempting to receive data that is within the interference range of a different sender, there is potential for signal degradation, possibly resulting in frame losses.

Simulations often assume that the interference range is the same as the carrier sensing range. Figure 2.3 displays a node S that is transmitting. Node B represents a node that is able to receive data correctly from node S because it is within its transmission range. Node C is unable to receive data correctly from S, but is able to detect that a transmission is taking place because it is within the carrier sensing range. Node D is outside the carrier sensing range and is unable to determine that a transmission is taking place. Furthermore, node D, if it were to receive data from another node, would not be interfered with by the transmission of node S.

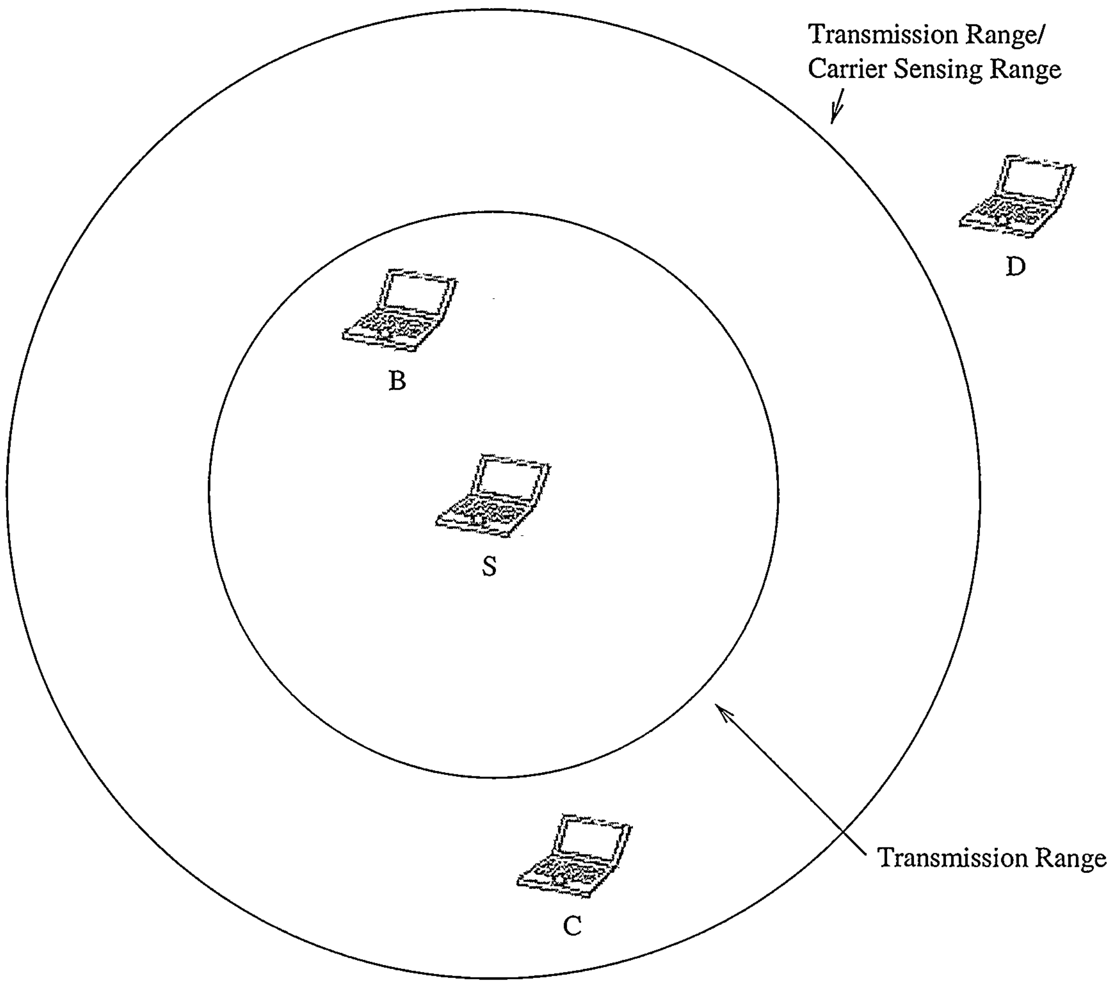


Figure 2.3: Wireless transmission ranges. Node S is transmitting data.

2.2 IEEE 802.11 MAC Protocols

Both infrastructure mode and ad-hoc networks can utilize the IEEE 802.11 Medium Access Control (MAC) protocols [9]. MAC protocols are concerned with controlling access to the physical transmission medium so that multiple stations are not simultaneously transmitting on the medium. When concurrent transmissions occur the overlapping transmissions may propagate to the receiver resulting in a garbled signal. This is known as a frame collision. The IEEE 802.11 MAC protocols provide two methods for reducing frame collisions: a mandatory physical-carrier sensing mechanism, and an optional virtual-carrier sensing mechanism.

The physical-carrier sensing mechanism of the IEEE 802.11 MAC protocol is called the Distributed Coördination Function (DCF). It utilizes a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme in order to reduce the likelihood that overlapping transmissions occur. Carrier Sensing implies that a transmitter will first attempt to detect the presence of a signal from another node prior to attempting to transmit. If the node doesn't detect a signal it will then begin its transmission. However, if a signal is detected, the node will wait a random amount of time, known as the backoff time, and then repeat the CSMA/CA scheme. Note that if two transmitter-receiver pairs are separated by a large distance then both may be able to use the wireless channel at the same time if their carrier sensing and interference ranges do not overlap.

An important factor in the CSMA/CA scheme is the backoff time used when the physical-carrier sensing mechanism detects that the channel is in use. In order to avoid any synchronization between nodes deferring, randomization is used in the backoff time calculation.

In addition, the IEEE 802.11 MAC protocol can utilize a virtual-carrier sensing mechanism that employs a Request To Send / Clear To Send (RTS/CTS) scheme to further reduce the possibility of conflicting transmissions by neighbouring nodes. A transmitting node with a large data frame to send will first use physical-carrier sensing mechanism to transmit a short RTS control frame to the intended destination node. Upon receiving an RTS frame, the destination node transmits a CTS frame back to the node that initiated the transmission. At this point, the transmitting node can begin communicating data to the destination node. Once this data frame has been transmitted successfully, the receiver responds with an Acknowledgment (ACK) frame. The virtual-carrier sensing relies on a *duration* field in the RTS and CTS header that is used to indicate the length of time required to transmit the data and ACK frames between neighbouring nodes.

Any node able to overhear the RTS / CTS exchange knows to defer access to the wireless channel for the specified period of time. The Network Allocation Vector (NAV) is used to track future channel usage with information from the *duration* field from the RTS or CTS frames. The NAV is updated only if the new value will be greater than the current value, and only if the frame being received is not destined for itself. Once the NAV expires, access to the channel can be attempted.

The RTS / CTS scheme reduces the effects of the hidden terminal problem. The hidden terminal problem occurs when two nodes beyond the carrier sensing range of each other attempt to send data simultaneously to a node that is within interference range of both nodes. Figure 2.4 shows an example of the hidden terminal problem. Node A begins to transmit data to node B. Shortly after, node C performs virtual carrier sensing and, because it cannot hear node A due to it being outside the carrier sensing range, also tries to communicate with node B. Thus, Node B receives wireless signals simultaneously from both Node A and Node C, resulting in a frame collision.

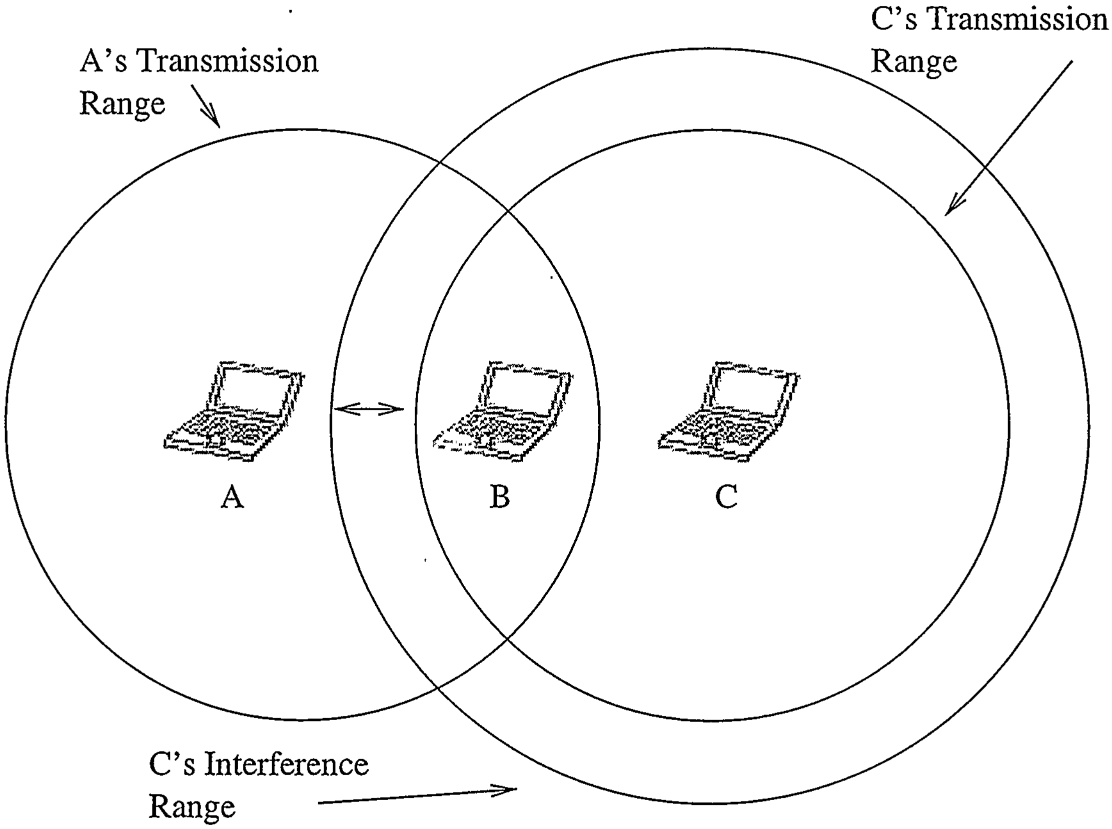


Figure 2.4: The Hidden Terminal Problem - Node B is unable to receive data from A because C causes interference.

By utilizing the RTS / CTS scheme, the hidden terminal problem can be handled. Both Node A and Node C will first be required to transmit an RTS frame to Node B in order to begin a data exchange. Node B, upon receiving an RTS from Node A, will transmit a CTS frame back to Node A. Node C will also receive this frame, and know to defer access to the physical channel for a period of time. The RTS / CTS scheme therefore reduces data packet loss. However, the RTS and CTS frames themselves are still susceptible to the hidden terminal problem. For example, Node A and C could concurrently transmit an RTS frame to node B, resulting in neither frame being received correctly. Retransmission of RTS frames after a random backoff time can resolve this problem.

A similar problem is that of the exposed terminal problem depicted in Figure 2.5. This problem manifests itself when a node near a transmitting node believes it can neither send nor receive due to ongoing communication between other neighbouring nodes. In this case, node D cannot receive data from node C due to the communication from Node B to Node A. Node C will perform physical carrier sensing and determine that Node B is transmitting, therefore deferring its transmission. However, Node C can transmit data to Node D without any data loss occurring because Node B's interference range does not extend to Node D. Likewise, Node C's interference range does not extend to Node A.

Unfortunately, the RTS / CTS scheme does not alleviate the exposed terminal problem. Prior to sending an RTS frame, a node is required to perform physical carrier sensing. The backoff mechanism when the physical channel is busy will result in Node C continually backing off. Node B will have a short backoff value and have a much higher chance of acquiring the physical channel in order to transmit data to Node A. This results in Node C having a lower opportunity to access the channel when Node B has a lot of data to transmit to Node A.

In an ad-hoc network, packets traverse a chain of intermediate nodes from a source node towards a destination. This results in MAC contention as a given node in the chain is attempting to forward a data packet to the next hop in the chain, as well as receive packets from the previous hop in the chain.

This MAC contention, hidden and exposed terminal problems, and RTS/CTS overhead can result in severe degradation of network performance. It has been shown that an ideal MAC protocol can achieve $\frac{1}{4}$ utilization on a chain topology given an ideal MAC scheduler [17]. This utilization is derived from the fact that

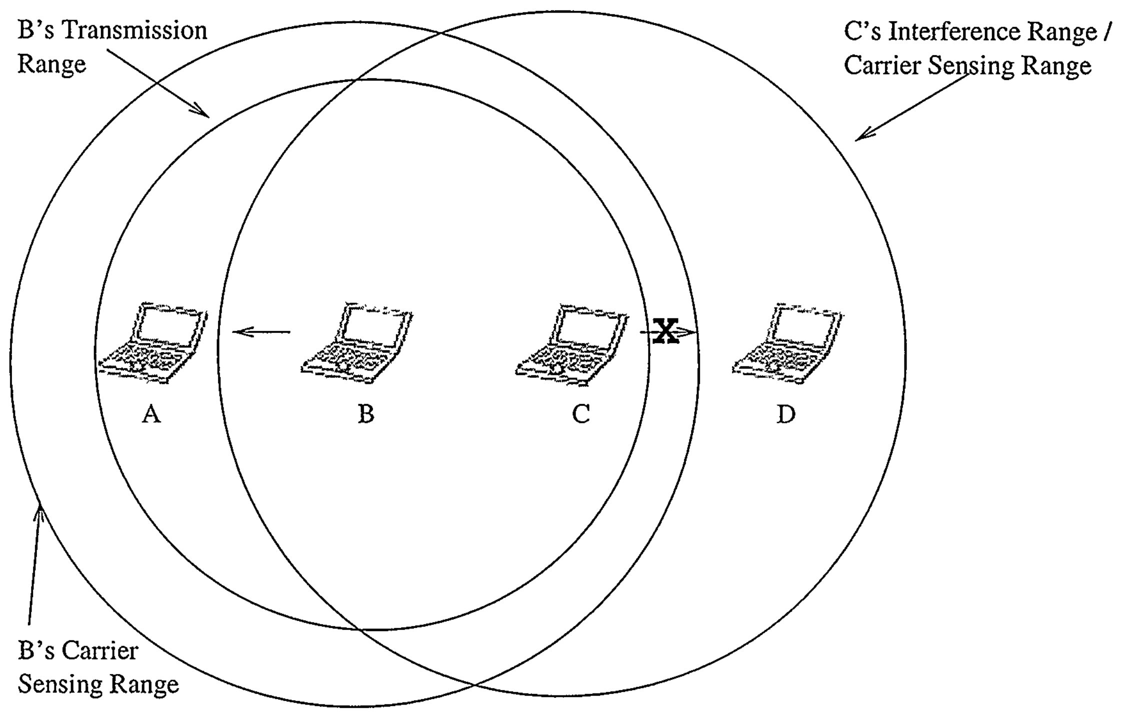


Figure 2.5: The Exposed Terminal Problem - Node C will not attempt transmission to Node D due to communication between node A and node B.

only nodes that are four hops apart in the chain can transmit at a time, resulting in 3 nodes in the backoff phase for every single node that is transmitting. Therefore, the ideal throughput for IEEE 802.11 MAC protocol is $\frac{C}{4}$, where C is the transmission rate of nodes in bits per second. However, simulations and experiments have shown that actual utilization on a chain topology is closer to $\frac{1}{7}$ due to MAC contention, the exposed terminal problem, RTS / CTS overhead, and the inability of the MAC protocol to achieve the optimal schedule [17].

2.3 Multi-Channel MAC Protocols

A Multi-Channel MAC (MCMAC) protocol is an extension of the IEEE 802.11 MAC with more than one physical channel. By utilizing more than one channel, it is expected that throughput gains can be made by allowing multiple transmissions to occur simultaneously. Because these simultaneous transmissions are occurring on different wireless channels, and there is no inter-channel interference, the number of frame collisions can be reduced.

Although many variants of these protocols exist [10] [21] [22] [23], this thesis utilizes a generic model of the multi-channel MAC protocols found in the literature [14]. The protocol uses multiple (K) physical layer channels, where one of the K channels is a control channel, and the remaining K - 1 channels are data channels. All channels have the same data rate and the same physical-layer characteristics. Furthermore, a node maintains a list of the available and free channels.

In order for a node to transmit data to another node, a channel negotiation phase must first occur. During this phase, the two nodes involved in the data transmission must agree on a data channel to use. Once a data channel has been selected, the nodes switch to the data channel for actual data transmission to occur.

The channel negotiation phase functions similarly to the regular IEEE 802.11 MAC, with a few small exceptions. In order to facilitate channel selection, the RTS and CTS frames each have an additional field. In the RTS frame, a bitmask field, *AvailableChannels*, signifies the sender's set of available data channels. In the CTS frame, an integer field, *SelectedChannel*, signifies the receiver's selected data channel on which transmission should occur.

The full protocol is described as follows. The transmitting node constructs an RTS frame with the *AvailableChannels* bitmask set according to the data channels that it thinks are available. It then transmits this RTS frame to the destination node. The destination node, upon reception of the RTS frame, determines which data channels it thinks are available. It then performs an intersection of its available data channels with those from the RTS frame. If no data channels are available after calculating the intersection, the RTS frame is ignored. If, however, there is at least one data channel available, the destination node selects a data channel. It sets the *SelectedChannel* in a CTS frame and sends it to the transmitting node. Upon completion of the transmission, it will tune its radio receiver to the selected data channel and prepare to receive the data. When the transmitting node receives the CTS frame, it will also tune its radio receiver to the channel specified by the *SelectedChannel* in the CTS frame.

At this point, both the sender and receiver have tuned to the selected data channel, leaving the control channel free for other nodes to perform channel negotiation. The transmitting node then sends a data packet to the receiving node. Upon successful reception of the data frame, the receiving node transmits an ACK frame signifying the successful reception to the transmitting node. Both nodes then tune their radio channels back to the control channel. The frame exchange is now complete, and both nodes are free to communicate again. Only one data frame is allowed to be transmitted in order to prevent a pair of nodes from monopolizing a data channel for a prolonged period of time.

Although the sender and receiver are aware of the data channel in use, other nodes can also acquire this information if they are able to overhear the RTS / CTS exchange. Upon hearing a CTS frame that is not destined for itself, a node removes the *SelectedChannel* from its own available channel list. Furthermore, by having a NAV timer for every channel, it is able to add the channel back to its available channel list upon expiry. This reduces the chance that two pairs of nodes select the same data channel.

The MCMAC protocol is able to reduce the hidden terminal problem and avoid the exposed node problem. By using the RTS / CTS mechanism, all nodes in the neighbourhood are aware of a data transmission between two nodes, and defer access to the data channel they selected. As was the case with the IEEE 802.11 MAC, the RTS and CTS frames are still susceptible to the hidden terminal problem. The exposed node problem, however, is solved by the use of multiple channels. When a pair of nodes begin transmitting data on a data channel, the nodes that would have been susceptible to the exposed node problem are able to perform channel negotiation on the control channel. Multiple channels make it impossible for two nodes to monopolize both the control channel and the data channel at the same time (assuming neither are acting maliciously).

Although the MCMAC protocol as described addresses many problems with the IEEE 802.11 MAC protocol, it also introduces new ones. The first problem is a result of the less efficient virtual-carrier sensing in the MCMAC. The RTS frame only carries a list of possible channels for use, while the CTS frame contains the actual channel selection. Any nodes receiving only the RTS frame are unable to set their NAV timer for the (unknown) data channel that is selected, and as a result may later select a conflicting data channel. If two nodes within interference range use the same data channel, frame collisions can occur just as in 802.11. For this reason, it has been recommended that the MCMAC protocol be augmented with a channel reservation phase. In this stage, the transmitting node simply transmits a broadcast frame on the control channel that informs all nodes able to receive it about which channel will be used, as well as the duration for which it will be used. This frame is known as a Channel Reservation Notification (CRN) frame. Any node receiving this frame can then update its available channel list with the information contained in the CRN. Utilizing this mechanism, the nodes that receive the CRN frame have duration information regarding how long the selected data channel will be used and can update their NAV timers accordingly. Once this time expires, these nodes can again add this data channel to their list of available channels to use.

A second problem arises when a node is transmitting on a data channel and a pair of adjacent nodes perform channel negotiation, resulting in the data transmitting node missing information about the newly selected data channel. This is known as the *Missed Reservation Problem* [14] [31]. Analyzing this problem is a large contribution of this thesis, and is discussed in detail in Chapter 5.

The MCMAC protocol is able to increase the utilization of a chain topology. A large factor in the increase in throughput is an increase in chain utilization to $\frac{1}{2}$ as the contention distance is reduced from 3 in the IEEE MAC protocol to 2 in the MCMAC protocol. The MCMAC protocol alleviates Data-Data contention (i.e, contention between multiple data frames destined for a receiver). However, Data-ACK contention occurs for protocols such as the Transmission Control Protocol (TCP) that use acknowledgments at the transport layer. Receiver-destined TCP data packets have to compete for wireless resources with sender-destined TCP ACK packets. Simulations have shown that the MCMAC protocol provides a 47.5% throughput advantage over the IEEE 802.11 MAC protocol for large chain topologies [14].

2.4 Bi-Directional Multi-Channel MAC Protocol

The Bi-Directional Multi-Channel MAC Protocol (Bi-MCMAC) extends the Multi-Channel MAC protocol to facilitate two data packet exchanges in opposite directions utilizing only one RTS / CTS / CRN exchange [14]. Like the MCMAC Protocol, it utilizes K physical channels. One of the K channels is used as a control channel while the other K - 1 channels are used as data channels. The protocol also uses both the channel negotiation phase and channel reservation phase for channel selection and notification. Where it differs from the MCMAC, however, is that during the data transmission phase, the two nodes have the option of each transmitting a data frame, therefore providing a bi-directional frame exchange mechanism.

The channel negotiation phase is the same as in the MCMAC Protocol. In order to perform a data transmission using the Bi-MCMAC Protocol, a node must first transmit an RTS to the intended receiver. Again, like the MCMAC Protocol, available channel information is sent in the RTS frame via the *AvailableChannels* field. Upon receiving the RTS frame, the receiver will perform an intersection of its available channels with those it received in the RTS frame. It then makes a channel selection from the intersection of these channel sets and transmits this selection in the *SelectedChannel* field of a CTS frame. At this point, the receiving node can now tune to the selected data channel and await the data transmission. The transmitting node, however, enters the channel reservation stage of the protocol. After transmitting the CRN frame, the sending node can then tune to the data channel that was specified in the CTS frame. Both the sending and receiving nodes are now tuned to the same data channel, and can commence the data transmission stage of the protocol. The other nodes are now free to utilize the control channel in order to begin the control phase in an effort to transmit data frames.

The data transmission stage of the protocol facilitates the sender transmitting a data frame to the receiver, and potentially, the receiver transmitting a data frame to the sender. The sender will begin the data transmission stage by transmitting its data frame to the receiver. Upon successful reception of the data frame, the receiver has two possible choices. If the receiver has no data frames that are destined for the sender, it will respond as in the multi-channel MAC protocol with an ACK frame. If the receiver does have data it desires to transmit to the sender, however, it will transmit that data frame. In this case, the ACK for the original inbound data frame is "piggy-backed" with the outbound data frame. If this second data frame is received successfully by the original sending node, then it will respond with an ACK frame in order to complete the data transmission phase of the protocol.

The Bi-MCMAC protocol alleviates the hidden terminal problem through its use of the RTS / CTS mechanism. Again, like the MCMAC protocol, RTS and CTS frames are still susceptible. It also alleviates the exposed node problem through its use of multiple channels. The CRN frame, however, reduces the chance that two pairs of nodes select the same data channel, because the nodes that were able to receive the RTS should also receive the CRN (assuming they have not changed locations). They can then update their NAV timers for the selected data channel.

Unfortunately the Bi-MCMAC protocol does not solve the missed reservation problem. Nodes busy transmitting on the data channels are unable to determine which channels have been selected on the control channel during the duration of their data exchange.

Based on simulations, the throughput for the Bi-MCMAC protocol is 67.1% higher than the IEEE 802.11 MAC protocol on a chain topology utilizing TCP at the transport layer. This is better than the MCMAC protocol's 47.5% advantage over IEEE 802.11 MAC protocol. The Bi-MCMAC protocol's increase can be attributed to its reduction of TCP Data - ACK contention. The bi-directional mechanism permits TCP ACK packets to traverse the chain in the reverse direction with lower control channel overhead than the MCMAC protocol. Furthermore, the Bi-MCMAC protocol has been shown to provide better fairness than both the IEEE

802.11 and MCMAC protocols [14].

2.5 Related Work

This section describes research that is being carried out in an effort to improve IEEE 802.11 performance.

Multi-channel MAC protocols have provided substantial performance gains in wireless networks. Nasipuri and colleagues have extensively studied multi-channel protocols in wireless networks [10] [21] [22] [23]. They first examined a multi-channel system that required the ability to listen on all physical channels at a time [21] [23]. When a packet needs to be sent, the transmitting node selects the channel with the lowest interference for transmission. Because the receiving node is listening on all available channels, it would then be able to receive the data packet. They also compared the soft-reservation channel selection scheme with a random channel selection scheme attributing the soft-reservation schemes higher performance to its ability to "reserve" a channel for data transmission for every node. Next, they utilized the control-and-data channel model to use the RTS / CTS mechanism to allow the receiving node to choose the channel by selecting the channel with the best signal-to-noise ratio [10]. They extended this mechanism to include dynamic channel selection, which attempts to maximize the signal-to-noise ratio at the receiver, and reduce the interference with other neighbouring nodes [22]. It was found that this scheme increases the average throughput of all nodes on a network.

Vaidya and colleagues propose a multi-channel MAC protocol that solves the multi-channel hidden terminal problem using temporal synchronization [28]. Nodes negotiate data channels for use during an "ATIM period" that occurs at specified intervals. Any nodes that do not negotiate a channel for use must wait for the next interval in order to transmit data. Their results show increased throughput for Constant Bit Rate (CBR) traffic. They attribute this advantage to the elimination of the multi-channel hidden terminal problem. Although the proposed protocol requires only one transceiver at each node, timing issues add to the complexity.

Fu et al. [4] have proposed two techniques for improving TCP performance on multi-hop wireless networks. They argue that TCP acheives the highest throughpout when it operates with a congestion window specifically tailored to the length of the chain. In particular, they propose a congestion window of $\frac{h}{4}$ where h is the number of hops in the chain network. TCP, however, does not operate around this congestion window, instead rising much larger. They introduce two mechanisms to improve TCP performance: Distributed Link-layer RED (LRED), and adaptive pacing. By using these methods, they improved TCP throughput by 5% to 30% for different network topologies.

LRED is used to provide an early sign of network overload in an attempt to reduce the rate at which TCP injects packets into the network. They modified the IEEE 802.11 MAC protocol to increase the drop probability of MAC frames when the number of MAC retransmissions increases. By dropping packets with fewer MAC layer retransmissions, less time is spent attempting to transmit a MAC frame. In turn, TCP will reduce the rate at which it introduces data to the MAC layer.

Adaptive pacing at the link layer is used to improve spatial channel reuse. Their technique attempts to coordinate the forwarding nodes between a TCP source and destination. By coordinating these nodes, data traffic, as well as losses, will be balanced amongst the nodes on the network. Furthermore, they propose integrating the LRED and adaptive pacing mechanisms to work in concert with each other. The pacing mechanism can be enabled by LRED when the average number of retransmissions exceeds a threshold. This aims to further increase coordination amongst forwarding nodes when the network load changes.

Another technique for improving multi-hop wireless network performance is via power control. Power control is a method for modifying the transmission in order to reduce the interference range for a given transmission. Jung et al. [13] propose a multi-channel MAC protocol with power control that improves wireless network performance. In particular, their protocol increases channel reuse in a physical area. By manipulating the transmission and interference range, pairs of nodes that would previously not have been able to communicate due to carrier sensing, can now communicate. However, their protocol requires two transceivers: one transceiver to listen constantly on the control channel, and a second for transmitting on a data channel.

2.6 Chapter Summary

Improving the performance of IEEE 802.11 networks has garnered much research attention. MCMAC protocols, in particular, are a promising method for increasing performance of multi-hop wireless ad-hoc networks. Their ability to alleviate the hidden terminal and exposed terminal problem contribute to this performance gain.

The next chapter discusses the simulation methodology used in this thesis to investigate the performance of multi-channel MAC protocols.

Chapter 3

Simulation Methodology

3.1 Simulation Models

The *ns-2* network simulator was used for all simulations in this thesis. *ns-2* is a discrete event simulator that provides support for the simulation of different network scenarios and protocols [8]. Of use in this thesis are the wireless models, the IEEE 802.11 MAC protocol, the TCP NewReno model, an IP model with a static routing protocol called NO Ad-Hoc (NOAH) Routing Agent, and the File Transfer Protocol (FTP) application layer model. The work in this thesis builds on simulator modifications made by Tianbo Kuang in order to support both the MCMAC and the Bi-MCMAC protocols [14]. Kuang also modified the default IEEE 802.11 MAC protocol to follow the standard more closely.

This chapter discusses these changes, the additional modules used, as well as the changes performed to facilitate the channel selection algorithms. Figure 3.1 shows a high level overview of how all of these modules fit together inside *ns-2*.

3.1.1 Wireless Channel Model

The wireless channel model used for all simulations is a modified version of the *ns-2* channel model. It has been altered to provide support for multiple physical layer channels. The center frequency of the control channel is 2.412 GHz, while the data channels are 2.427 GHz, 2.447 GHz, and 2.462 GHz. These channels are predefined in the IEEE 802.11 specification, and have been suggested as non-overlapping [18] [20]. Furthemor, it is assumed that there is no inter-channel interference, and that

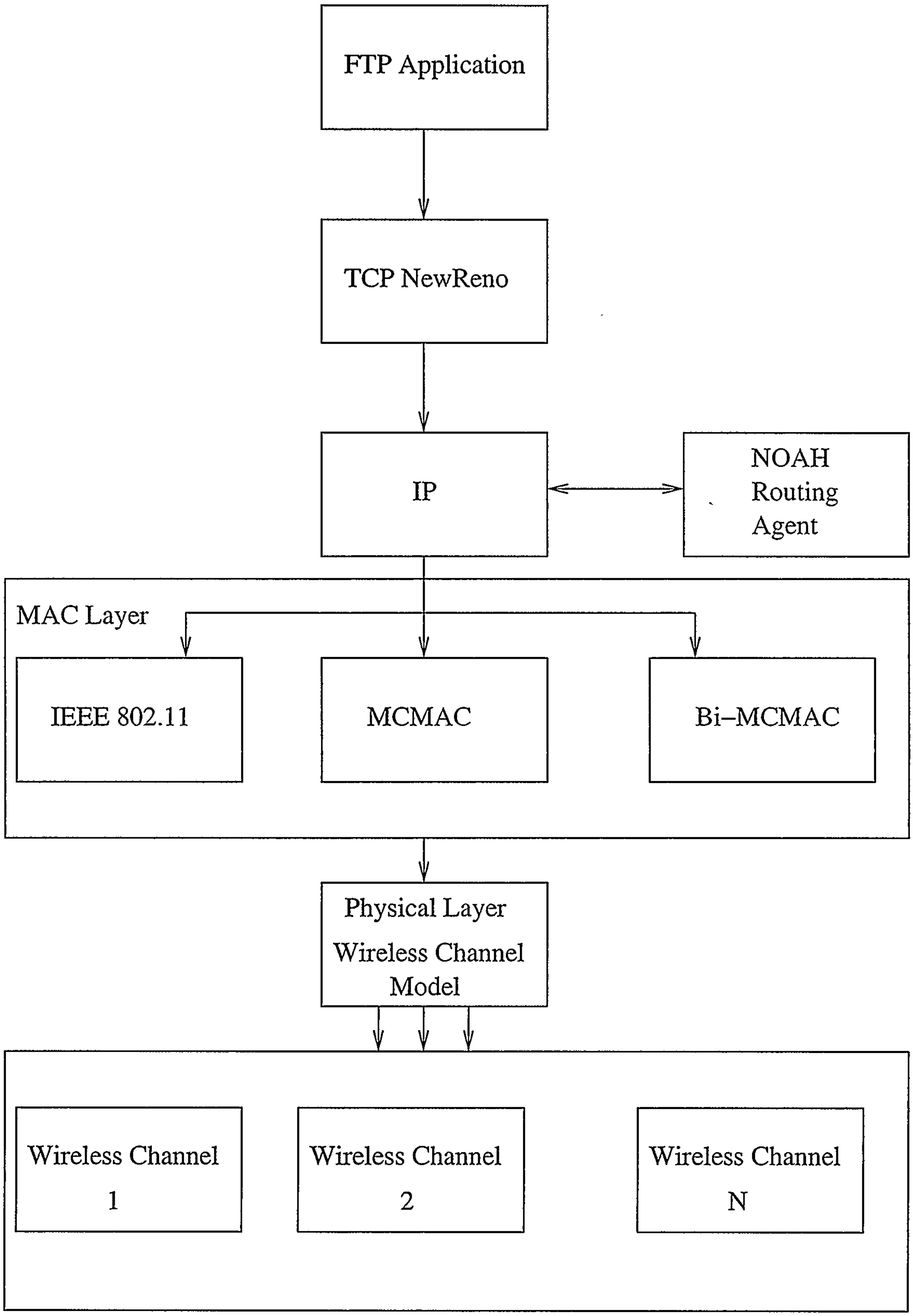


Figure 3.1: High level overview of simulation architecture using ns-2.

all of the channels share the same physical properties.

For the purposes of determining the impact of the number of physical layer channels used, additional channels were defined at 2.482 GHz, 2.502 GHz, and 2.522 GHz. Although these are not defined by the IEEE 802.11 standard, they are sufficient for analyzing the effects of increasing the number of data channels used.

Two-ray ground reflection is used as the wireless propagation model. This model considers a direct path from the transmitter to the receiver, as well as a path that is reflected off the ground [27]. This is the common propagation model used in the literature for analyzing 802.11 MAC protocol performance.

3.1.2 MAC Protocol Models

Three MAC protocols, as described in Chapter 2, are used for this thesis. All three of these protocols are modifications to the default *ns-2* IEEE 802.11 MAC protocol. This section details the simulation implementation of these protocols.

The default 802.11 protocol in *ns-2* does not strictly adhere to the IEEE standard. Modifications were made, specifically to the backoff mechanism in the physical carrier sensing stage, in order to respect the standard. The default *ns-2* 802.11 protocol utilizes a fixed backoff value that can result in two nodes having synchronized backoff times. This synchronization leads to performance problems as both nodes are continually in the backoff states. The appropriate random backoff calculation, $BackoffTime = Random() \cdot aSlotTime$ was utilized. This modified 802.11 protocol was then used as the basis for the MCMAC and Bi-MCMAC protocols.

In order to facilitate both the Multi-Channel MAC protocol and the Bi-Directional Multi-Channel MAC protocol, the 802.11 MAC protocol was modified to support multiple physical layer channels. This required the addition of an array of timers for the NAV, one for each channel. The timers are updated upon reception of a CTS frame. Both protocols require the *AvailableChannels* field in the RTS frame and the *SelectedChannel* field in the CTS frame. They also make use of the reservation phase described in Section 2.3. The addition of the CRN frames with the *SelectedChannel* field is used for this reservation phase. The bidirectional mechanism is also implemented in the Bi-MCMAC protocol.

The simulation output trace file format was altered to portray the changes to the multi-channel MAC protocols. An example of the modified format is shown

Table 3.1: Simulation trace output description

Column	Description	Options
1	Event Type	s: packet transmission r: packet reception D: packet dropped
2	Simulation time	
3	ID of node where event occurred	
4	Protocol where event occurred	
5	Reason why the event took place	
6	Packet type ID	0: MAC frame 6: TCP packet
7	Packet Type	MAC frames: RTS, CTS, CRN, ACK TCP pkts: tcp, ack
8	Packet size n bytes	
9	Expected transmission time (hex)	
10	Receiving node ID	
11	Sending node ID	
12	MAC Type details	
Additions		
13,14	Channel used for transmission	
15,16	Transmission time for packet (seconds)	
17,18	Data Channel Information	
	cs	Candidate set of channels proposed (RTS only)
	use	Channel selected for use (CTS only)

in Figure 3.2. The additions include the channel that was used to transmit or receive the given frame, the amount of time spent transmitting this frame on the channel, and some frame-specific information for RTS and CTS frames. The RTS frame contains extra information to inform the receiving node of the candidate set (cs) of data channels it believes are available. The candidate set is represented as a bitmask of the channels that are already in use, with the first data channel as the most significant bit in the mask. The CTS frame contains the data channel selected for use (use). Table 3.1 describes each column of the trace file.

In order to provide a simple method for testing various channel selection strategies, a generic interface was provided. Currently, the four channel selection algo-

```
s 100.553054667 _0_ MAC  --- 0 RTS 46 [2708 1 0 0] c 0 t 0.000368 cs {0}
r 100.553423500 _1_ MAC  --- 0 RTS 46 [2708 1 0 0] c 0 t 0.000368 cs {0}

s 100.553433500 _1_ MAC  --- 0 CTS 46 [258e 0 1 0] c 0 t 0.000368 use {1}
r 100.553802333 _0_ MAC  --- 0 CTS 46 [258e 0 1 0] c 0 t 0.000368 use {1}

s 100.563113167 _1_ MAC  --- 0 ACK 38 [0 0 0 100] c 1 t 0.000304
r 100.563418000 _0_ MAC  --- 0 ACK 38 [0 0 0 100] c 1 t 0.000304
```

Figure 3.2: Trace file output format

rithms described in Section 3.4 are implemented. Four simple changes are required to add a new channel selection technique. Once the changes are made and the simulator is recompiled, any of the channel selection strategies can be used. Two files need to be changed: `mac/mac-802_11.h` and `mac/mac-802_11.cc`. The relevant source code snippets are shown in Figure 3.3 and Figure 3.4, respectively. A function pointer (`ScheduleRoutinePTR ScheduleRoutine`) is defined in the Mac802_11 class in the file `mac/mac-802_11.h`. A new prototype can be added for the desired channel selection technique. Next, in order for the simulator to select the new technique, the variable `ScheduleRoutine` must point to the newly added function as defined by the prototype. Third, `mac/mac-802_11.cc` must be altered to provide an implementation of the new technique. A template for a channel selection technique is provided in Figure 3.4. Finally, the channel selection technique must be selected in the tcl code that defines the simulation scenario. Figure 3.5 contains an example where the Random channel selection technique is used.

3.1.3 TCP Protocol Model

The TCP protocol is used to provide reliable end-to-end data transfer. It uses data segments containing sequence numbers to ensure data is received in order and not duplicated. Acknowledgements are used to notify a sender that data has been received correctly. These acknowledgments, or lack thereof, are used by the sender to determine the network conditions between itself and the receiver. Using this information the sender can determine, via flow control and congestion control

```
/* Function pointer type */
typedef void (Mac802_11::*ScheduleRoutinePTR)(Packet *p);

/* Function pointer to be set */
ScheduleRoutinePTR ScheduleRoutine;

/* Implemented channel selection techniques (prototypes) */
void ScheduleSticky(Packet *p);
void ScheduleRandom(Packet *p);
void ScheduleStickyRandom(Packet *p);
void ScheduleLowestChannelFirst(Packet *p);

/* Constructor */
...
 /* Determine the scheduler type to use as set in .tcl file */
 tcl.evalf("Mac/802_11 set schedulerType_");
 if (strcmp(tcl.result(), "Random") == 0)
   ScheduleRoutine = &Mac802_11::ScheduleRandom;
...
```

Figure 3.3: File: mac/mac-802_11.h

```
void Mac802_11::ChannelSelectionTemplate(Packet *p)
{
  int AvailChannels[15];
  int AvailChannelsCount = 0;

  /* Acquire access to RTS packet headers */
  struct rts_frame *rf = (struct rts_frame*)p->access(hdr_mac::offset_);

  /* Retrieve the candidate set of channels from sending node */
  u_char recvmap = rf->rf_channel;

  /* Determine possible available channels given
     candidate set from RTS, as well as this nodes
     free channel list */

  int i;
  for(i=0; i<dataChannels_; i++) {
      if(usedChan_[i].no == 1)
          SET_CHANMAP(i, recvmap);
  }

   for(i=0; i<dataChannels_; i++)
      if(((recvmap >> i) & 0x01) == 0)
        AvailChannels[AvailChannelsCount++]=i+1;

  /* Channel selection
     set channel_ to specify channel */

  channel_ = ... ;

  /* Transmit CTS packet to indicate chosen data channel */
  sendCTS(ETHER_ADDR(rf->rf_ta), rf->rf_duration);

}
```

Figure 3.4: File: mac/mac-802_11.cc

```
# Select channel selection technique
Mac/802_11 set schedulerType_ Random
```

Figure 3.5: Simulation scenario tcl file

algorithms, the rate at which to inject packets to the lower layer network protocols.

TCP NewReno was selected as the transport layer protocol for all simulations. Studies indicate that TCP NewReno is widely deployed on the Internet [19] [24].

Delayed ACKs are a modification to a TCP receivers acknowledgement mechanism [3]. Rather than transmit an ACK for every TCP data packet received, using delayed ACKs a receiver instead ACKs every second TCP data packet. This reduces the contention between TCP data packets flowing in the forward direction and TCP ACK packets flowing in the reverse direction due to the reduced number of ACKs in the network. Delayed ACKs have been shown to improve TCP performance in networks using the IEEE 802.11 MAC protocol [16].

3.1.4 FTP Traffic Model

File Transfer Protocol (FTP) is an application layer protocol for copying one or more files between a client and a server [15]. The simulator contains an implementation of FTP as an application layer agent that is used to provide bulk data to a lower layer protocol. Although the FTP agent ensures there are always packets available to send, TCP controls the rate at which the data are sent, according to its flow and congestion control algorithms.

3.1.5 NOAH Static Routing Protocol

There are several possible choices for the routing protocol in wireless ad-hoc networks. These protocols can have an effect on wireless network performance [6] [17]. Broadcast storms due to routing protocol updates, routing table cache expiration, and route discovery mechanisms all can degrade TCP performance [17] [30]. For this reason, a static predetermined route is established to eliminate all of these problems.

The NO Ad-Hoc (NOAH) Routing Protocol is a wireless routing agent for *ns-2* that only supports direct communication between adjacent wireless nodes. This agent allows static routes to be defined at simulation startup rather than dynamically based on protocols such as Ad-hoc On-demand Distance Vector (AODV) [25], Destination-Sequenced Distance-Vector Routing (DSDV) [26], and Dynamic Source Routing (DSR) [12]. No routing messages are sent during the simulation. This design permits a cleaner analysis of the MAC protocols without routing anomalies such as lost routes or route detection broadcast storms [30].

3.1.6 Network Model

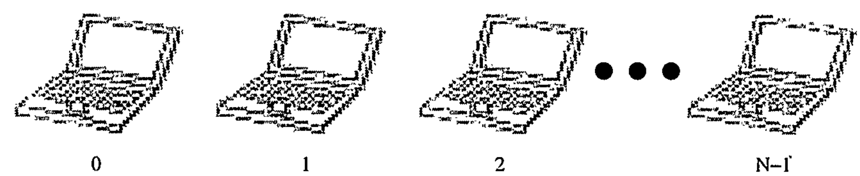


Figure 3.6: Sample Chain Topology.

A chain topology as depicted in Figure 3.6 was used for all experiments. The two endpoints of the chain are used as the TCP sender and receiver. The intermediate nodes in the chain are packet forwarding nodes only; they do not generate any traffic of their own. Data is transmitted in the forward direction, while acknowledgments are transmitted in the reverse direction. An N-chain topology has N-1 hops that a packet must traverse from the sender to the receiver. The two node scenario has direct connectivity between the sender and the receiver. The sixteen node scenario has fourteen intermediate nodes that each packet must pass through. Each node is 250 metres from each neighbour.

3.2 Simulation Instrumentation

An important part of the performance analysis in this thesis is the adequate collection and analysis of simulation trace data. As mentioned in Section 3.1.2, the MAC protocols were modified in order to monitor metrics not previously accessible via standard *ns-2* tracing mechanisms. Furthermore, simulation post processing facilities enable both detailed analysis and an overview of simulation results. Finally, shell scripts were created for executing a large number of simulations.

Three major tools were built for graphically displaying and summarizing simulation results. The first tool works in conjunction with the batch simulation system to extract simulation metrics such as TCP throughput, MAC frame losses, and data channel packet loss rates. Examples of these graphs can be seen in Figure 5.1, Figure 5.2, and Figure 5.7, respectively. These graphs provide an overview of a batch of simulation scenarios.

Second, a tool that creates a graphical representation of all network traffic organized by channel was designed. The output of this tool provides data files that can be viewed in xplot [32]. Xplot allows the user to zoom in and out of a graph. This allows data traffic patterns to be analyzed on different time scales.

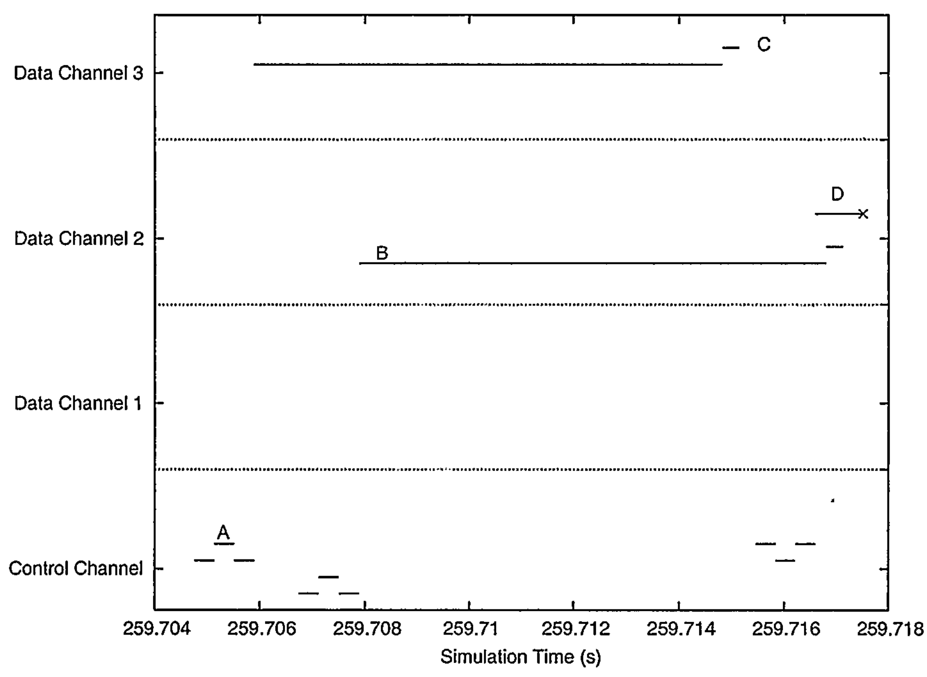


Figure 3.7: Example network traffic graph.

Figure 3.7 displays a network traffic graph. The horizontal axis is time. The vertical axis represents two values: the channel containing a frame transmission, and the node responsible for this transmission. The physical channels are labeled as *Control Channel*, *Data Channel 1*, *Data Channel 2*, and *Data Channel 3*. Each channel can display the traffic for every node on the network.

In the 4 node scenario shown in Figure 3.7, the activity, indicated by the lines on the graph, closest to the X axis for each of the channels denotes transmissions by node 0. The line above shows transmission by node 1. The topmost activity in each channel shows transmissions by node 3, while the activity under it shows transmissions by node 2. An X over any transmission indicates that the frame was lost due to a frame collision.

The control channel only contains RTS, CTS, and CRN frames. The label A denotes a RTS / CTS / CRN handshake, with the RTS frame being the first to be transmitted, followed by the CTS frame, and concluded with the CRN frame. On the data channels, TCP data frames (B) are the lines with the longest duration. The shortest activity on the data channels are MAC ACK frames (C). The medium duration frame transmissions are TCP ACK packets (D).

Finally, in order to analyze packet loss characteristics, a tool that plots the location of packet losses was developed. The output is a histogram representation showing the number of packets discarded due to collision at each node on the network. Figure 5.14 and Figure 5.15 in Chapter 5 are two such graphs.

3.3 Multi-Channel Hidden Terminal Problem

This section examines the multi-channel hidden terminal problem as described by So and Vaidya [28]. Figure 3.8 shows an example of the problem extracted from simulation trace files. This problem is also known as the missed reservation problem [14].

The multi-channel hidden terminal problem occurs when a node is busy transmitting or receiving on a data channel when a neighbouring node initiates a channel reservation handshake. Because a node is active on a data channel, it is unable to learn of the channel its neighbour selected and, in turn, may choose the same channel when it begins its next data exchange.

In Figure 3.8, nodes 2 and 3 perform a successful RTS / CTS /CRN handshake on the control channel (A). Next, these nodes begin communicating via data channel 3 (B). Node 0 and 1 then perform a successful RTS / CTS / CRN handshake

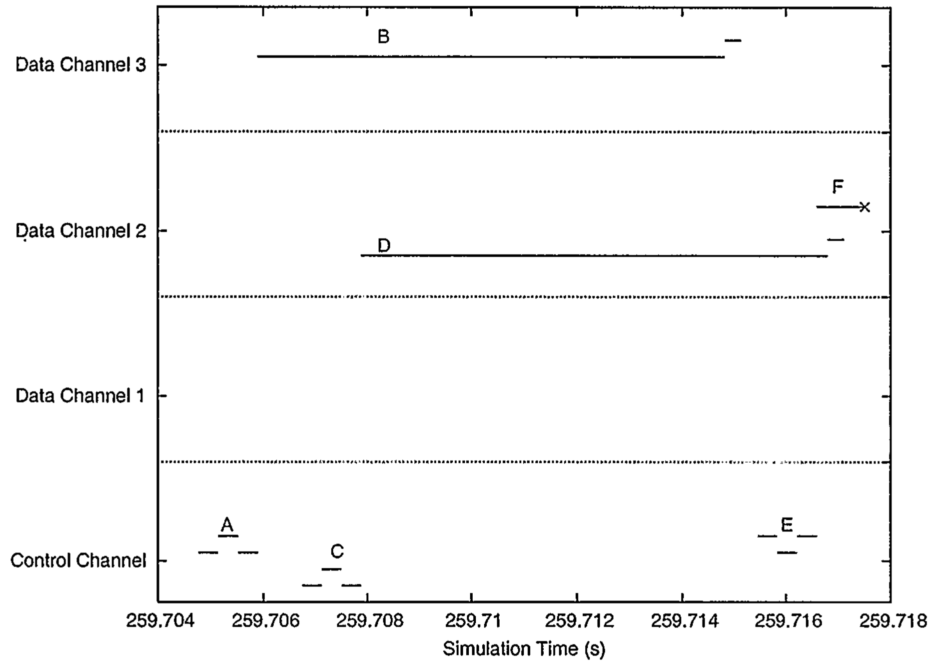


Figure 3.8: Multi-channel hidden terminal problem.

(C), selecting data channel 2 for use. Because node 1 received the CRN from node 2, it did not include data channel 3 in its list of usable channels. It then began its data transmission on channel 2 (D). Once nodes 1 and 2 conclude their data transmission on data channel 3, they perform a successful RTS / CTS / CRN handshake (E). Since nodes 1 and 2 were participating on data channel 3 at the time nodes 0 and 1 chose data channel 2, they are unaware that data channel 2 is in use. When nodes 1 and 2 begin transmission on data channel 2, a frame collision occurs (F).

One potential solution to the multi-channel hidden terminal problem is the use of multiple network transceivers. This would enable a node to listen on the control channel and all data channels at the same time. By doing this, nodes could perform carrier sensing to ensure other nodes are not transmitting on a data channel prior to channel selection. However, using multiple transceivers has drawbacks. One issue is the cost of deploying devices with many transceivers. Second, for small devices such as a PDA or sensor modules, the size of many network transceivers may have a large impact on the practicality of the device. For these reasons, this thesis assumes each node has a single network transceiver capable of half duplex communication.

Channel selection may also have a role in alleviating the multi-channel hidden terminal problem. By intelligently selecting data channels, the probability of falling victim to the multi-channel hidden terminal problem may be reduced. Furthermore, scarce wireless resources may be used more efficiently under different channel selection strategies. This, combined with the reduction of the multi-channel hidden terminal problem, may improve TCP performance.

3.4 Channel Selection Strategies

When using the Multi-Channel MAC protocol or the Bi-Directional Multi-Channel MAC protocol, a channel selection must be made in order to transmit a data frame. In both of these protocols, the node that will be receiving the data determines the data channel to be used. The CTS and CRN frames are used to notify the transmitter and neighbouring nodes of this selection. This section explores the strategies for performing this channel selection.

Channel selection is the responsibility of the receiving node for a given data transmission. The node making the channel selection receives from the transmitting node a list of channels that the sender thinks are available for use. This information, in conjunction with the list of channels that the receiving node thinks are available, are the major factor in determining which channel to use. By performing an intersection between these two lists, the node making the channel selection can select a channel that both nodes believe is available. Using this information, and employing one of the channel selection algorithms described below, a channel for data transmission can be selected.

3.4.1 Simple Channel Selection Strategies

Randomized channel selection is utilized as a baseline channel selection strategy. This technique simply selects a random channel from the intersection of those channels that each node thinks are available. By choosing a channel randomly, it is expected that two transmitting nodes will be unlikely to use the same data channel. Furthermore, it is expected that traffic will be balanced evenly across all data channels.

A second simple channel selection technique, Lowest Channel Available, is also used. This strategy simply selects channels in numerical order, selecting the lowest numbered channel that is thought to be available. If a node is not aware of any current transmissions on a data channel, it will simply select data channel 1.

3.4.2 Soft Reservation Channel Selection

The Soft Reservation channel selection strategy employs a memory scheme to select a channel that last had a successful transmission. By utilizing a channel that was used successfully last, it is hoped that this channel is still available. Furthermore, it is hoped that neighbouring nodes will select differing data channels for short periods of time before choosing a new channel.

If the channel that was last used successfully is not available, two possible solutions exist. The first solution is to select the lowest channel that is available. The second solution is to randomly choose an available channel.

3.5 Experimental Design

This section discusses the simulation settings used for the performance analysis. Fixed simulation parameters, experimental factors, and performance metrics are discussed.

3.5.1 Simulation Parameters

Important simulation parameters used throughout this thesis are as follows:

FTP data: FTP provides unlimited data to the TCP layer for transmission.

TCP NewReno: TCP NewReno is used for all simulations.

Wireless Channel Data Rate: The data rate for all wireless channels is set to 1 Mbps.

Wireless Transmission Range: The transmission range is set to 250 meters.

Wireless Interference Range: The range at which other nodes will cause interference is set to 500 meters.

Wireless Carrier Sensing Range: The range at which nodes can detect someone using the wireless link is 500 meters.

3.5.2 Experimental Factors

This thesis uses a multi-factor approach to experimental design. Each of the following factors are examined:

MAC Protocol: IEEE 802.11, MCMAC, and Bi-MCMAC protocols, as described in Chapter 2 are used.

Channel Selection Technique: Four channel selection techniques are used, as described in Section 3.4.

Chain Length: Chain length is varied from 2 to 16 nodes.

Number of Data Channels: The number of data channels ranges from 1 to 6.

TCP Parameters: TCP parameters are as follows:

Packet Size: Packet size is varied from 64 bytes to 1500 bytes.

Delayed ACKs: Delayed ACKs are enabled or disabled.

3.5.3 Performance Metrics

Multiple metrics are used in the simulation experiments to measure performance. The following metrics are used:

TCP Throughput: TCP throughput is calculated as the number of successfully delivered data bytes from the TCP sender to the destination per unit of time. This metric is expressed in kilobits per second (kbps). A higher throughput value indicates better performance.

MAC Collision Count: MAC collisions are calculated as the number of unicast MAC layer frames that are lost due to interference from competing transmissions. The types of frames susceptible to collisions are: RTS, CTS, Data, and ACK frames. CRN frames are broadcast and are therefore not included in this metric. A lower value of this metric indicates better protocol performance.

MAC Frame Loss Rate: MAC layer frame loss rate is calculated as the ratio between the number of MAC collisions and the number of successfully received MAC layer frames. A lower value indicates better performance.

Data Channel Loss Count: Data channel loss counts are calculated as the number of MAC layer losses that contained TCP Data, TCP ACK, or MAC ACK frames. These frames are transmitted exclusively on a data channel. A lower loss count is better because less time is wasted on a data channel with failed transmissions.

Data Channel Loss Rate: Data channel loss rate is calculated as the ratio of lost data channel packets to the number of successfully transmitted data channel packets. Again, data channel packets include TCP data, TCP ACK, or MAC ACK frames. A lower data channel loss rate indicates less time wasted transmitting on a data channel.

Data Packet per Handshake Ratio: Data packet per handshake ratio measures the number of data packets that are transmitted per RTS / CTS / CRN handshake. 802.11 protocol will have a ratio of 1.0, whereas the Bi-MC MAC will have a ratio between 1.0 and 2.0. The higher the ratio is, the lower is the control frame overhead per data packet transmitted.

3.6 Chapter Summary

The multi-channel hidden terminal problem degrades the performance of multi-channel wireless ad-hoc networks. Channel selection strategies for multi-channel MAC protocols provide the potential for added TCP performance in these networks. By analyzing particular performance metrics such as throughput and packet losses, the benefit of using a given channel selection strategy can be determined. Furthermore, the effect of these strategies on the multi-channel hidden terminal problem can be analyzed.

Chapter 4

Simulation Validation

This chapter discusses the efforts made to ensure the simulation infrastructure is correct. Two main simulator modules need to be validated: the MAC protocols and channel selection techniques. A two node simulation case is run with each of the three MAC protocols in order to ensure their proper functioning. Next, upper bounds are calculated and simulations are performed to ensure these bounds are not exceeded. Third, comparing the simulation results with other results in the literature demonstrates that the general trends are similar. Finally, these scenarios are then run with each of the channel selection techniques and network traffic diagrams are analyzed to verify proper channel selection operation.

4.1 Simple Scenario

Regardless of the MAC protocol chosen, no packet collisions occur when examining a simple two node simulation scenario. Because both nodes are within the carrier sensing range of each other, neither will attempt to transmit while the other is transmitting. When one node is transmitting, the other node must be receiving; given the half-duplex nature of each node, both nodes cannot transmit on the wireless channel simultaneously. Analyzing the MAC collision count for the two node simulation cases shows that there are indeed no collisions for any of the MAC protocols.

4.2 Simple Bounds

By analyzing the overhead associated with transmitting data packets, rough upper bounds for protocol efficiency, and therefore throughput, can be established for each of the three MAC protocols.

Kuang and Williamson [14] show that the protocol efficiency of IEEE 802.11 is 77.8% when using TCP NewReno with delayed ACKs. Their analysis is as follows. In order to transfer a successful 1024 byte TCP data packet, a 40 byte TCP ACK packet must be received acknowledging the transmitted data. The time to transmit a single TCP Data packet with associated RTS, CTS, and ACK frames is 9648 μs, while the time to transmit a single TCP ACK packet is 1776 μs. TCP with delayed ACKs transmits one TCP ACK packet for every two TCP data packets. When TCP reaches a steady state, two TCP packets are sent within three RTS / CTS / ACK cycles, leading to an estimated efficiency of 77.8%. Simulation results for a two node scenario generate a throughput of 753 kbps, which is below the maximum throughput of 778 kbps using a 1 Mbps wireless channel.

A similar method can be used to calculate the maximum throughput for the MCMAC and Bi-MCMAC protocols. Frame sizes are increased to 45 bytes for the RTS and CTS frames due to the added data channel information they contain. Kuang and Williamson [14] show that the upper bound on throughput in a 2 node scenario for MCMAC is 732 kbps. The simulated throughput is 710 kbps. Likewise, the upper bound for the Bi-MCMAC is 784 kbps, while the simulated result is 763 kbps. In both cases, the simulated result is lower than the upper bound calculation.

4.3 Comparison to Literature

Results in the literature for the IEEE MAC protocols exhibit a trend of throughput decreasing as a function of chain length. Li et al. [17] showed that the expected throughput given an ideal MAC schedule for a chain topology is $\frac{C}{4}$, where C is the one hop throughput. However, they note that in reality the IEEE 802.11 MAC is only able to realize $\frac{C}{7}$. Furthermore, they showed that using current hardware resulted in similar results.

Figure 4.1 shows that this trend holds for the IEEE 802.11 protocol. For the

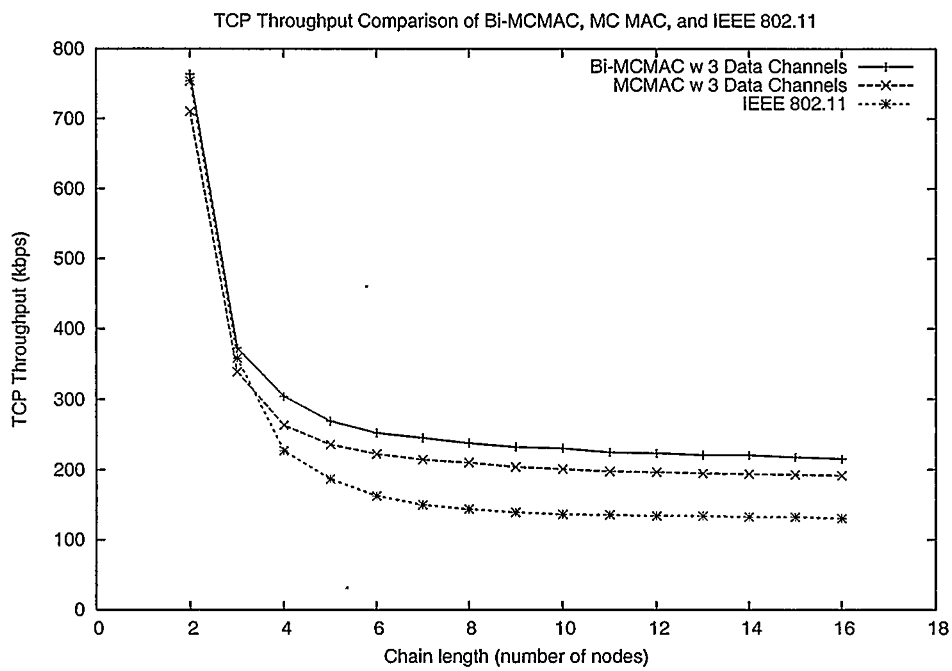


Figure 4.1: TCP throughput comparison of 3 different MAC protocols over varying chain lengths.

Table 4.1: Number of frames transmitted per data channel for 2 node scenario.

Technique	Channel 1	Channel 2	Channel 3
Random Selection	33,123	32,389	32,757
Lowest Channel Available	98,383	0	0
Soft Reservation	98,383	0	0

Table 4.2: Number of frames transmitted per data channel for 4 node scenario.

Technique	Channel 1	Channel 2	Channel 3
Random Selection	26,739	26,511	26,849
Lowest Channel Available	67,073	13,684	0
Soft Reservation	43,010	44,122	0

10 node scenario, the throughput is 136 kbps. This is close to the estimated throughput of 142 kbps.

Multi-channel MAC protocols have also been shown to follow this trend. Kuang and Williamson [14] show that although multi-channel protocols improve throughput, the performance still decreases as chain length increases. They also show that, due to spatial channel reuse and reduced contention, the decrease in throughput is not as severe as 802.11. Simulation results displayed in Figure 4.1 show that throughput for MCMAC is slightly above the IEEE 802.11 protocol. Due to the lower overhead per TCP packet for the bi-directional mechanism, it performs better than the MCMAC.

4.4 Channel Selection Validation

In order to verify that the channel selection techniques function properly, network channel usage was analysed for the simple 2 node scenario. Each simulation case used 3 data channels and a single control channel. Simulations were run using the three channel selection techniques.

Table 4.1 displays the number of packets transmitted on each data channel for the three channel selection techniques for a scenario with 2 nodes. Table 4.2 displays the same information for a 4 node scenario.

The Random channel selection technique should use all data channels equally. At every point where a channel selection is made, all channels will be available for use. Given the random selection of data channel, each channel has the same probability to be chosen. Furthermore, by analyzing the simulation trace file, it was found that each data channel is used with equal probability. Data channel 1 is used 33,123 times, data channel 2 is used 32,389 times, and data channel 3 is used 32,757 times as seen in Table 4.1. Table 4.2 shows that random channel selection uses all three data channels in the 4 node scenario as well.

The Lowest Channel Available channel selection technique should only use a single data channel for the simple two node scenario. Again, because only two nodes exist, no other node can be occupying a data channel. Therefore, the nodes communicating should always use the lowest data channel. Table 4.1 shows that all frames are transmitted on data channel 1, and no frames are transmitted on the other two remaining data channels. In a 4 node scenario, it is expected that the two lowest data channels will be used, and data channel 3 will remain idle. This is due to the ability to have two concurrent data transmissions at a time. Analyzing the channel utilization from the simulation scenario confirms this. Table 4.2 shows that there are 67,073 packets on data channel 1, and 13,684 packets on data channel 2. Data channel 3 has no data packet transmissions.

The Soft Channel Reservation channel selection technique should also only use a single data channel. Again, in the two node scenario there is no competition for data channels, therefore the lowest data channel will always be selected. Like the Lowest Channel Available technique, only the lowest data channel should be used for transmission. As seen in Table 4.1, no frames are transmitted on data channels 2 or 3, rather data channel 1 is used for all transmission. For the four node chain topology scenario, data channel 3 should never be used. Table 4.2 shows that 43,010 packets are transmitted on data channel 1, and 44,122 on data channel 2.

4.5 Chapter Summary

Two main simulator modules have been validated: the MAC protocols and the channel selection techniques. It was shown that the simulator achieves TCP throughput performance very close to a roughly calculated maximum. Furthermore, the results follow trends exhibited in the literature. The channel selection techniques were shown to use the proper channels given predefined parameters.

Chapter 5

Simulation Results

This chapter explores simulation results for the chain topology in detail. First, the performance of the three MAC protocols is presented. Second, channel selection techniques are evaluated. A thorough analysis of the implications of using a Random channel selection technique versus the Soft Channel Reservation technique is given. This analysis includes traffic visualization across the control and data channels, packet loss trends on the data channel, and the scenarios under which data losses occur. Third, the effect of the number of data channels is discussed. Finally, the last set of experiments explores the implications of different TCP configuration parameters.

5.1 MAC Protocol Evaluation

Figure 5.1 compares the TCP throughput results using the IEEE 802.11 MAC protocol, the Multi-Channel MAC protocol with three data channels (and one control channel), and the Bi-Directional Multi-Channel MAC protocol with three data channels (and one control channel). Both the Bi-MCMAC and the MCMAC utilize the Soft Channel Reservation channel selection technique. Chain length is varied from 2 to 16 nodes. The FTP protocol is used at the application layer providing a continuous supply of data to TCP for transmission. All simulations were run for 300 seconds.

There are two general trends that are evident in Figure 5.1 when comparing the three MAC protocols. First, throughput decreases as chain length increases, which

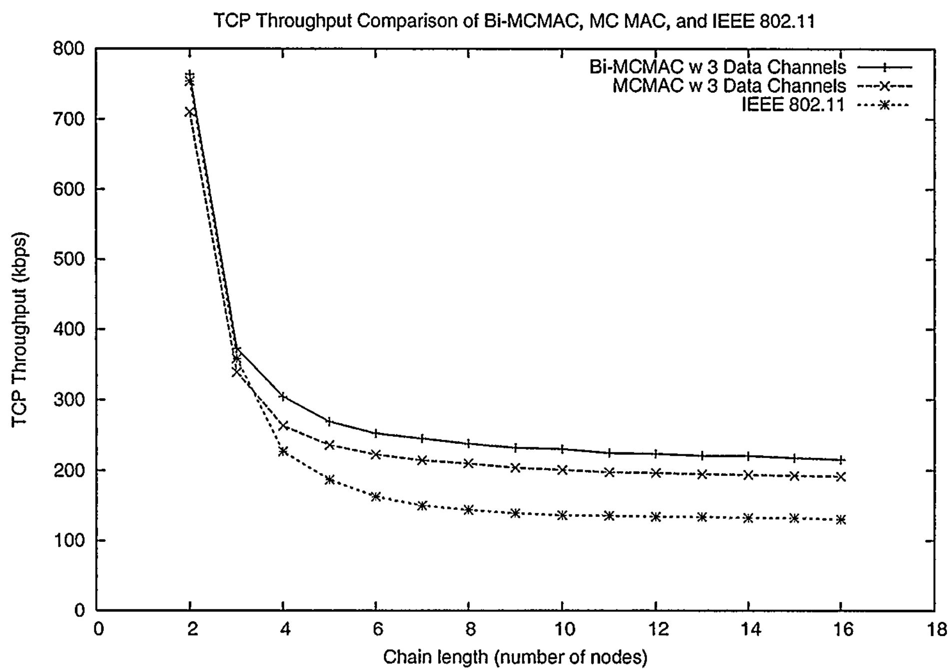


Figure 5.1: TCP throughput comparison of 3 different MAC protocols over varying chain lengths.

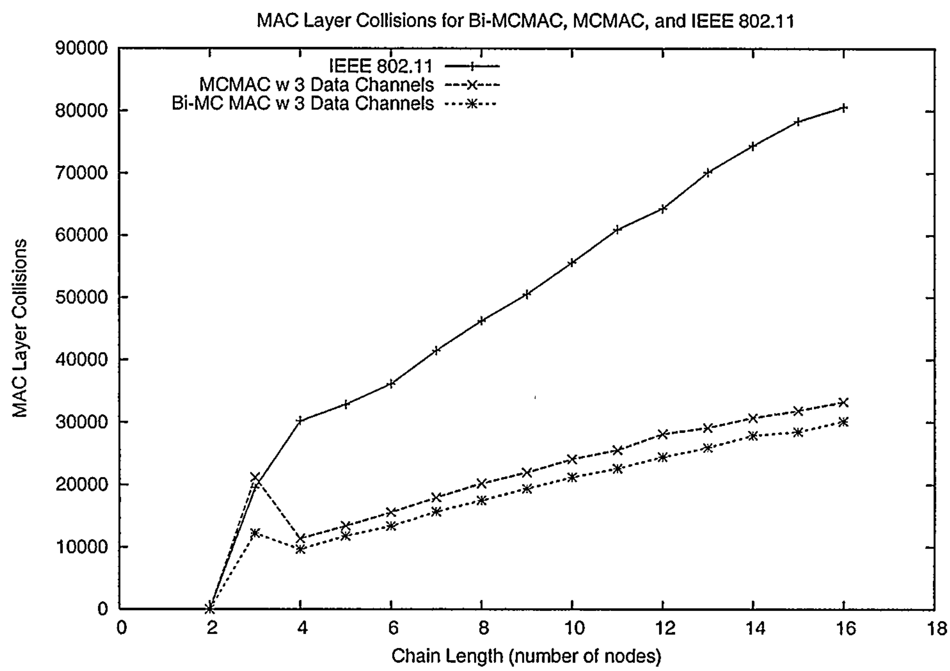


Figure 5.2: The number of MAC layer collisions using 3 different MAC protocols over varying chain lengths.

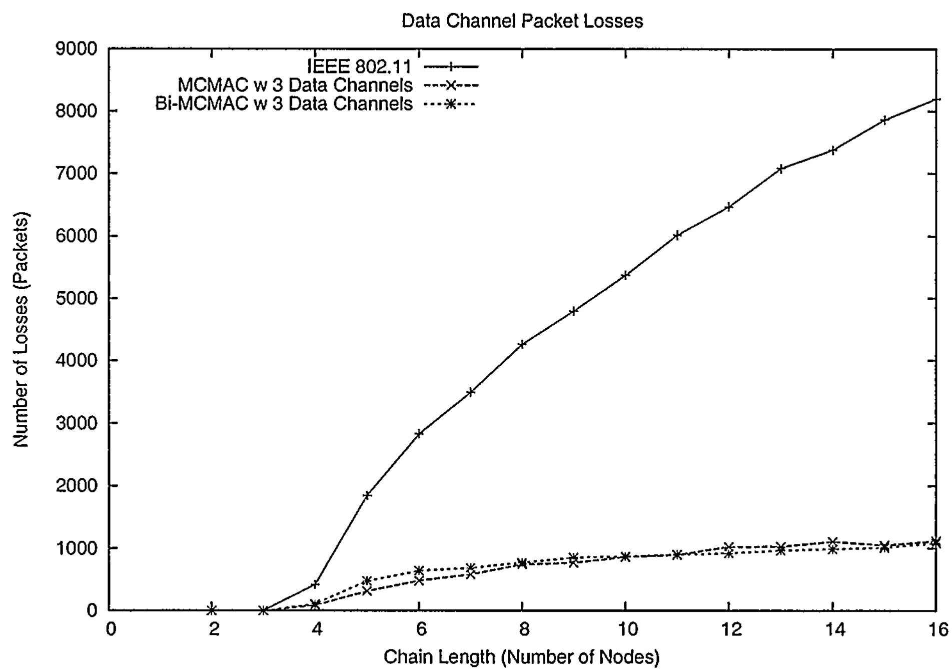


Figure 5.3: The number of data packet losses at the MAC layer using 3 different MAC protocols over varying chain lengths.

is consistent with other findings in the literature [4] [14] [17]. When chain length increases, data packets must traverse the chain a single hop at a time. This hop by hop transmission reduces throughput. Second, the Bi-MCMAC protocol provides better performance than both the MCMAC and IEEE 802.11 MAC protocols for all chain lengths. Again, this finding is consistent with those in the literature [14]. The bi-directional mechanism improves protocol efficiency by reducing the number of control frames required to transmit a data packet. Figures 5.2 to 5.4 provide additional results illustrating these performance differences.

In the simulation case with only 2 nodes, the Bi-MCMAC shows slight performance benefits over both IEEE 802.11 and MCMAC. Drastic improvements are not realized by the multi-channel protocols because there is only one concurrent transmission occurring given that there are only 2 nodes participating. This results in the multi-channel capability providing little benefit in a network of size 2 nodes.

The results for the 2 node case are dominated by the MAC layer overhead of the RTS, CTS, and ACK frames. However, even considering the MAC layer overhead, the TCP throughput between the three protocols still shows small differences. The MCMAC protocol performs the worst of the three protocols. This is due to the overhead of the CRN frame which, in the case of MCMAC, is only used for relaying channel selection to neighbours. Because the 802.11 protocol does not utilize the CRN frame, it is able to use this time transferring data, therefore resulting in a higher TCP throughput than the MCMAC. The Bi-MCMAC, however, makes use of the CRN frame to enable an extra data packet exchange per handshake. Even though the extra packet being exchanged is small in size (TCP ACK packets are 40 bytes vs TCP data packets which are 1024 bytes) the reduction in the number of handshakes results in higher TCP throughput. Figure 5.4 shows that the Bi-MCMAC protocol has a consistently higher data packet per handshake ratio than the MCMAC.

The results in Figure 5.1 for the simulation case with 3 nodes show a drastic decrease in TCP throughput for all three MAC protocols tested. The throughput reduction of almost 50% is the result of two factors. The first factor is that all packets require two hops from the sender to the receiver, with the first transmission from the sender to the intermediate node and the second from the intermediate node to the receiver. Second, contention between the sending node transmitting TCP data packets in the forward direction and the receiving node transmitting

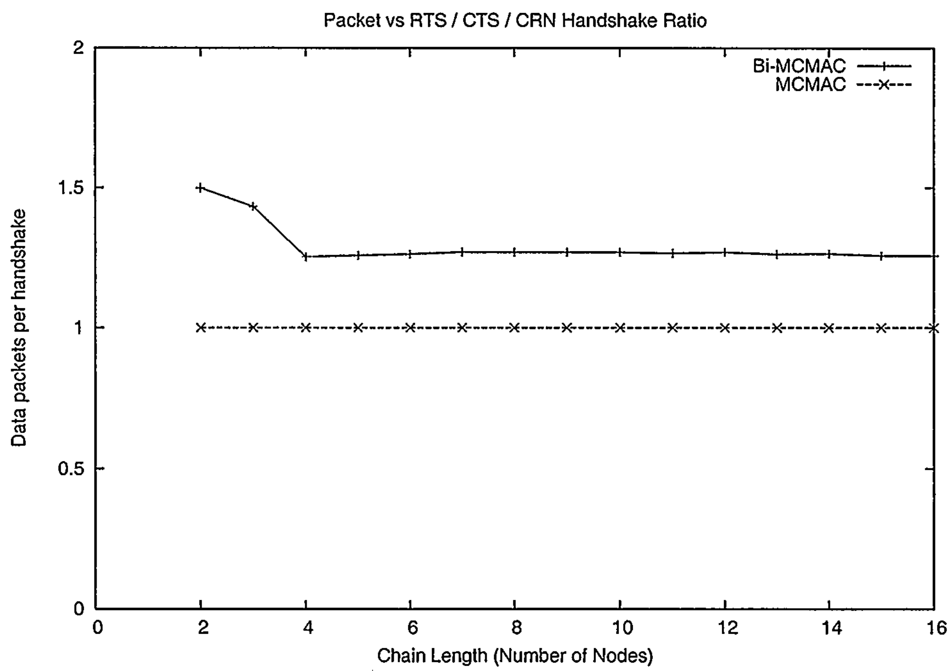


Figure 5.4: Ratio of data packets transmitted versus the number of RTS / CTS / CRN handshakes for varying chain lengths.

TCP ACK packets in the reverse direction results in collisions during the RTS/CTS handshake. The 2 node case has no MAC layer collisions, as shown in Figure 5.2, while the 3 node case has many collisions. These collisions result in a lower channel utilization for transmitting TCP data.

Although there are MAC layer collisions, Figure 5.3 shows that no data packets are lost for any of the MAC protocols in the 3 node case. When using 3 nodes, only one node can be transmitting at a time, as the sender and receiver are within the carrier sensing range of each other. This prevents concurrent transmissions for multi-channel protocols and, as was the case with only 2 nodes, results in minimal throughput advantage. However, as seen in Figure 5.4, the bi-directional mechanism of the Bi-MCMAC reduces the number of handshakes required to transmit data packets. Furthermore, this reduction in handshakes also reduces the MAC collisions by alleviating the Data / ACK contention, ultimately resulting in a slight performance improvement.

Throughput differences between the three MAC protocols in Figure 5.1 are more evident in the 4 node case. The two multi-channel MAC protocols have higher throughput since they allow two nodes to transmit on different data channels concurrently. Node 1 is able to communicate with node 2 while node 3 is communicating with node 4, when they have chosen different data channels. The IEEE 802.11 MAC protocol, however, does not have this spatial reuse; node 3 cannot transmit because of the exposed node problem [4]. Node 3 will ignore any RTS frames it receives from node 4 because it is in the carrier sensing range of node 1. Node 4 will continually attempt to transmit RTS frames, resulting in more MAC layer collisions at node 2 (hidden terminal problem). This is confirmed in Figure 5.2 where the IEEE 802.11 protocol experiences roughly twice as many MAC layer collisions as the multi-channel MAC protocols.

The Bi-MCMAC protocol outperforms the MCMAC in the 4 node scenario. Similar to the 2 and 3 node case, the Bi-MCMAC has lower overhead due to the bidirectional mechanism in the 4 node scenario. The data packet to handshake ratio is higher for the Bi-MCMAC. This reduces the opportunity for MAC layer collisions as fewer control channel packets are transmitted.

Although the IEEE 802.11 protocol is unable to have multiple concurrent transmissions occurring in a 4 node chain topology, it can in a 5 node chain. This results in a maximum channel utilization of 1/4 for IEEE 802.11 and 1/3 for both the Bi-

MCMAC and the MCMAC [14] [17].

For chain lengths beyond 4 nodes a slow reduction in throughput is seen for all three MAC protocols. Li et al.[17] show that as chain length increases, throughput can remain linear if an optimal send rate is maintained at the sender. Fu et al. [4] explain that TCP is unable to inject packets into the network at the optimal send rate. They explain that this is due to the inability of TCP to operate at the optimal window size required in order to exploit spatial channel reuse. Rather, TCP often uses a window size much larger. MAC layer frame losses are not significant enough to stabilize the TCP window at this optimal value, which they show to be $\frac{h}{4}$, where h is the number of hops in a chain [4]. The impact of TCP parameters are examined further in Section 5.4.

5.2 Effect of Channel Selection Technique

Next, the four channel selection techniques were evaluated using the Bi-MCMAC protocol. Three data channels, as well as the control channel, were used for all simulations. Again, the application layer traffic model was FTP providing TCP with a continuous stream of packets for transmission. Chain length was varied from 2 to 16 nodes. Simulations were run for 300 seconds.

Figure 5.5 shows the TCP throughput results for four channel selection techniques: Random channel selection, Lowest Channel Available, and the Soft Channel Reservation technique with and without randomization. Figure 5.6 shows the packet loss rates, while Figure 5.7 shows the data channel packet loss rates for the three techniques.

When the chain length is 2 or 3 nodes the throughput results are identical for all four channel selection techniques. The reason for this is that there are never concurrent data channels in use at a time. Because there is only one data channel in use at a time, the method for selecting this channel is not important; any of the channels selected will suffice. As was seen in Figure 5.2 in the previous section, the 3 node case shows a drastic reduction in throughput as compared to the 2 node case. The reasoning is unchanged: as chain length increases the number of hops to the destination increases, requiring more exchanges to take place.

Beginning with the 4 node scenario, the Soft Channel Reservation techniques

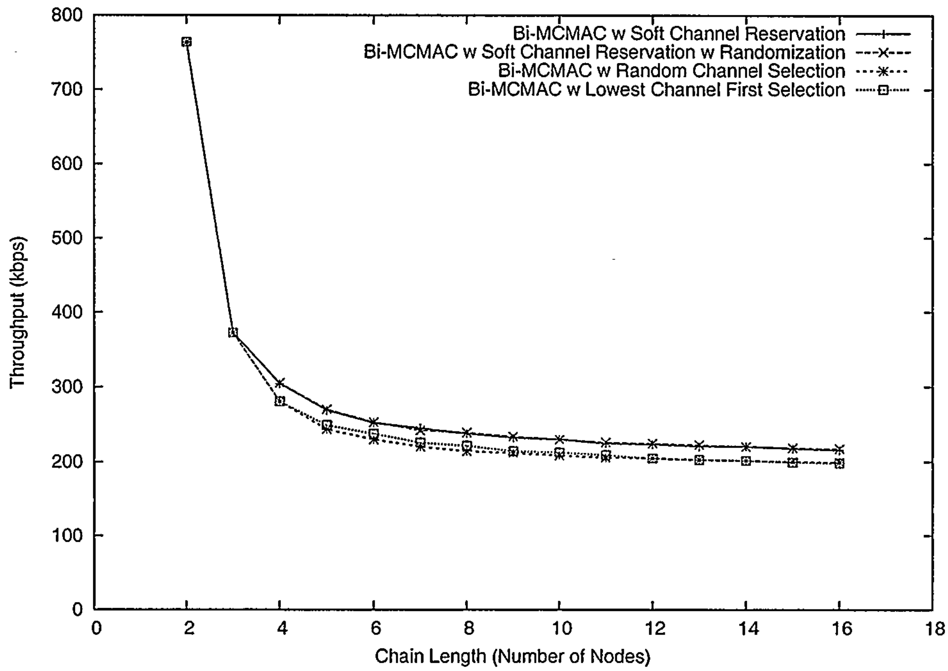


Figure 5.5: Effect of channel selection technique on TCP throughput for varying chain lengths.

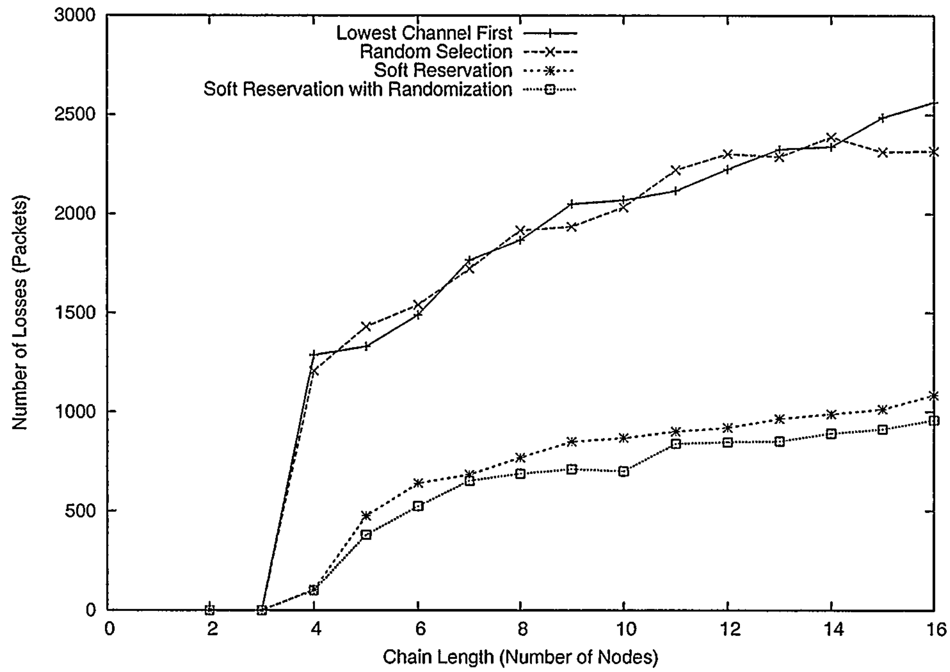


Figure 5.6: Effect of channel selection technique on MAC layer data packet loss counts for varying chain lengths.

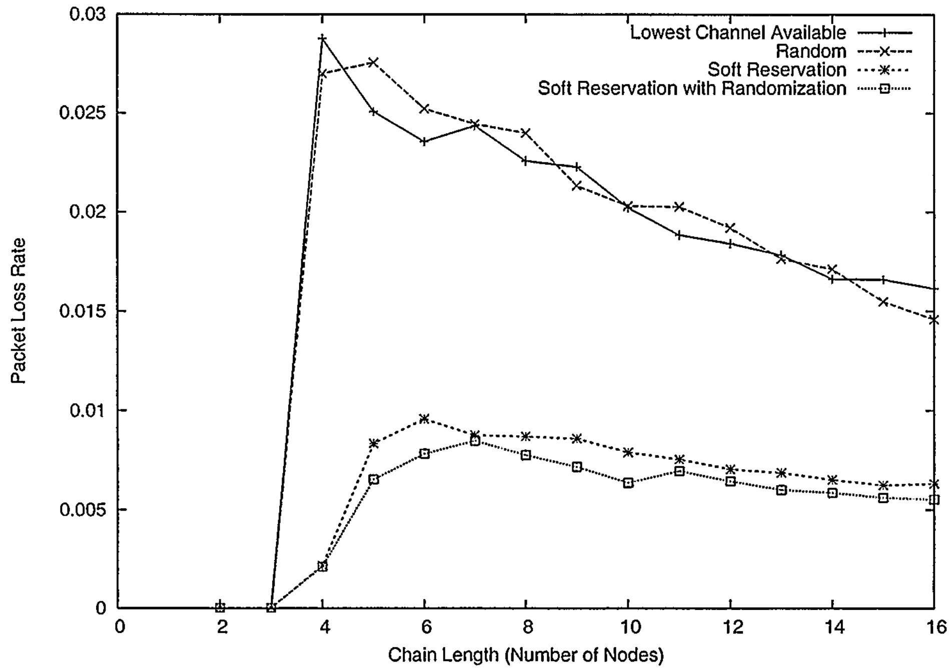


Figure 5.7: Effect of channel selection techniques on MAC layer data packet loss rates for varying chain lengths.

show a 10 % TCP throughput performance improvement versus the Lowest Channel First and Random channel selection techniques. This throughput benefit carries forward to the 16 node case. The reason for this improvement becomes apparent in Figure 5.6 and Figure 5.7. At the 4 node case, the Soft Channel Reservation techniques has significantly reduced data channel packet losses. The Soft Channel Reservation technique is able to maintain a lower data channel packet loss rate as the chain length increases. This reduction is due to a variety of factors induced by the channel selection technique and its prevalence in reducing the Multi-Channel Hidden Terminal Problem. The remainder of this section examines these factors in detail.

5.2.1 Data Channel Loss Scenarios

The multi-channel hidden terminal problem is the cause of data packet loss in the multi-channel MAC protocols. Upon close inspection of all data channel losses, it was found that there are four specific scenarios under which these losses occur. The channel selection technique has a large impact on the severity of this problem. The soft channel reservation techniques are able to alleviate one of the four scenarios. The remainder of this section explores these scenarios in detail.

Simulation experiments with 4 nodes are analyzed. Nodes are numbered 0 to 3, with Node 0 being the TCP source, Node 3 being the TCP destination, and nodes 1 and 2 intermediate nodes.

Scenario 1: Missed Reservation

The first data channel loss scenario is known in the literature as the missed reservation problem. In general, a node is busy transmitting or receiving on a data channel when a neighbouring node initiates a channel reservation handshake. Because a node is active on a data channel, it is unable to learn of the channel its neighbour selected and, in turn, may choose the same channel when it begins its next data exchange.

Figure 5.8 graphically displays data extracted from simulation trace files depicting an example of this loss scenario. Nodes 2 and 3 perform an RTS/CTS/CRN handshake exchange at time 259.705 and begin communicating on data channel 3. Node 1 is aware of their channel selection because it received the CRN from node

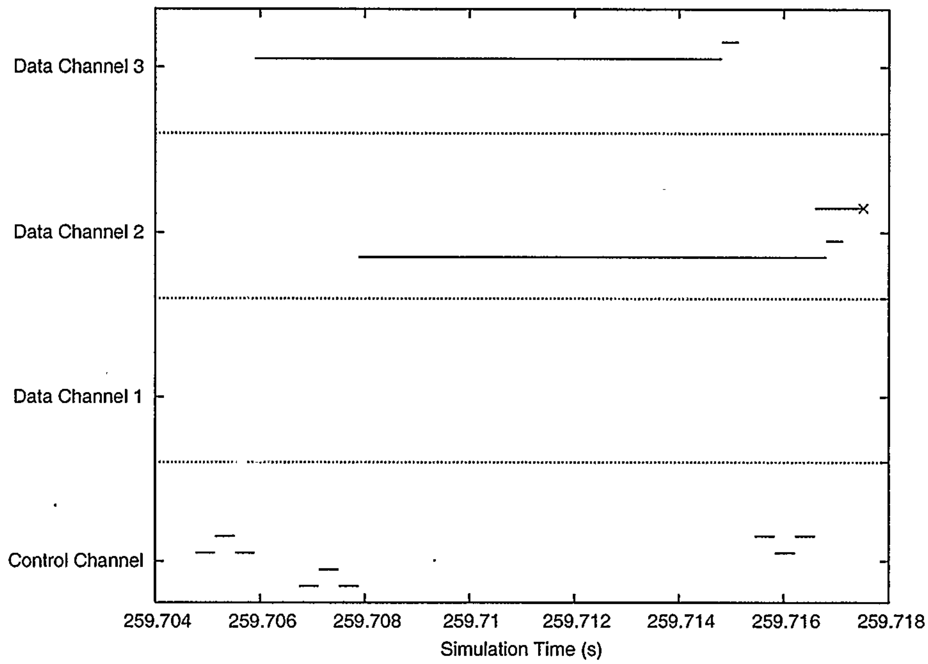


Figure 5.8: Data channel loss scenario 1: Missed reservation

2. Nodes 0 and 1 then perform an RTS/CTS/CRN handshake exchange at time 259.707, choosing data channel 2 for their exchange. However, when Node 1 sent the CRN, node 2 was unable to receive it because it was in a data exchange on channel 3 with node 3. Node 2 has missed the channel reservation. Once node 2 and 3 conclude their data exchange at time 259.715, they perform the RTS/CTS/CRN handshake exchange at time 259.716 in order to begin another data exchange. Unfortunately, because node 2 missed the channel reservation notification, it believes data channel 2 is available, and will include this channel in its *AvailableChannels* field in the RTS frame that it sends to node 3. Node 3 then decides to use data channel 2. When node 3 begins to listen on data channel 2, it receives two signals, one from node 2 as expected, and one from node 1 who is transmitting to node 0. In this case, the transmission from node 1 to 0 will not be corrupted as node 0 is outside the interference range of node 2. However, the data being transmitted from node 2 will be corrupted resulting in a data channel loss.

The Soft Channel Reservation technique alleviates the data channel losses associated with this scenario. Because the Soft Reservation technique prefers using the last channel successfully used (if it is available), neighbouring nodes tend to use separate data channels over a short period of time. The bi-directional mechanism enables a pair of nodes to communicate via the same channel for a longer duration before switching channels

Both the Lowest Channel Available and the Random technique are susceptible to losses due to this scenario. Because a channel selection is made with incomplete information, and this selection does not consider past channel usage, there is a probability that nodes may select conflicting channels. Although this probability varies depending on the number of data channels used, both nodes deciding on the data channel to be used have an intersection of the channels that they think are available. Given that this intersection occurs, there will always be a chance that both nodes select the same channel.

Scenario 2: Missed CTS

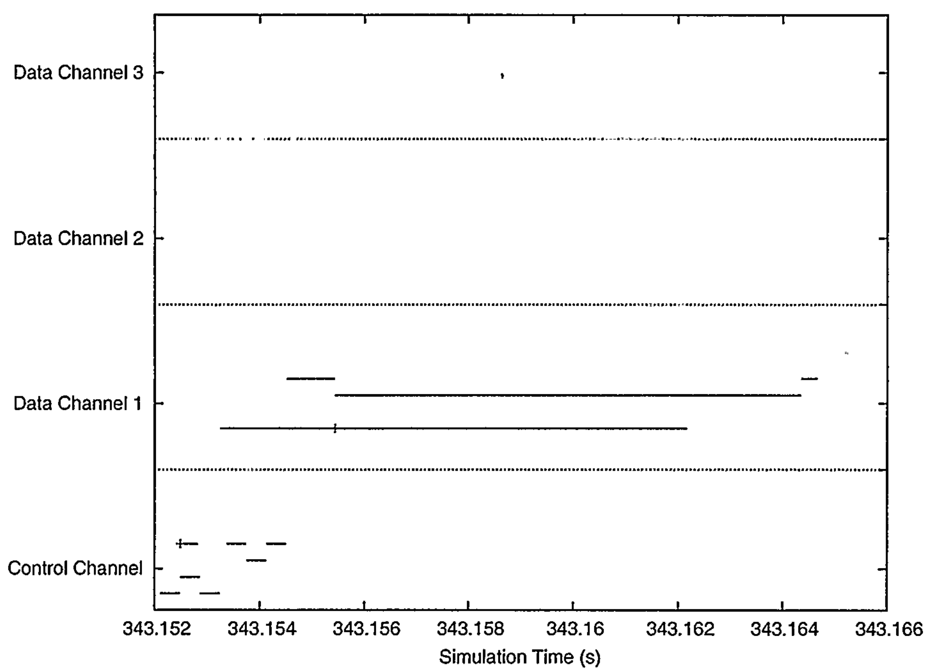


Figure 5.9: Data channel loss scenario 2: Missed CTS

The second data loss scenario is the result of a node missing a CTS frame that is transmitted between two other nodes. In this case, the node that would receive the CTS frame from its neighbour is busy attempting to receive an RTS frame from its other neighbour. Later, this node uses the same data channel as the node whose CTS it missed.

This data loss scenario is depicted in Figure 5.9 with data extracted from simulation traces. Node 0 begins an RTS / CTS / CRN handshake with node 1 at time 343.152 by transmitting an RTS frame to it. Node 3 then attempts to begin a separate RTS / CTS / CRN handshake with node 2 by sending an RTS frame. While node 3 is transmitting, node 1 performs physical carrier sensing and, due to the hidden terminal problem, begins transmitting the CTS to node 0 with the data channel it selected for use. In order for node 2 to learn of the channel selected, it must overhear this frame. However, node 2 is currently receiving the RTS frame from node 3. Node 2 therefore misses the CTS frame informing it of the channel that nodes 0 and 1 have chosen for their data transmission.

As well, the RTS frame that node 3 was transmitting at time 343.1525 is not properly received by node 2. Because it receives two different signals, a frame collision results. Node 3 does not receive a response from node 2 due to the frame collision and will therefore attempt the RTS / CTS / CRN / handshake after a backoff period. This handshake at time 343.154 will be successful but, because node 2 does not know which channel nodes 0 and 1 are using, the opportunity for selecting the same data channel exists. In this case, node 2 selects the same data channel that nodes 0 and 1 are using, corrupting the transmission from node 0 to node 1.

All of the channel selection techniques are susceptible to data channel losses under this scenario. However, the fact that a node misses a CTS frame does not guarantee that a data channel loss will occur. For example, using randomized channel selection and 3 data channels, node 2 has a 33% chance of selecting the same data channel that is being used. Likewise, if node 2 and node 3 had used different data channels during the previous transmission using the Soft Channel Reservation technique, they would choose their previous channels again, therefore avoiding a data channel loss. However, regardless of the channel selection technique used, it appears there is always a chance that a loss can occur in this scenario. Each node making the channel selection decision will be operating on a set of possible data channels. Because at least one node is unaware of a channel selection made, its list of possible channels for use will include the channel that was selected.

Scenario 3: Missed CRN

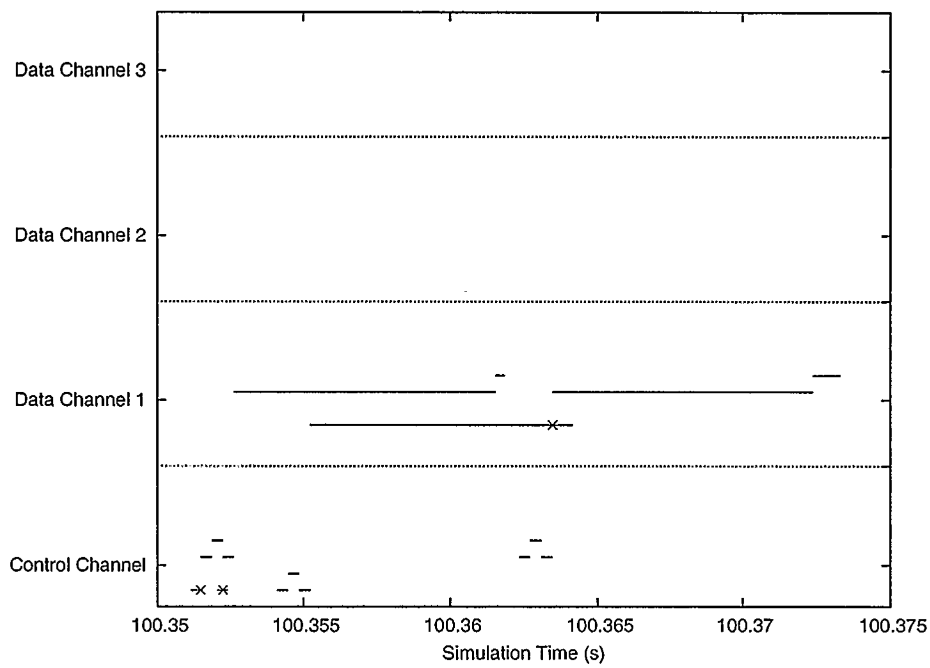


Figure 5.10: Data channel loss scenario 3: Missed CRN

The third data channel loss scenario is a complex scenario that initially starts with a node missing a CRN frame. A node needs to receive a CRN frame in order to know which data channel a pair of nodes will use. This CRN may collide with an RTS frame. The result is a node not knowing the data channel its neighbour has chosen.

Figure 5.10 shows this scenario from the simulation trace files. Node 0 and Node 2 both send RTS frames near time 100.351 to node 1 and node 4, respectively. The RTS destined for node 2 will be interfered with by the RTS being transmitted to node 4, resulting in a collision at node 1. Node 4, however, can receive the RTS destined for it, since node 4 is outside the interference range of node 0. Node 0, after sending the RTS, sets a timeout value to wait for a response from node 1. Due to the collision at node 1, it will not respond with a CTS.

During the timeout period of node 0, node 4 will transmit a CTS frame to node 3 with data channel 1 as the selected channel. Node 3 will then transmit the CRN frame to inform the neighbouring nodes about the channel they have selected.

While node 3 is in the process of sending the CRN frame the timer at node 0 expires, and it attempts to retransmit another RTS to node 1 at time 100.354.

A collision results at node 1 as it receives the CRN from node 3 and the RTS from node 0 concurrently. After the CRN has been transmitted, nodes 3 and 4 tune to data channel 1, and node 3 will commence transmission of a TCP data packet to node 4. Node 0 again enters a timeout stage waiting for a CTS response from node 1. Again, the timer will expire and node 0 will instigate the RTS / CTS / CRN handshake by sending an RTS frame to node 1.

This time, because node 3 and 4 have completed their channel selection and are now utilizing the data channel, node 1 can respond with a CTS. However, due to the RTS frame colliding with the CRN frame destined for node 1, node 1 is not aware that nodes 3 and 4 are currently using data channel 1. Node 1 makes the decision to use channel 1. Nodes 0 and 1 then tune to channel 1 and begin their data transmission. Once Node 3 and Node 4 finish their transmission, they will attempt to select another data channel for a subsequent transmission. Because they were transmitting on the data channel when Node 1 transmitted the CTS with the decision to use channel 1, they are unaware that it is in use. They then select data channel 1, and begin transmitting data. Unfortunately, Node 1 is receiving data from node 0 on that channel and when node 3 begins transmitting, node 2 receives corrupted data.

Although this scenario is very similar to scenario number 1, there is a subtle difference in the events leading to the data packet loss. Due to the missed CRN by node 1, it inadvertently selects the same channel that it would have known was unavailable had it received the CRN. Loss scenario 1 was the result of a node using the data channel at the time of channel selection, whereas this scenario is the result of a missed CRN.

Scenario 4: Simultaneous Handshake

The fourth data channel loss scenario is the result of a simultaneous handshake between two pairs of nodes. This simultaneous handshake results in neither pair of nodes being aware of which data channel the other pair has selected.

Figure 5.11 contains an example of this problem which was extracted from the simulation trace files. Nodes 1 and 3 both decide to transmit data at the same

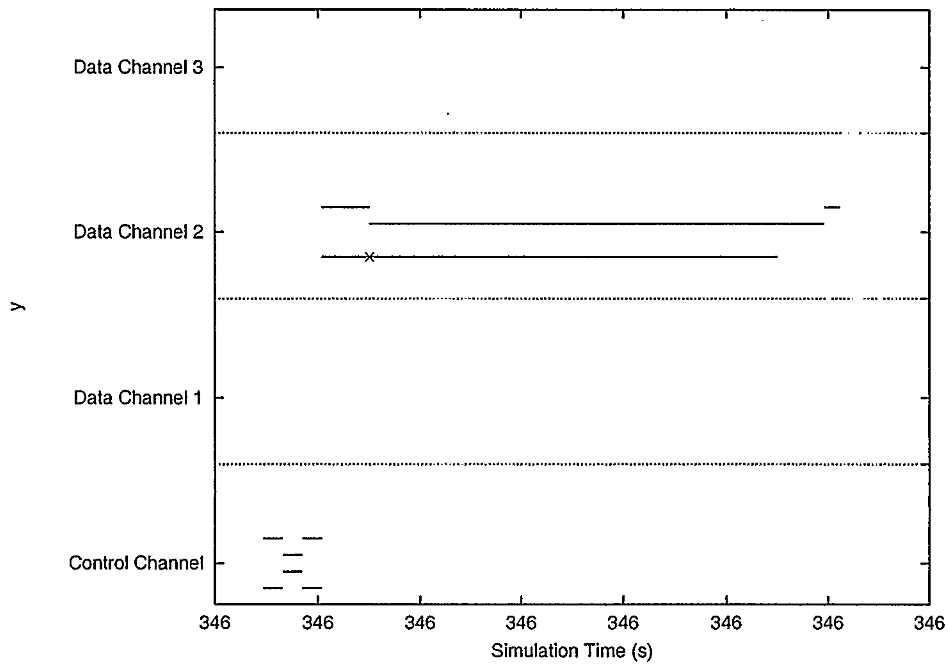


Figure 5.11: Data channel loss scenario 4: Simultaneous RTS / CTS / CRN handshake

time, and therefore begin the RTS / CTS / CRN handshaking mechanism. Given that the interference range of these nodes does not extend beyond 2 hops, nodes 1 and 2 are able to receive the RTS. Simultaneously, nodes 1 and 2 respond to the RTS with a CTS frame, and their channel selection. Again, due to the interference range not extending beyond 2 hops, the CTS frames are received correctly.

The channel selections made by nodes 1 and 2 were performed without knowledge of the channels that are being selected, therefore the opportunity for a data channel collision is possible. In this case, both pairs have selected the same data channel for their exchange, and a data channel loss will occur. Node 4 begins by transmitting a short TCP ACK packet to node 3, while node 0 simultaneously transmits a long TCP Data packet to node 1. The TCP ACK packet is received successfully by node 3. As well, at this point node 1 has successfully received a portion of the TCP Data packet being transmitted by node 0. However, when node 2 begins to transmit a large TCP Data packet back to node 3 using the bi-directional mechanism, node 1 is within node 2's interference range. This results in node 1 receiving two wireless signals, causing the TCP Data being transmitted from node 0 to node 1 to be lost. Furthermore, due to the data packet being lost, node 1 will not utilize the bi-directional mechanism to transmit data to node 0.

Similar to scenario 2, all of the channel selection techniques are susceptible to this loss scenario. However, the Lowest Channel Available technique will often result in a data loss. Because neither pair of nodes knows the data channel chosen by their neighbour, they will both choose the lowest data channel, resulting in a frame collision.

Random channel selection has a lower chance of collision than Lowest Channel Available. Both nodes making the channel selection decision do so independently, without regard for which data channel the other node is selecting. For this reason, random channel selection has a collision probability of $\frac{1}{3}$ when using three data channels, and $\frac{1}{N}$ in the general case of N data channels.

Soft Channel Selection's susceptibility to this scenario, however, is dependent on previous channel selection decisions. In order for a loss to occur, both nodes must have previously transmitted a data frame on the same channel. However, because a node tends to use the same channel for a short period of time, this is unlikely to occur.

Next, the prevalence of each data channel loss scenario was estimated. A

Table 5.1: Estimated prevalence of each data channel loss scenario

Loss scenario	Count
Missed Reservation Problem	21
Missed CTS	6
Missed CRN	2
Simultaneous Handshake	1

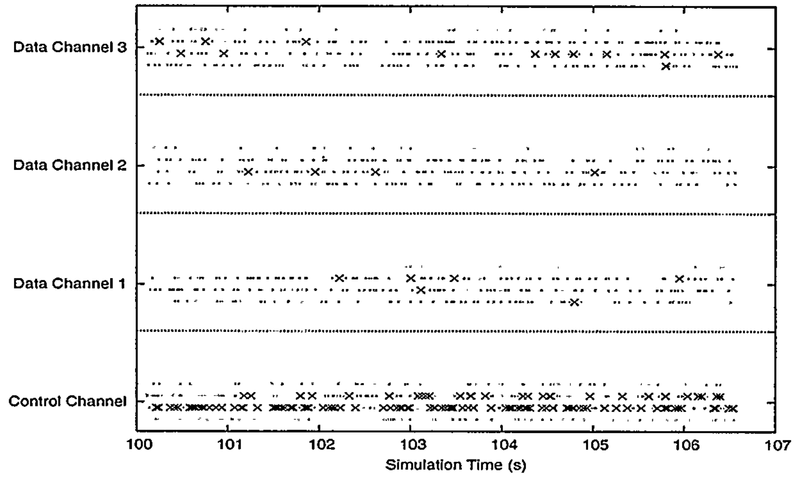
sequence of 30 data channel losses (out of 1206) were inspected, and the number of occurences of each scenario was noted. The results are displayed in Table 5.1. The missed reservation problem is the most common data channel loss scenario, while the simultaneous handshake scenario is least likely to occur. The drastic reduction in data channel losses when using Soft Channel Reservation is due to its ability to alleviate the Missed Reservation Problem.

5.2.2 Traffic Visualization

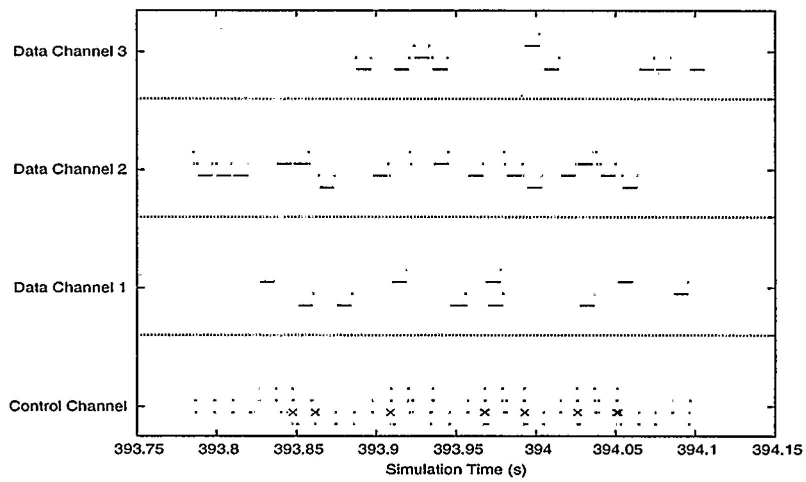
In order to further understand the interactions between the channel selection strategy and data packet loss, channel traffic graphs are constructed. These graphs display the traffic on each wireless channel between all nodes on the network. Viewing these graphs at different time scales provides insights into recurring traffic patterns and loss trends that appear. Simulation cases with four nodes and three data channels, along with the single control channel, were run and analyzed.

Figure 5.12 displays the network traffic at different time scales when using the Random channel selection technique. The random channel selection technique uses all three data channels. The majority of MAC layer losses occur on the control channel due to the hidden terminal problem. Furthermore, the medium time scale traffic graph shows that nodes often switch between data channels for transmission. A node is unlikely to use the same channel repeatedly over a prolonged period of time.

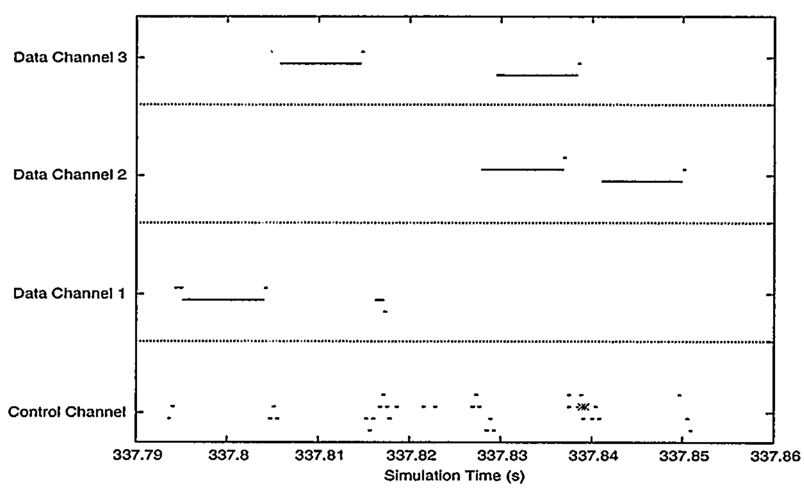
Soft Channel Reservation traffic visualization is displayed in Figure 5.13. Due to the use of only 4 nodes, no traffic is found on data channel 3. Like random channel selection, the long duration timescale shows that the majority of losses occur on the control channel due to the hidden terminal problem. However, the number of losses on the data channels is substantially lower. By analyzing the medium duration figure, it becomes apparent that soft channel reservation is likely



(a) Long duration (10,000 frames; 7 seconds)



(b) Medium duration (500 frames; 0.4 seconds)



(c) Short duration (100 frames; 0.07 seconds)

Figure 5.12: Traffic visualization of random channel selection

to use the same data channel that was last successfully used. Nodes rarely switch between data channels. For example, node 1 uses data channel 1 for the majority of its transmissions over the time period analyzed. Furthermore, consulting the short duration traffic visualization shows a similar trend.

5.2.3 Data Channel Loss Trends

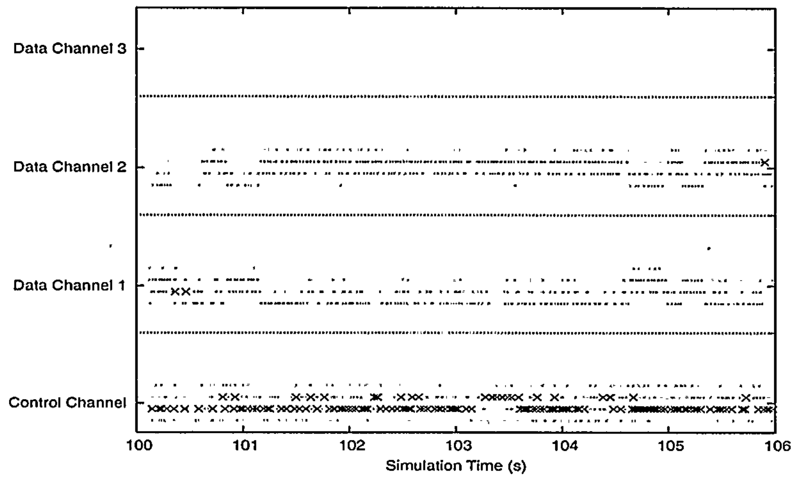
Data channel losses exhibit two interesting trends. The first trend is the relative location of MAC frame losses in the chain network. All simulation cases show that more losses occur near the TCP sender at the start of the chain, indicating that the TCP protocol is injecting packets into the network too quickly. Second, the Soft Channel Reservation technique was found to transmit more successful data frames between losses than the Random channel selection. These loss trends are analyzed further in this section.

Simulation experiments were analyzed with 4, 10, and 16 nodes using both Random channel selection and Soft Channel Reservation. Figure 5.14 displays the number of losses occurring at each node for the Random channel selection scenarios, while Figure 5.15 displays the Soft Channel Reservation scenario.

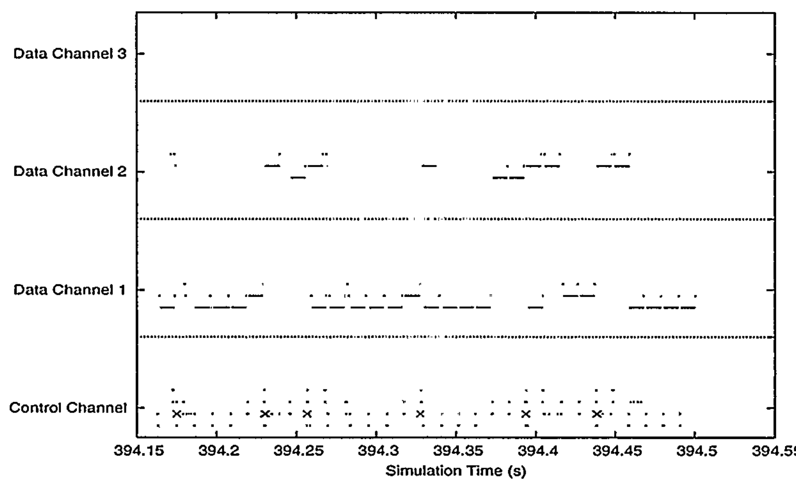
Regardless of channel selection technique or chain length, the node experiencing the most frame losses is the second node (node 1) in the chain. This is due to the TCP sender, node 0, attempting to inject data into the network at a faster rate than can be maintained. Node 0 will continually attempt to transmit RTS frames to node 1 while node 3 is attempting to transmit RTS frames to node 4. Node 4 will be able to receive the RTS frames, however RTS frames sent from node 0 to node 1 will collide with RTS frames being transmitted by node 3. Because the RTS frames are transmitted on the control channel, the channel selection technique does not affect this loss rate.

The 16 node scenario sees an increase in collisions at the second last node in the chain. This is due to contention with RTS frames being transmitted by node 15 to node 14 in an attempt to transmit TCP ACK packets to the TCP source. These RTS frames collide with RTS frames being transmitted from node 13 to node 14 in order to send TCP data to the destination.

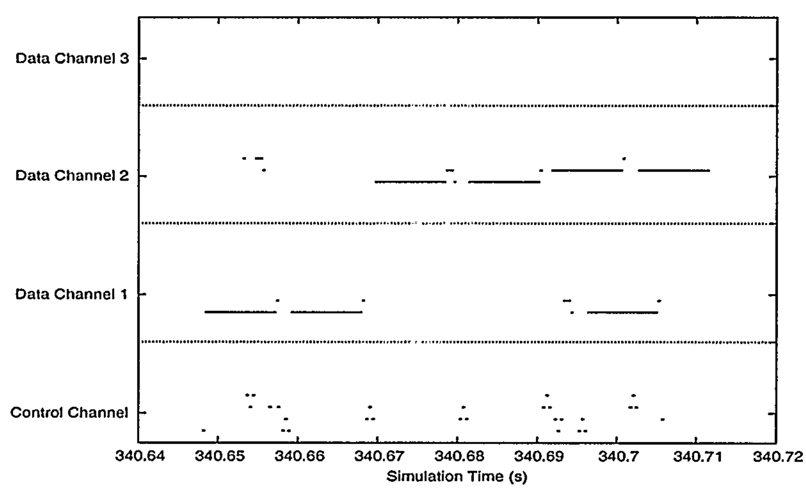
The second trend explored is the number of successfully transmitted data channel frames between data channel losses. A simulation experiment with 4 nodes was



(a) Long duration (10,000 frames; 6 seconds)

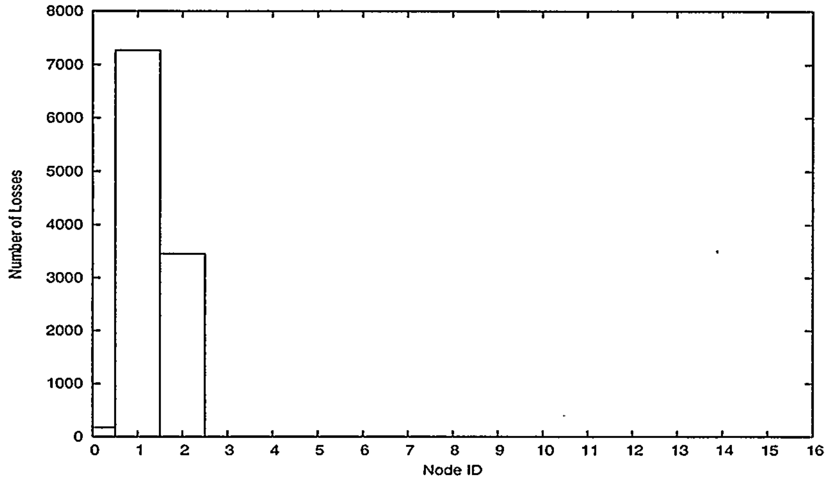


(b) Medium duration (500 frames; 0.4 seconds)

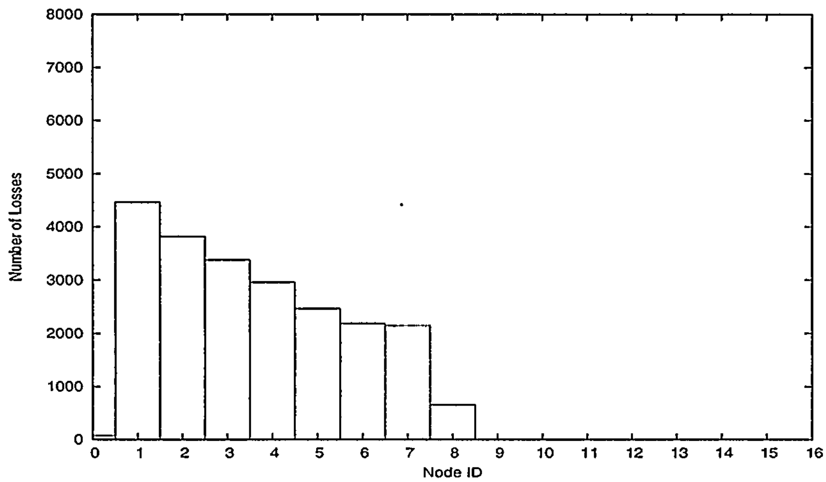


(c) Short duration (100 frames; 0.08 seconds)

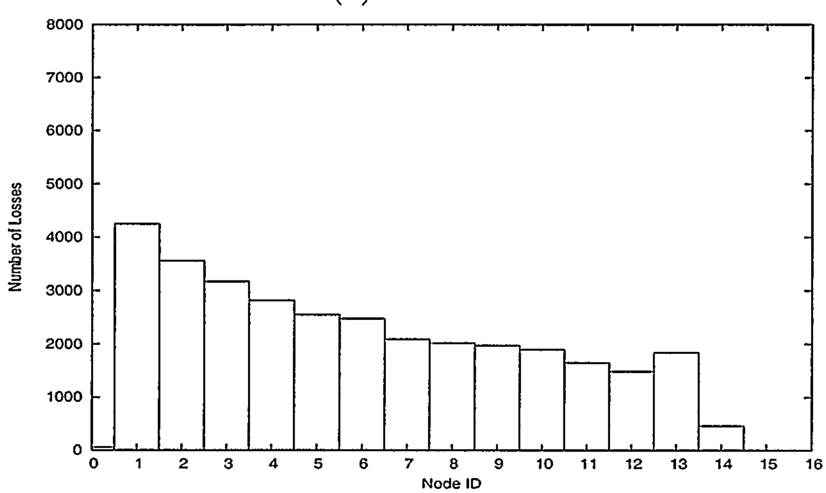
Figure 5.13: Traffic visualization of Soft Channel Reservation



(a) 4 Nodes

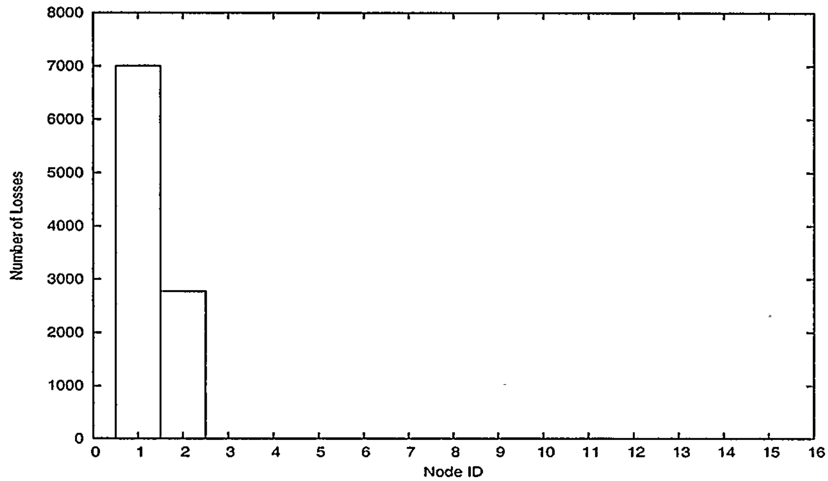


(b) 10 Nodes

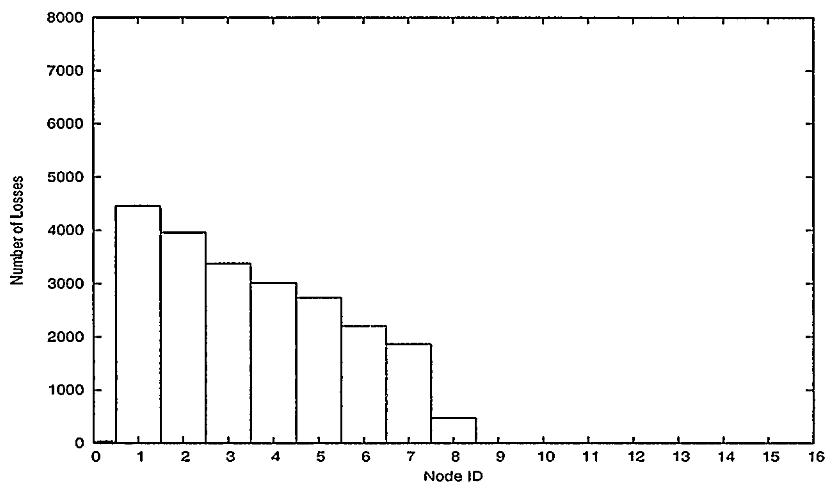


(c) 16 Nodes

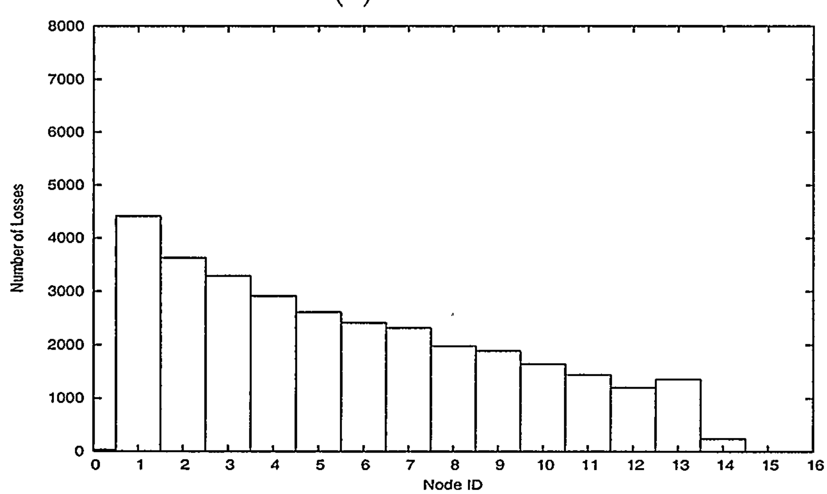
Figure 5.14: Loss Locations for Random channel selection



(a) 4 Nodes



(b) 10 Nodes



(c) 16 Nodes

Figure 5.15: Loss Locations for Soft Channel Reservation

run using both the Random channel selection technique, and Soft Channel Reservation. Two data channels and the single control channel were used. Data channel loss trends at the second node in the chain (node 1) were analyzed for both data channels. A successful transmission was noted when any frame destined to, or originating from, node 2 on a data channel occurred. A loss was noted when any frame destined to, or originating from, node 2 on a data channel occurred.

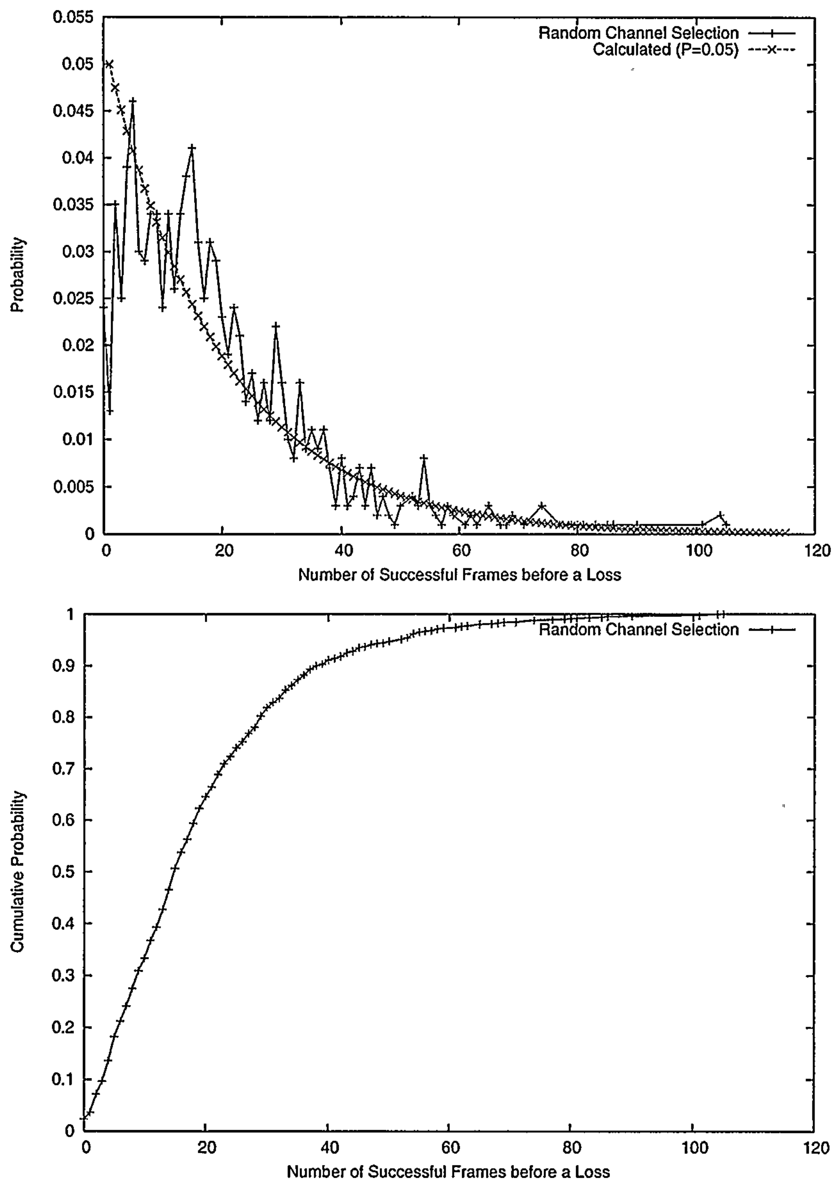


Figure 5.16: Probability of the number of successful frames prior to a packet loss.

Figure 5.16 displays the number of successful data frames prior to a collision for random channel selection. The probability that the Random channel selection transmits many successful data frames between losses is low. The number of successful data transmissions prior to a loss appears to follow a geometric distribution with P=0.05. Furthermore, the Random technique has a 50% chance of transmitting no more than 16 successful data frames prior to a loss. Soft Channel Reservation, on the other hand, does not follow this geometric trend. This is the result of Soft Channel reservations short term use of 'good' data channels for transmission.

5.3 Effect of Number of Data Channels

This section discusses the impact that the number of data channels has on the TCP throughput of the Bi-MCMAC protocol on the chain topology.

Wu et al. [31] indicate that a single control channel can, given an ideal MAC protocol, support up to at most N data channels, where $N \leq \frac{L_d}{L_c}$. In this formula, L_d is the length (either bytes or time) of transmission on a data channel for a given data frame transfer. This includes data as well as a MAC layer ACK. L_c is the length of the control channel handshake required in order to transmit a data packet. For the Bi-MCMAC, L_d is the length of an RTS (40 bytes), CTS (39 bytes), and CRN (40 bytes), for a total of 119 bytes. Assuming the bi-directional mechanism is used, a TCP Data packet (1060 bytes), a TCP ACK packet (40 bytes), and a MAC ACK (40 bytes) are transmitted using a 1 Mbps data channel, then L_d is 1140 bytes. The resulting number of channels that the control channel can support is 9. However, as will be seen, none of the channel selection techniques greatly benefit from more than 5 channels.

Figure 5.17 displays the results of the Bi-MCMAC with Soft Channel Reservation when using multiple data channels. The performance when only one channel is used is slightly higher than the IEEE 802.11 protocol performance. The slight increase in performance versus IEEE 802.11 is due to the bi-directional mechanism as explained in Section 5.1.

The Soft Channel Reservation techniques perform better using 2 data channels than a single data channel. The reason for the lower performance for the single

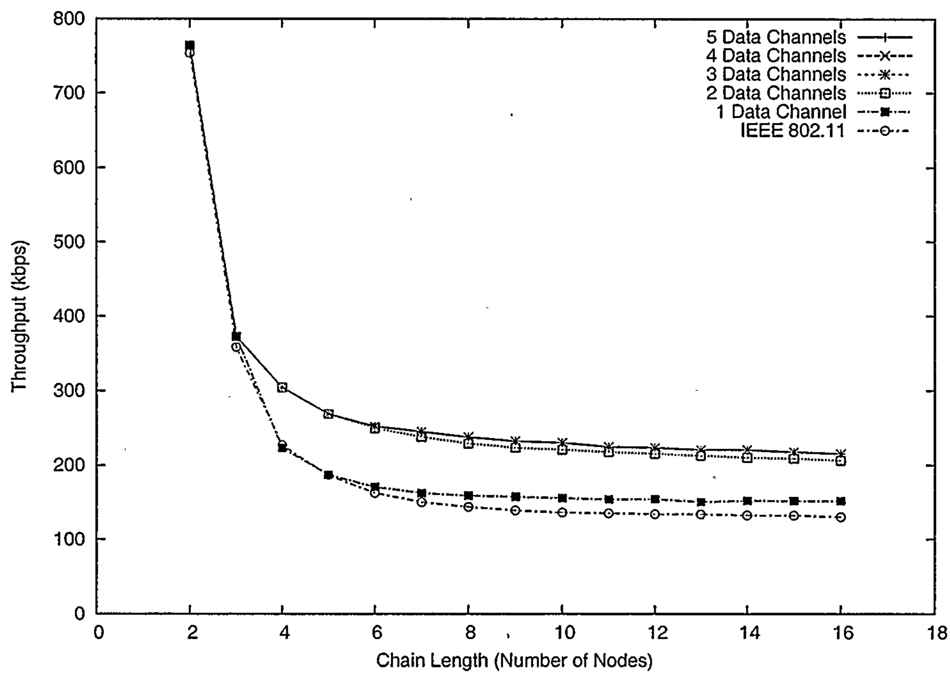


Figure 5.17: TCP throughput using Bi-MCMAC with Soft Channel Reservation with varying number of data channels.

channel scenarios is the inability for neighbouring nodes to transmit concurrently. This is due to data channel starvation [31]. Data channel starvation occurs when a node wishes to transmit data but is unable to do so due to a lack of available data channels.

When data channel starvation occurs, the throughput potential of all multi-channel protocols, regardless of the number of data channels in use, is reduced. In order to maximize throughput, data channel starvation must be eliminated. The simple solution to the problem is to increase the number of data channels. However, given current standards, a limited number of data channels are available. The IEEE 802.11 standard provides 4 non-overlapping channels for use, therefore allowing for one control channel and three data channels [18].

Using 3 data channels for Soft Channel Reservation provides minimal performance improvement compared to two data channels. Due to spatial channel reuse and two hop interference, only two data channels are required in order for all nodes on the chain to be participating in a data exchange. However, adding a third channel reduces the probability of data channel collisions due to the multi-channel hidden terminal problem.

The Soft Channel Reservation technique does not benefit by using 4 or more data channels. In fact, this technique will never transmit any data on a channel higher than number 3 in a chain topology. If the Soft Channel Reservation technique cannot use the channel it last used successfully, it will select the lowest possible channel. Due to two hop interference, there will always be such a channel available when 3 data channels are in use.

Soft Channel Reservation with randomization may transmit frames on a high numbered data channel. However, this does not translate into increased TCP throughput. Due to two hop interference, there always exists a data channel numbered less than 3 that could have been used.

Randomized channel selection provides for higher TCP throughput than IEEE 802.11 when only a single data channel is used. Figure 5.18 displays these results for multiple data channels. As was the case with Soft Channel Reservation, the bi-directional mechanism contributes to increased throughput for the single channel scenario.

Data channel starvation is reduced when two data channels are used with Random channel selection. The reduction in starvation allows for concurrent transmis-

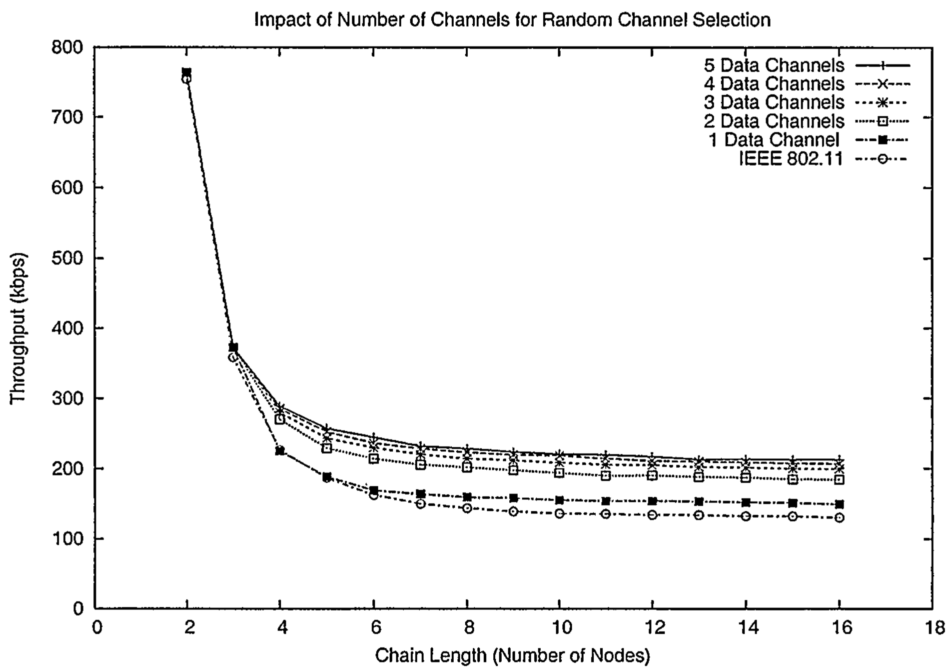


Figure 5.18: TCP throughput using Bi-MCMAC with Random channel selection with varying number of data channels.

sions to occur in closer proximity on the chain, therefore increasing throughput. However, due to the inability of random channel selection to avoid the missed reservation problem, its throughput with two data channels is lower than the throughput attained by Soft Channel Reservation with two data channels.

As more data channels are added when using the Random channel selection technique, the probability of data packet losses due to the missed reservation problem decreases. The greater the number of channels, the lower the chance of selecting a channel on which a pair of nodes are currently transmitting. However, even when using 5 data channels, the Random channel selection technique is unable to perform as well as the Soft Channel Reservation technique using only 3 data channels. Furthermore, the performance benefits of using more than 5 data channels are extremely small. Using 5 data channels only provides a 0.5% improvement versus 4 data channels. The benefit of adding a data channel diminishes as more are added when using Random channel selection.

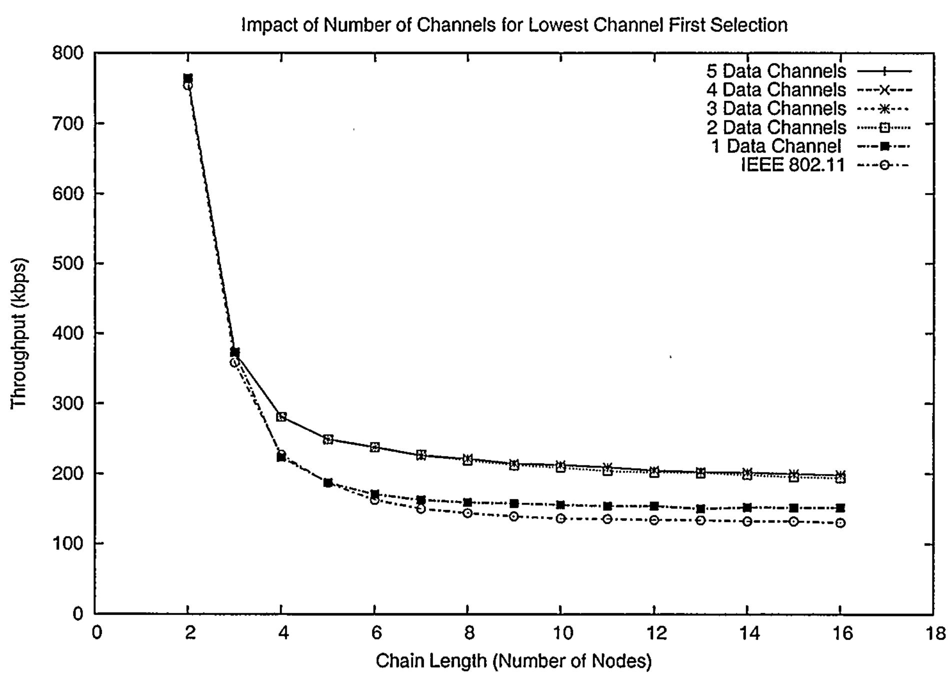


Figure 5.19: TCP throughput using Bi-MCMAC with Lowest Channel Available channel selection with varying number of data channels.

Lowest Channel Available channel selection does not benefit from adding more than 3 data channels. The argument is the same as Soft Channel Reservation. The nature of the chain topology and two-hop interference means that there will always be a data channel numbered 1 to 3 that is available for transmission. Furthermore, the benefit from using a third data channel is extremely small.

5.4 Effect of TCP Parameters

This section explores the implications of TCP parameters on the TCP throughput. The performance effect of enabling and disabling TCP delayed ACKs are tested. Different TCP packet sizes and their effect on TCP throughput are discussed

Simulation experiments were run with three MAC protocols with TCP delayed ACK enabled and disabled. Three data channels and one control channel were used for the multi-channel protocols. The TCP throughput results are displayed in Figure 5.20.

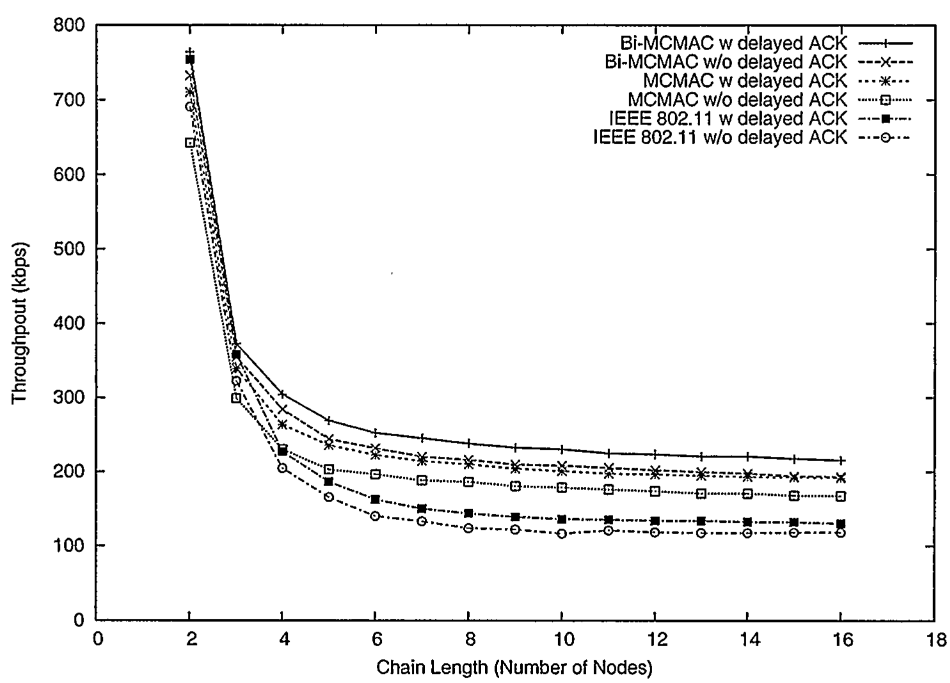


Figure 5.20: TCP throughput for three MAC protocols with and without TCP delayed ACK.

TCP throughput performance improves when delayed ACKs are enabled for all three MAC protocols tested. Delayed ACKs transmit a single ACK for every two data packets received. Therefore, traffic in the backward direction is halved. This reduces Data-ACK contention, and lessens the number of MAC layer frame collisions. TCP throughput performance improves as a result of fewer frame collisions.

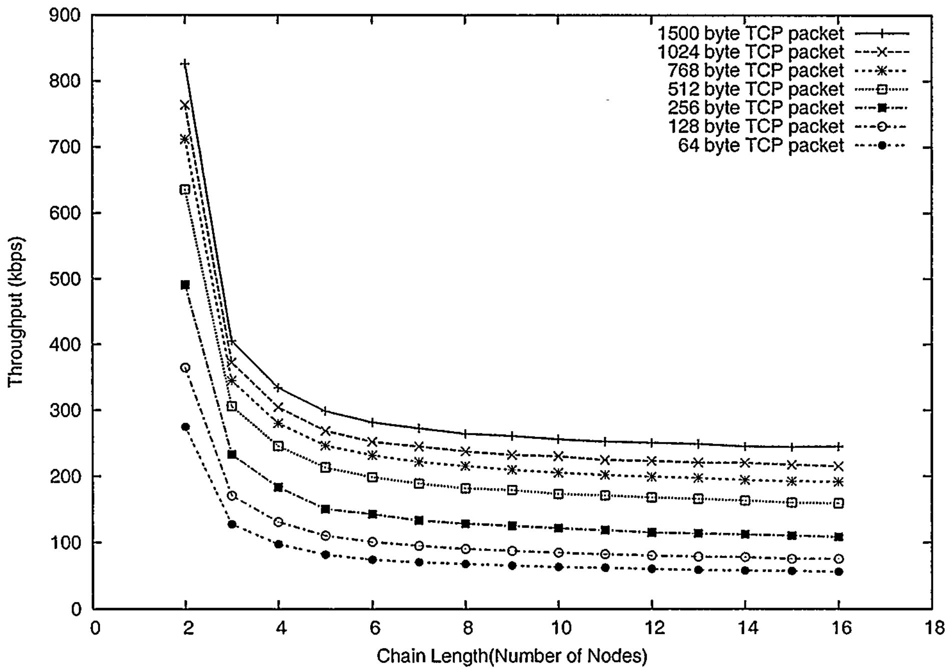


Figure 5.21: TCP throughput using Bi-MCMAC with Soft Channel Selection for different TCP packet sizes.

TCP packet size has a large impact on the performance of a wireless ad-hoc multi-hop network. A small TCP packet size results in a low throughput, while a large packet size improves performance. Figure 5.21 displays the results from simulation experiments using the Bi-MCMAC with three data channels and one control channel for various TCP packet sizes.

TCP performance suffers when the packet size is small due to the control channel becoming saturated. When TCP data packets are 64 bytes, only a single data channel is used. The control channel overhead required to send a single TCP packet, as calculated in Section 5.3, is 119 bytes. This is much higher than the amount of data to be transmitted, therefore saturating the control channel. Table 5.2 displays the ratio of control channel bytes versus data bytes transmitted. For

Table 5.2: Control channel overhead per data byte using different sized TCP packets

TCP Packet Size	Control versus data ratio
64	1.85
128	0.92
256	0.46
512	0.23
768	0.15
1024	0.11
1500	0.08

64 byte data packets, the control channel transmissions dominate.

Large packet sizes provide higher TCP throughput than smaller packets. Using larger data packets amortizes the control channel overhead associated with transmitting each packet. In the case of the 1500 byte data frame, a small amount of time is spent in the RTS / CTS / CRN sequence compared to actual data transmission. The control channel overhead per byte of data for 1500 byte data frames is 0.08, as seen in Table 5.2.

5.5 Chapter Summary

This chapter explored simulation results for the chain topology in detail. It was found that the Bi-MCMAC protocol outperforms the MCMAC protocol due to its reduction of Data-ACK contention. Both the Bi-MCMAC and MCMAC protocols TCP achieve higher throughput performance than IEEE 802.11 due to reduced Data-Data contention.

The Soft Channel Reservation channel selection technique reduces data channel frame losses. A thorough analysis of the data channel loss scenarios showed that the Soft Channel Reservation technique was able to alleviate the effect of the missed reservation problem. Furthermore, Soft Channel Reservation is able to successfully transmit more successful data frames between data channel losses than random channel selection.

The effects of basic TCP parameters were explored. Delayed ACKs reduce the Data-ACK contention at the MAC layer, resulting in higher throughput. Larger data packet sizes reduce the control channel contention.

Chapter 6

Conclusions

This chapter summarizes the main conclusions from this thesis, discusses the contributions, and suggests areas for future work.

6.1 Thesis Summary

This thesis analyzes channel selection techniques and their ability to alleviate the multi-channel hidden terminal problem. First, three MAC protocols are evaluated on a chain topology in an effort to improve TCP throughput. Next, channel selection techniques are explored for the purposes of reducing the multi-channel hidden terminal problem in the Bi-Directional Multi-Channel MAC Protocol. The impacts of the number of data channels are studied. Finally, the impacts of specific TCP parameters are evaluated.

The main contribution of this thesis is the detailed analysis of the circumstances surrounding data channel losses for multi-channel wireless networks. The channel selection technique plays a large role in reducing such losses, and in turn increasing TCP throughput. The Soft Channel Reservation technique attempts to use the data channel on which a successful data transmission has recently occurred if that channel is available. It was found through simulation that this short term channel reservation helps to reduce the number of data channel losses, and therefore increase TCP throughput. Furthermore, this mechanism is very successful when only three data channels are provided in the system.

6.2 Conclusions

There are four main conclusions from the simulation experiments:

- A multi-channel MAC protocol can alleviate, but not solve, the hidden terminal problem. The number of data packets lost is reduced because RTS frames no longer collide with data packets when multiple data channels are used. However, the multi-channel hidden terminal still has data channel losses if two transmitting nodes select the same data channel for transmission.

- The exposed node problem can also be mitigated by using a multi-channel MAC protocol. The degree of reduction, however, is dependent on the extent of data channel starvation. If a node wishes to transmit but there are no free data channels available for use, then the exposed node problem increases. By using at least two data channels, TCP performance improves due to the reduction of data channel starvation

- The multi-channel hidden terminal problem has a substantial impact on TCP performance. The channel selection technique used, however, can alleviate data packet losses associated with it. Using Soft Channel Reservation results in 10% TCP performance improvement versus randomized channel selection for long chain topologies. This improvement is the result of reduced data channel losses for the Soft Channel Reservation technique. It is able to avoid the most prevalent of the data channel loss scenarios, therefore reducing the impact of the multi-channel hidden terminal problem.

- The number of data channels used has a significant impact on TCP performance. Simulations show that on a chain topology, the benefit of using two data channels versus one data channel is 18% higher for randomized channel selection and 30% higher for the soft channel reservation technique. Furthermore, soft channel reservation achieves its peak TCP throughput using only three data channels. Randomized channel selection, however, is unable to match the performance of the soft channel reservation even when it uses as many as six data channels. The performance improvement from adding a data channel diminishes as more channels are added when using the randomized channel selection technique.

Multi-channel MAC protocols can improve TCP performance on multihop ad-hoc wireless networks. Careful channel selection can maximize the benefit of a multi-channel wireless network. Furthermore, only a modest number of data channels are required if they are managed carefully. The careful selection of TCP parameters can further improve network performance.

6.3 Future Work

The simulation results shown in this thesis show that the multi-channel hidden terminal problem can be alleviated using soft channel reservations. However, only a chain topology was studied. More complex network topologies, such as grid or randomized networks, need to be studied further. These topologies introduce added channel contention as the number of neighbouring nodes increases.

Mobility may introduce new data channel loss scenarios as transmitting nodes move into range of other receiving nodes. Physical layer characteristics of the wireless channel are affected when nodes are moving at different speeds.

The FTP traffic model provides infinite data to the TCP layer for transmission. This traffic model is simplistic. The HTTP protocol, as well as peer-to-peer protocols are prevalent on the Internet. These protocols may alter the viability of multi-channel MAC protocols, and therefore require further study.

The detailed data channel collision analysis in this thesis provides insights into reducing the multi-channel hidden terminal problem. It was shown that one of the loss scenarios is alleviated with soft channel reservation. However, the three remaining scenarios do not appear to be solvable by channel selection techniques. One potential area of exploration is reducing the likelihood of RTS frames colliding with CTS and CRN frames containing channel reservation information. In order to increase data channel utilization, concurrent transmissions on the same data channel may take place if the receiving nodes are distant from the competing transmitter. Integrating the channel selection technique with routing protocols may provide the information required to facilitate these decisions.

References

[1] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997.

[2] S. ElRakabawy, A. Klemm, and C. Lindemann. TCP with adaptive pacing for multihop wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 288–299, New York, NY, USA, 2005. ACM Press.

[3] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782 (Proposed Standard), April 2004.

[4] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP performance. *IEEE Transactions on Mobile Computing*, 4(2):209–221, March/April 2005.

[5] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang. TCP over wireless multihop protocols: Simulation and experiments. In *Proceedings of IEEE International Conference on Communications*, pages 1089–1094, June 1999.

[6] A. Gupta, I. Wormsbecker, and C. Williamson. Experimental evaluation of TCP performance in multi-hop wireless ad hoc networks. In *MASCOTS*, pages 3–11, 2004.

[7] G. Holland and N. Vaidya. Analysis of TCP performance over mobile ad hoc networks. *Wireless Networks*, 8(2-3):275–288, 2002.

[8] http://www.isi.edu/nsnam/ns/. The network simulator ns2.

[9] IEEE. IEEE std. 802.11, wireless LAN media access control (MAC) and physical layer (phy) specification, 1999.

[10] N. Jain, S. Das, and A. Nasipuri. A multichannel MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN2001)*, October 2001.

[11] R. Jiang, V. Gupta, and C. Ravishankar. Interactions between TCP and the IEEE 802.11 MAC protocol. In *DARPA Information Survivability Conference and Exposition*, page 273. DARPA, 2003.

[12] D. Johnson, D. Maltz, and Y. Hu. The dynamic source routing protocol for mobile ad hoc networks (DSR). IETF MANET Working Group., July 2004.

[13] E. Jung and N. Vaidya. A power control MAC protocol for ad hoc networks. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 36–47, New York, NY, USA, 2002. ACM Press.

[14] T. Kuang and C. Williamson. A bidirectional multi-channel MAC protocol for improving TCP performance on multihop wireless ad hoc networks. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 301–310, New York, NY, USA, 2004. ACM Press.

[15] J. Kurose and K. Rose. *Computer Networking: A top down approach featuring the Internet.* Addison-Wesley, 2003.

[16] T. Lang and D. Floreani. The impact of delayed acknowledgments on TCP performance over satellite links. In *WMI '01: Proceedings of the first workshop on Wireless mobile internet*, pages 56–61, New York, NY, USA, 2001. ACM Press.

[17] J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69, New York, NY, USA, 2001. ACM Press.

[18] J. Louderback. 4 simultaneous channels okay for 802.11b at http://www.extremetech.com/article2/0,3973,709067,00.asp.

[19] A. Medina, M. Allman, and S. Floyd. Measuring the evolution of transport protocols in the internet. *Computer Communication Review*, 35(2):37–52, 2005.

[20] A. Mishra, E. Rozner, and S. Banerjee. Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage. In *Internet Measurement Conference (IMC 2005)*, pages 311–316, 2005.

[21] A. Nasipuri and S. Das. Multichannel CSMA with signal power-based channel selection for multihop wireless networks. In *Proceedings of the IEEE Fall Vehicular Technology Conference (VTC 2000)*, September 2000.

[22] A. Nasipuri and J. Mondhe. Multi-channel MAC with dynamic channel selection for ad hoc networks, January 2004. Available at http://www.ece.uncc.edu/ anasipur/pubs/MC-04.pdf.

[23] A. Nasipuri, J. Zhuang, and S. Das. A multichannel CSMA MAC protocol for multihop wireless networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'99)*, September 1999.

[24] J. Padhye and S. Floyd. On inferring TCP behavior. In *SIGCOMM*, pages 287–298, 2001.

[25] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[26] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.

[27] T. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[28] J. So and N. Vaidya. Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 222–233, New York, NY, USA, 2004. ACM Press.

[29] K. Sundaresan, V. Anantharaman, H. Hsieh, and R. Sivakumar. ATP: A reliable transport protocol for ad-hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2003.

[30] Y. Tseng, S. Ni, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, 2002.

[31] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *ISPAN '00: Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00)*, page 232, Washington, DC, USA, 2000. IEEE Computer Society.

[32] XPlot. http://www.xplot.org.

[33] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saaranen. TCP performance issues over wireless links. *IEEE Communications Magazine*, 39(4):52–58, 2001.