A Dynamic MAC-Layer Mechanism Using Individual Mobility State Recognition to Increase Mobile Ad-Hoc Network Throughput Performance

by

Aaron L. Phillips, BSIE

A Thesis

In

Industrial Engineering

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for the Degree of

MASTER OF SCIENCE
in
INDUSTRIAL ENGINEERING

Approved by

Timothy I. Matis
Thesis Committee Chair

John Kobza

Milton Smith

Fred Hartmeister
Dean of the Graduate School

August, 2008

ACKNOWLEDGEMENTS

First and foremost, I would like to begin this section by thanking my parents, Casey and Vicki Phillips. Your love, support and the way you lead by example are the main reasons I have gotten to the point I am at today. I can only hope to someday be able to provide and care for my children the way you two have for me.

Before continuing on with family, I would like to thank Dr. Tim Matis for not only his academic and research guidance, but also for just sharing his personal advice and his general point of view on life. I look forward to working with you over the next few years.

As for the rest of my family, I would like to tell all of you how much I have appreciated all of your encouragement and supportiveness.

I would also like to thank my fiancée, Trista. The way you keep me going day in and day out with your upbeat attitude and dedication is something unmatched by any woman I have ever met. I'm also extremely grateful for her parents, Mark and Tina Wuensche, her brother Britt and her sister Kalee, as well as the rest of her very large family that I could not fit onto this page. The way you all took me in like family so quickly and have never been shy about encouraging words has definitely provided a great boost in my life.

Among a line similar to Dr. Matis, I must also say thanks to Dr. Ivan Guardiola. Through sharing your knowledge with me without holding anything back, you gave me the ability and a good knowledge base to begin this research. I also look forward to hopefully working with you over the next few years as well.

I want to also say thank you to Dr. John Kobza and Dr. Milton Smith for their roles on my thesis committee as well as their input and guidance involving my research.

Last but not least, I would like to express my gratitude to Kristen Godeke for the time intensive collection process of thousands of data points. Your hard work was very appreciated.

TABLE OF CONTENTS

ABSTRACT

This research is based on the inner workings of the Data Link Layer's Media Access Control Sub-Layer. Moreover, it is concerned with creating two dynamic variables with the Request To Send/Clear To Send mechanism that resides in the previously stated sub-layer. This document proposes the creation of a dynamic Short Retry Limit and a dynamic Long Retry Limit. This work goes on to give both a statistical and scientific backing for deciding how the Short and Long Retry Limits should adjust themselves, and what levels they should adjust themselves to, in order to ensure improved network performance when compared to current IEEE default settings.

Overall, this thesis presents a thorough introduction into a mobile ad-hoc network as well as an in-depth exploration of how the Media Access Control Sub-Layer performs its duties. It also addresses factors that affect performance such as fading environments and mobility. After all of the factors and inner workings have been elaborated on in great detail, the results presented in this paper will be concerning both how the dynamic levels were chosen as well as how the final dynamic Short and Long Retry Limits will be applied and methodically analyzed to understand the impact it had on improving the wireless mobile ad-hoc network performance.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Understanding Wireless Mobile Ad-Hoc Networks

Wireless communications have become more and more commonplace over the past decade, working their way not only into business, military and disaster applications, but also everyday life. With this wide acceptance comes the desire to push this technology to perform new applications, one of these being a mobile ad-hoc network.

The current common wireless networks consist of infrastructure-based systems, in which all data runs through one or many centralized points and is then disbursed to other users or devices accordingly. One problem that is being seen is the scenario in which a wireless infrastructure does not exist, such as natural disasters like Hurricane Katrina in New Orleans, LA where the infrastructure was wiped out, or man-made disasters such as the terrorist attacks on September 11, 2001 in New York City, NY, where the infrastructure could not handle the volume of data that needed to be transmitted. Situations such as these call for a more effective approach that is versatile to what is going on in the environment. The solution would be a properly functioning mobile ad-hoc network, which would allow for more than one pathway for information to flow to and from wireless devices.

To be used in these situations, the mobile ad-hoc network would have to have a performance level comparable to a wireless network with an almost unstrained infrastructure. In order to make this possible, the devices and mechanisms which make wireless networks with infrastructures possible must be modified to instead work amongst each other, which is what this paper seeks to do.

1.2 Related Works

Similar work has been done in this area. (Kim, Lee, & Yeom, 2008) proposes a dynamic Short Retry Limit (SRL) based on first differentiating between collision and route failure losses and then increasing the Short Retry Limit accordingly. In order to differentiate between the two, they apply a heuristic to control a created table of Short Retry Limits for neighboring nodes. If the current node overhears a neighboring node's packets, the current node then proceeds to increase the overheard node's SRL on its table based on the assumption that since it heard the node recently, it must still be nearby. If, during a small duration timer, it fails to overhear a neighboring node, then it begins to decrease that node's SRL. Through simulations, they showed they successfully cut down the amount of times route maintenance and re-discovery was initiated while still providing throughput competitive with statically setting the SRL at 30 when the route involved a hop count greater than seven.

In (Gunes, Hecker, & Bouazizi, 2003), an adaptive method of retransmitting RTS and CTS packets is proposed and tested. They labeled their creation the Dynamic Short Retry Limit (DSRL). The DSRL changes its value based on the density of the wireless nodes in a defined area. The basis of this adaptation is to increase the Short Retry Limit in high density areas and decrease the Short Retry Limit in low density areas. (Gunes, Hecker & Bouazizi, 2003) chose to set the maximum DSRL, dsrlmax, to 20 and the minimum DSRL, dsrlmin, to 7. Using an FTP agent over TCP, they found that a DSRL level higher than 20 failed to increase network performance, and a DSRL level lower than 7 offers a throughput worse than the IEEE 802.11 standard.

Through the mechanism presented in Kim et al., 2008, the goal was to decrease overhead due to routing maintenance, yet they create a table of Short Retry Limit values that must be updated and maintained, which may lead to more overhead than the network began with.

Both papers fail to address the effects of the Long Retry Limit, which should intuitively be called into question based on the reasoning that the amount of times a node tries to resend the data packet itself would play an almost similarly, if not more, significant role in network performance when compared to the amount of times the node tries to reserve the transmission space.

In this paper, a new design is proposed for creating adaptive, dynamic Short and Long Retry Limits. This new method involves transmitting nodes understanding and adjusting to their own mobility states. As the transmitting node's speed increase, both the Short Retry Limit and the Long Retry Limit begin to decrease in value. The values that they are allowed to decrease and increase themselves to are chosen through designed experimentation, and then applied and compared to the IEEE 802.11 Request To Send/Clear To Send mechanism's default settings. By choosing appropriate retry limits based on each individual transmitter's mobile state, this new routine also helps apply proper times for route maintenance. This is completed under the idea that the slower a node travels, the less likely it is that the route that current node is using will break, and vice versa. Based on that idea, it makes sense for the retry limits to be allowed to increase without as much worry of a link breakage, and just the opposite can be said for higher speeds.

The main discrepancies between this newly proposed method, the adaptive SRL that maintains a table, and DSRL are that this new method widens the scope by making both the Short and Long Retry Limits dynamic, and it deals with the problem of mobility first-hand by having the transmitter check its own speed to determine its own settings. This paper also takes into consideration the effects that different types of multipath fading can have on the network by applying it to both the default IEEE 802.11b settings as well as to the proposed scheme which the two aforementioned papers did not include.

CHAPTER 2
BACKGROUND INFORMATION

2.1 The Media Access Control Layer

In the OSI (Open System Interconnection) model, there are 7 layers involved in the use of a protocol and the processing of data.

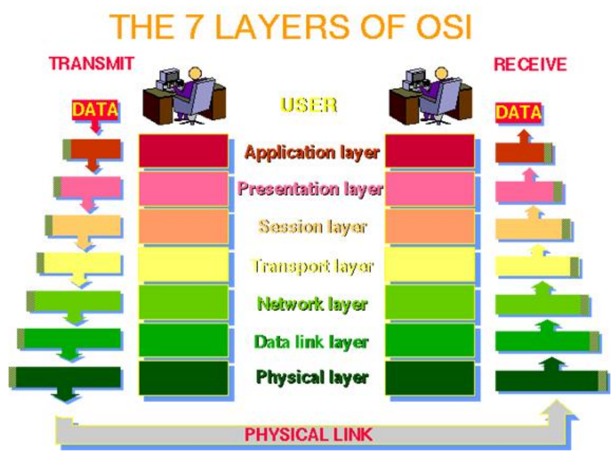

Figure 2.1. The 7 Layers of the Open System Interconnection Model

The layer of concern in this paper is the Data Link Layer (DLL), or more specifically, the first of the DLL's two sub-layers, the Media Access Control (MAC) layer. The MAC-layer has many responsibilities, including channel allocation procedures, protocol data unit addressing, frame formatting, error checking, and fragmentation and reassembly.

Within the MAC-layer are two means of control, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The DCF is used as an asynchronous data transmission function, which works best with delay insensitive data, while the PCF is an optional function that can be used in combination with the DCF for synchronous, or delay-sensitive, data (Aad & Castelluccia, 2001).

The IEEE 802.11 standard describes a Basic Service Set (BSS) as a group of stations that are under the direct control of a single coordination (i.e. a DCF or PCF). The area a BSS covers is known as the Basic Service Area (BSA), which can be exemplified by a cell in a cellular communications network . Using these definitions, an ad-hoc network, technically known as an independent Basic Service Set, can be properly defined as a collection of stations into a BSS for the internetworked communications without the aid of an infrastructure network (Crow, Widjaja, Kim, & Sakai, 1997). The communication between different Basic Service Sets can then be managed through creating links, or routes, using intermediate nodes to forward the source's data packet(s). These routes are created by protocols that administer routing techniques in the Network Layer, which will be discussed in the next sub-section.

In mobile ad-hoc networks (MANETs), the DCF, which operates on its own in an ad-hoc network, is chosen because it allows the nodes or devices to work in *contention mode*. This means that each transmitting device stands a better chance of getting an equal allocation of the channel over time, but that each node or device must compete for use of the medium at each frame of its transmission. Contention and fair channel access in a MANET must be accepted and dealt with accordingly, especially in high density networks. This being the case, using the PCF, which sets the Basic Service Set to *contention-free mode*, cannot be done because this setting relies on a centralized access point, which is not present in a MANET, to control the channel allocations.

The method used for DCF is the Carrier Sense Multiple Access with Collision Avoidance. The carrier sensing is done both at the physical and virtual levels. Physical carrier sensing is done by the source node by listening for other IEEE 802.11 users through packet detection and by checking usage of the medium through assessing received signal strengths from neighboring devices.

The virtual carrier sensing, which is this paper's main concern, is done through the sending of MAC Protocol Data Unit (MDPU) transmission length information which are attached to request to sends (RTS), clear to sends (CTS), and data frames (Crow et al., 1997)). The exchange of these packets allow the nodes to carry out proper insight without having to overhear neighboring transmissions and serves as a thorough secondary check to back up the physical carrier sensing.

2.2 The Routing Protocol

The routing protocol used in this paper is the Ad-hoc On-Demand Distance Vector (AODV) Routing protocol (Perkins & Royer, 1999). This is a reactive protocol which keeps a cache table of active routes and neighboring nodes. Each routing table entry consists of the following information:

1. Destination
2. Next Hop
3. Number of Hops
4. Sequence number for the destination (The order of the nodes in a given route)
5. Active neighbors for the current route
6. Expiration time for the route table entry

The AODV routing protocol only creates routes when they are needed. Route discovery is initiated by the transmitting node via propagating out a *Route Request* (RREQ) packet throughout the network. The neighboring nodes hear the RREQ and check the destination ID, known as the ip-address. If they are not the destination, they first check their cache table to see if they have knowledge of a route that leads to the destination. If they do not, they increase the hop count attached to the RREQ by 1 and then they proceed to rebroadcast it. If they do have a cached route that will lead to the destination, the node sends back a *Route Reply* (RREP) to the destination to notify the source that a route has been secured. Once the destination receives the route request and sends a route reply back to the source, a route is established for data to travel along and the transmitter can now prepare to send its data packet(s). After the route is established, nodes send out periodic *hello* messages in order to detect link failures. These *hello* messages are also used to keep their routing tables updated for future use.

CHAPTER 3
FACTORS AFFECTING NETWORK PERFORMANCE

3.1 Mobility

Mobility is one of the main benefits of wireless networks. It becomes an even greater asset in a mobile ad-hoc network because the mobility is no longer hampered by the need to be around a central access point. This benefit can also be a downfall. As shown in (Bhat et al., 2003), the bit error rate begins to increase as the speed of the nodes increase. The *bit error rate* (BER) is the ratio of bits with errors divided by the total number of bits received. This means that the closer the ratio is to 1, the worse the received quality.

Mobility failures occur when the transmitting node and the receiving nodes move out of range of each other. This can mean that either the source node is no longer in range of the destination and must then use the routing protocol to begin route discovery in order to bring into play intermediate "hops" to reach the destination, or it can represent the fact that intermediate nodes along a route have moved out of range of one another, causing the route to fail. To fix either of the previous scenarios, route maintenance must be initiated, which is controlled by the chosen routing protocol procedures.

3.2 Multipath Fading and Propagation Environments

Multipath fading can also play a large part in whether or not a packet properly arrives at its destination. Multipath fading is inherent in all wireless signals. It can not only degrade a signal and lead to false breakdowns in the link, but it can also amplify the signal and cause the creation of false, or improbable, routes, which will lead to an influx of lost data packets as well as a possible increase in delay until the nodes recognize the route does not work.

For this paper, two types of small-scale fading models, as well as one large-scale propagation model, were applied. Small-scale fading refers to the dramatic changes in signal amplitude and phase that can be experienced as a result of small changes that can occur due to changes in time and frequency, as well as being changes in the spatial position between the transmitter and receiver (Ware, Accessed May 28, 2008) (Punnoose, Nikitin, & Stancil, 2000). In large-scale fading, the strength of the signal is reduced and usually deals with large distances between the transmitter and the receiver.

The first small-scale fading type is Rayleigh fading. Rayleigh fading is considered to occur in urban environments, which include many walls, buildings and other obstacles. When the signals arrive at the destination, there are multiple indirect paths, none of which are dominant, which leaves the receiver without a clear desirable signal, thus creating a "worst case" scenario. After the signals have been deflected off of objects and obstacles, the true degradation occurs due to *multipath reception*. Multipath reception means that the device's antenna receives multiple variations of the same signal coming from different directions and angles, as well as at different times. All of these waves interfere with one another, either causing signal attenuation or signal amplification (Linnartz, Accessed May 27, 2008). The signals' instantaneous received powers can be interpreted as a sum of multiple, independent, stochastic random variables (Howald, Accessed May 27, 2008).

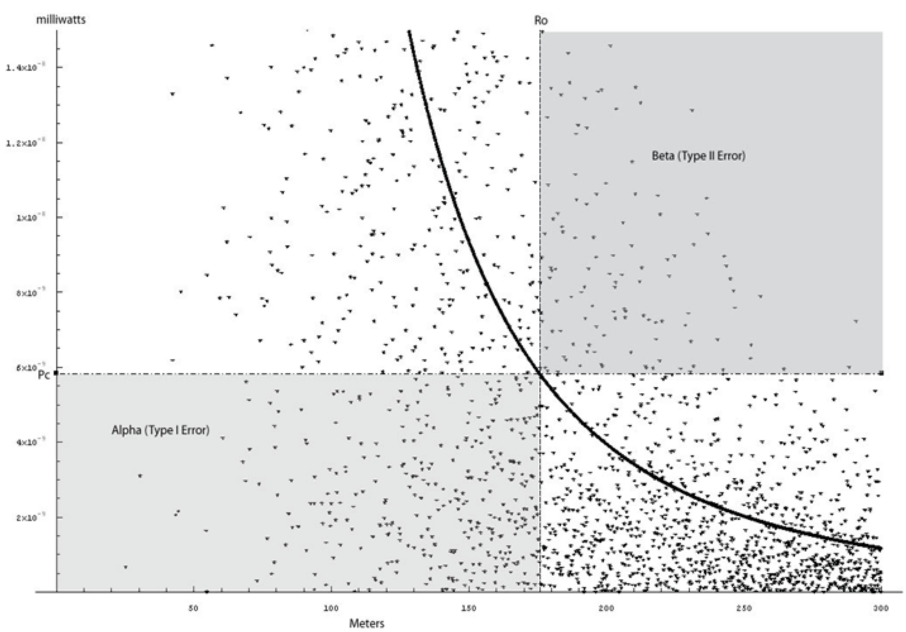

Figure 3.1. Rayleigh Fading Effects Scatter Plot

The Figure 3.1 (Guardiola & Matis, 2007) shows an illustration of the effects that Rayleigh fading can have just on small control packets. As shown, the Y-axis represents the received power in milliwatts while the X-axis represents the distance between the transmitting and the receiving node. The horizontal line, P_c , in the graph signifies the received power threshold, which is the minimum amount of received power the destination must obtain in order to understand that the transmitter is within range to establish a route. The vertical line, R_o , represents the maximum distance the transmitter and receiver can be from one another and still establish a route. The dark black curve represents the expectation of the received power at a given distance, and vice-versa.

The lower left quadrant represents the α, or type I, error. The observations in this quadrant show that the effect of Rayleigh fading caused signal attenuation, which made the received power lower than it should have been based on the transmitter's distance from the receiver. The consequence of this is the receiver will decide against creating a route with the transmitter due to the belief that the low power means the transmitting node is out of range for a stable link to be established. This error is not as detrimental to its counterpart, the type II error, because this problem can be easily fixed by simply resending the chosen protocol's control packet.

The upper right quadrant represents the β, or type II, error. These observations show that Rayleigh fading amplified their signal strength, causing the receiver to set up weak, if not non-working, routes based on the belief that the transmitter is well within range when in fact it is not. This is the worst of the two error types because setting up bad routes, as previously stated, leads to an influx of lost packets as well as increased delay times.

The second fading model applied to this work is Ricean fading. This fading model is considered a trait of short-term indoor propagation. In Ricean fading, the signals disperse into multiple paths, but when they reach the destination, there is a dominant path as well as the multiple weaker paths. The dominant path is possible because Ricean fading occurs when there is a line-of-sight between the transmitting and receiving nodes. The statistical implications of Ricean fading are similar to those of Rayleigh fading, except for the dominant path characteristic, which somewhat biases the Rayleigh distribution, generating a stochastic distribution about a more firmly characterized mean amplitude value (Howald, Accessed May 27, 2008). Figure 3.2(Howald, Accessed May 27, 2008) illustrates the Rayleigh and Ricean probability density functions. The last propagation model used is the Free

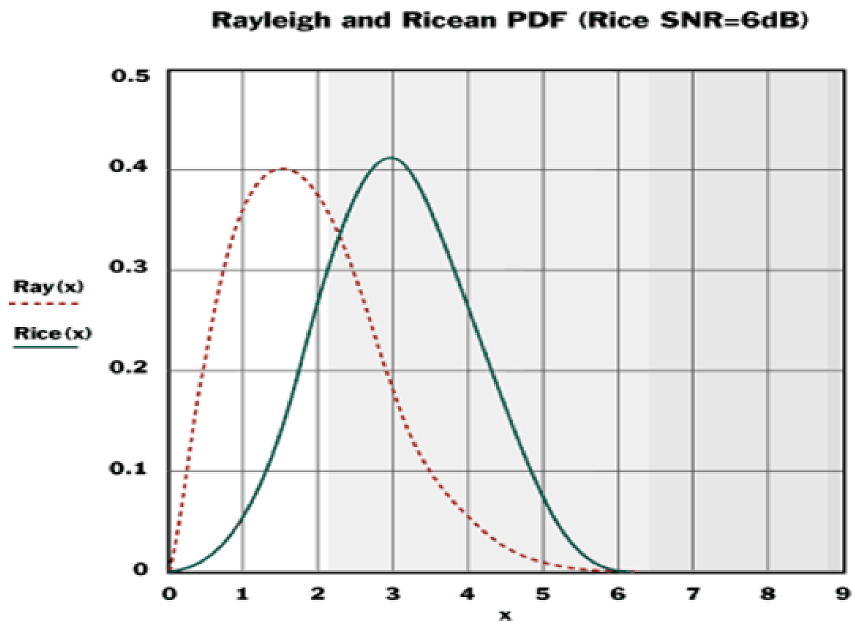

Figure 3.2. The Rayleigh and Ricean Probability Distribution Functions

Space Propagation model. This model assumes that the transmitting antenna and the receiving antenna are located in an otherwise empty environment. This includes no absorbing obstacles or reflecting surfaces. The influence of the earth's surface is also assumed to not be present (Linnartz, Accessed May 27, 2008). In this model, the energy emitted from the omni-directional antenna is assumed to spread out over the surface of a sphere, and the power received decreases proportionally to d^2, where d is the distance the signal is from the transmitter. This makes it possible to accurately judge the distance between the transmitting node and the receiving node based on the received signal power, and vice versa (Linnartz, Accessed May 27, 2008). The received signal power, which is what nodes use to measure packet quality in the NS-2.31 simulator, is calculated in the free space propagation model according to H.T. Friis' equation:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{3.1}$$

where P_t is the transmitted signal power, G_t and G_r are the antenna gains of the transmitting and receiving nodes respectively, L is the system loss which must be equal to or greater than one, and λ is the wavelength. The common and widely accepted NS-2 simulator settings are $G_t = G_r = L = 1$ (Fall & Varadhan, 2002).

3.3 Packet Collisions

Packet collisions can occur due to two factors:

1. Multipath Fading
2. A Hidden Terminal

Both of these factors have already been extensively discussed. To quickly revisit the second factor, neighboring nodes that begin transmitting data packets at the same time run the risk of suffering large numbers of lost packets due to collisions caused by sharing a medium. To combat this problem, it was previously explained that each node does physical carrier sensing to ensure that the medium is idle, and then does virtual carrier sensing through the RTS/CTS mechanism to reserve transmission space. The virtual carrier sensing procedure is specifically designed to eliminate the hidden terminal problem. The use of virtual sensing and the RTS/CTS mechanism aspect will be more thoroughly covered in Chapter 4.

CHAPTER 4
PROBLEM STATEMENT

The main goal of wireless networks is to have throughput and overall performance generally equitable to its wired counterpart. The reason this can become such a daunting task in mobile ad-hoc networks can be attributed to three aforementioned factors:

1. Mobility of the transmitting and receiving nodes
2. Multipath fading
3. Packet Collision

Due to these relatively uncontrollable factors which have been previously and thoroughly discussed, network modifications must be completed to achieve better network performance through adaptations. This research seeks to do just that through network simulations, Designed Experimentation and implementation focused on the Media Access Control layer's Request To Send/Clear To Send mechanism.

4.1 The Request To Send/Clear To Send Mechanism Components

The Request To Send/Clear To Send (RTS/CTS) mechanism resides in the MAC-Layer. When the Distributed Coordination Function with Collision Avoidance is in use, as previously mentioned, it is in contention mode. This means that each node must first gain the right to transmit its data through the two aforementioned sensing methods. Aside from physical carrier sensing, which checks for idleness in the medium, the other part of getting transmission access and reserving channel bandwidth comes from sending and receiving *Request To Send* and *Clear To Send* control frames as well as the MAC-layer information attached to each of them.

The main goal of this mechanism is to eliminate or minimize the effects of the hidden terminal problem. The hidden terminal problem is illustrated in the following set-up shown in Figure 4.1 (Weinmiller, Woesner, Ebert, & Wolisz, 1995):

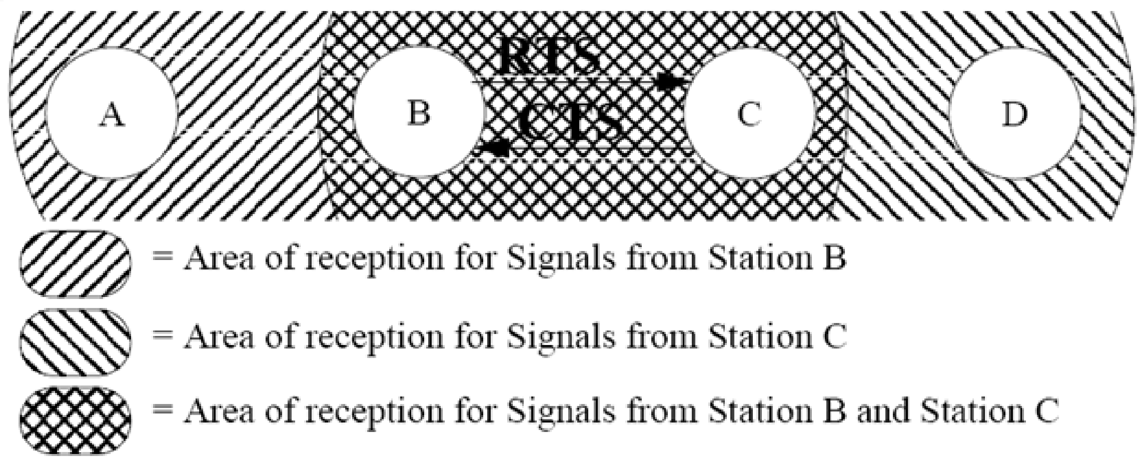

Figure 4.1. The Hidden Terminal Illustrated

In Figure 4.1 (Weinmiller et al., 1995), it can be seen that there are four nodes in close proximity to one another. This figure is the hidden terminal problem in its simplest state. For this scenario, node B wishes to communicate with node C. The problem with this scenario is that B is not in range of D to be aware of its presence, making node D a hidden terminal to B. Also note that node D is in range of node C. This means that if C fails to notify D of its transmissions with node B, then node D's transmissions, whether to C or anyone else, can corrupt and collide with node B's transmissions to node C. This same observation also holds true for node A with respect to nodes B and C. If node B fails to alert node A of its desired communication with node C, the transmissions from A can reach node B and corrupt or collide with node C's transmissions to B.

4.2 How The RTS/CTS Mechanism Works

When the source node has a data packet arrive in the transmission queue, it first checks the size of the data packet. If the data packet is larger than the *Request To Send Threshold* (RTS Threshold), then the RTS/CTS mechanism is activated. If the packet size is lower than the RTS Threshold, the packet is sent after the node verifies the medium is idle through only physical carrier sensing. After it is determined that the packet size exceeds the RTS Threshold, the transmitting node then checks to see if a route is available. If a route does not exist, route discovery is initiated by the chosen protocol as previously discussed. Once a route has been detected or created, the source node will first carry out physical carrier sensing by listening for other nodes transmitting packets on the same medium. Once the medium is considered idle, the transmitting node waits another time interval The time period the transmitting node waits is known as the Distributed Coordination Function Interframe Space time interval.There are two *Interframe Space* (IFS) time intervals that will be used in this paper:

- *Distributed Coordination Function Interframe Space* (DIFS)
- *Short Interframe Space* (SIFS)

Interframe space time intervals are used to control and delegate access priority in a wireless network (Crow, Widjaja, Kim & Sakai, 1997). The Short Interframe Space is the shortest time interval, followed by the Distributed Coordination Function Interframe Space. After the node has waited the DIFS time period to reach zero, it then waits for an added random time known as the *Backoff Time*.The Backoff Time is determined using the following equation:

$$BackoffTimer = [2^{2+i} \times rand()] \times SlotTime \tag{4.1}$$

In Equation 4.1, *SlotTime* represents a function of physical layer parameters, and *rand()* is a random function with a uniform distribution of [0,1] (Aad & Castelluccia, 2001). This random backoff time helps avoid collisions that may happen when two nodes have their DIFS time periods end at the same time.

This being the case, this means that nodes that are only waiting the SIFS time interval will have priority over those waiting the DIFS because the nodes using the SIFS will gain access to the wireless channel sooner. This scenario can occur when one node is waiting to send an Acknowledgement packet, which uses the SIFS, while a neighboring node is waiting to send a Data packet for the duration of the DIFS and Backoff time intervals. The ACK packet will receive priority over the data packet because its node is given the first opportunity to access the medium.

Once the source node appropriately determines the medium is idle and has awaited its allotted time intervals, the source will then desire to send a data packet along the created links. In order to begin doing so, the source node first transmits a Request To Send packet to all of its surrounding neighbors, including the destination or the next intermediate node in the route in its Basic Service Area. The distinction that these specific control frames, both the RTS and CTS packets, are only transmitted to the source node's BSA is important in understanding that these control frames are not meant to be relayed by other nodes throughout the network, but instead are meant to simply be read and understood by only the neighbors in the packet capture area.

After the RTS is propagated out to the source node's neighbors, the near-by nodes will receive the control frame and interpret it. Each node that receives the Request To Send and reads from the packet will make a decision: If it is not the desired destination, it will then proceed to read the MAC Protocol Data Unit from the RTS header. This header will provide the non-desired node information on how long the transmission of both the data packet and acknowledgement packet are expected to need to take place. Once the node reads the duration information, it sets its own *Network Allocation Vector* (NAV) accordingly. By setting its NAV according to this information, the non-destination node will now have an internal record of how long the transmission should take as well as how long it should wait to contend for the medium.

The second decision a node can make when receiving the RTS is that it is in fact the intended destination of that particular control frame. If this is the case, the destination then waits a duration known as the *Short Interframe Space* (SIFS) to assure that the medium is idle through physical carrier sensing. Once this is satisfied, the destination will respond with a Clear To Send. The CTS control frame also contains a MAC Protocol Data Unit with information such as transmission duration attached to its header. The neighboring nodes will also overhear the CTS and update their NAV, or expected transmission duration, accordingly. The usefulness of connecting the transmission duration to the responding CTS comes in the fact that it not only provides up-to-date information for those that received the RTS previously, but it also allows for a second sweep of the surrounding area to ensure that everyone is aware of the upcoming transmission. By creating two different opportunities for surrounding nodes to hear and understand the necessary

Network Allocation Vector, this mechanism helps greatly reduce the chances of having a hidden terminal.

Following the source node's reception of the CTS, the source node will then make the assumption that the medium is clear for transmission. The source node will then continue on and transmit its data frame to the destination node or next intermediate node in the route. After receiving the data frame, the destination or intermediate node will respond by sending an *Acknowledgement Packet* (ACK). This serves as a receipt of reception between the transmitting and receiving node. When the source node gets the ACK back, it knows the transmission was a success and may now go back and begin contending for channel bandwidth for its next data frame transmission. Figure 4.2 (Weinmiller et al., 1995) illustrates the order of both the wait times as well as the sending of the Request To Send and Clear To Send frames as well as the Data packet and its Acknowledgement packet receipt.

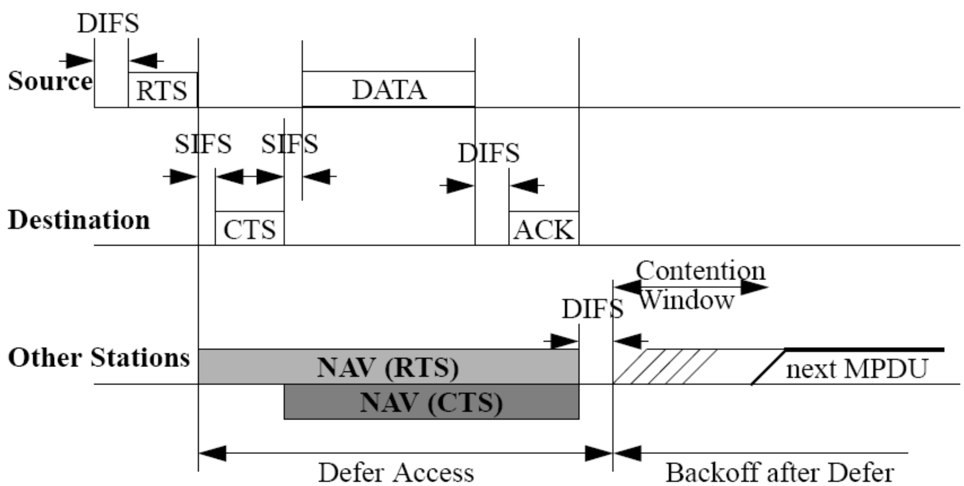

Figure 4.2. A Visual for the Packet Exchanges

(Weinmiller et al., 1995) showed through simulations that although the RTS/CTS mechanism is not perfect, it significantly improves network performance, especially in the presence of the hidden terminal problem and large data packet sizes. It also presented findings that show the RTS/CTS mechanism provides better throughput for larger packet sizes, as opposed to leaving the mechanism off.

4.3 The Short and Long Retry Limits

The RTS/CTS mechanism has two ”retry limits” associated with it:

- *Short Retry Limit* (SRL)
- *Long Retry Limit* (LRL)

The Short Retry Limit is associated with the transmitting node sending out the Request To Send control frame and receiving back a Clear To Send control frame from the destination. The SRL comes into play when the transmitting node sends out an RTS to the destination, and, after a small duration of time, the transmitting node does not receive back a CTS from the destination. Once that small duration, which is measured in microseconds, has elapsed, the transmitting node begins *retrying* to send the RTS. The transmitting node will attempt to resend the RTS to the destination, in the hopes of getting a CTS back, up to a certain number of attempts known as the Short Retry Limit. The amount of resend attempts is recorded in the *ssrc* counter variable within each packet. These variables are then compared to their respective retry limits to decide whether to retry sending the packet or if it is time to discard the packet. Once the Short Retry Limit is reached (i.e. the *ssrc* is equal to the SRL), the corresponding packet is discarded, and the transmitting node believes the destination node to no longer be accessible. Since the transmitting node deems the destination node out of reach, the route maintenance procedures are initiated (Kim et al., 2008). The current default value for the Short Retry Limit is set statically at 7 without any discoverable documentation or explanation, including (Mckay et al., n.d.), which is the US patent that was issued for this innovation.

The Long Retry Limit is associated with the trading of a data packet and an Acknowledgement packet. Now, assuming the RTS and CTS control packets are successfully exchanged, the transmitting node will assume the medium to be clear and send its data packet. After the source sends its data packet, it then awaits a small duration of time to receive back an Acknowledgement packet. If the duration of time expires without receiving the ACK packet back, the transmitting node will then attempt to resend the data packet. If the transmitting node tries up to a certain number of times known as the Long Retry Limit and never receives an Acknowledgement packet back, it makes the assumption that the destination is no longer available. The amount of attempts that have been made to send the Data packet is recorded in the each packet's *slrc* counter variable. This variable, similar to the SRL's counter variable, is compared to the Long Retry Limit to decide to either continue trying to resend the data packet or to discard it and move on to the next packet in the queue. Once this assumption is made, the data packet is dropped from the queue and the route maintenance activities of the chosen protocol ensue. The Long Retry Limit has a default value statically set at 4. (Mckay et al., n.d.) failed to provide any reasoning, scientific or otherwise, for determining and setting this static limit as well.

CHAPTER 5
RESEARCH APPROACH

The purpose of the research is to design a dynamic retry limit mechanism which can adjust itself based on the transmitting node's mobile state. In order to derive the dynamic settings, this paper considers multiple retry limit values and then using Designed Experimentation to test their significance. Supposing the dynamic settings do have a significant effect on the network metrics chosen, the next step involves choosing the levels that provide the best network performance for specific mobile states. Once these dynamic limits are chosen, they must then be statistically compared to the default provided in the IEEE 802.11b wireless network settings. To properly approach such a problem, an understanding must first be made about the shortcomings often seen in mobile ad-hoc networks. There are three such problems that contribute to insufficient MANETs.

5.1 Mobile Ad-hoc Network Shortcomings

The first problem is the well-known difficulty of poor throughput performance. This can be attributed to the mobile nodes' inabilities to maintain proper routes for data to travel along. Another role player in this problem is a node's failure to adjust its personal settings to the state of the network, even if it is only in its own Basic Service Area. Increased network performance through a node's recognition of its mobile state is the main goal of this work.

The second problem lies in the non-scalability of the routing protocols. As the size of the network increases the number of users it is servicing, protocols such as AODV, the protocol chosen for this paper, have been proven in work with which I have been involved (Guardiola & Matis, 2007) to lack the ability to maintain network performance. (Guardiola & Matis, 2007) showed through simulations that AODV's performance begins to significantly decrease as the number of nodes within the network increases from twenty-five nodes to fifty, and then one-hundred nodes. The scalability deficiency is believed to be attributed to the large amount of overhead involved in such protocols, which leads to unnecessary congestion within the network.

The third and final problem to be discussed is the low fidelity simulation environments that are typically used to test the abilities of newly created protocols and embedded mechanisms. Many of the creators fail to take a signal's behavioral characteristics into consideration within different environments. The nature of these signals can often be emulated using available fading models such as the one presented in (Punnoose et al., 2000). However, most of the protocol and mechanism creators opt for more of a "perfect world" setting such as the Free Space propagation model. A model such as this assumes there are no other objects, including the surface of the earth, in the environment to cause effects such as multipath fading, signal attenuation and signal amplification. By not accounting for such possible variations in an environment when testing a protocol or mechanism, a large amount of uncertainty will remain about its possible performance in the real world. This gap of uncertainty can lead researchers astray by making the assumption that if the protocol or mechanism performed well in computer simulations, it is likely to do the same in true, tangible environments. This shortcoming will be addressed through creating high-fidelity simulations using two well-researched fading models as well as the Free Space propagation model.

5.2 The Simulation Set-Up

To even begin the analysis, a simulator had to be decided on. The simulator NS-2.31 was chosen to perform the simulations depicted in this paper. NS-2.31 is a discrete event simulator created for researching network behaviors and performance . This simulator is an ongoing effort of research and development, which got its start as NS in 1989 as an alternative to the REAL Network Simulator. In 1995, the NS simulator continued its progress with the support of the Defense Advanced Research Projects Agency (DARPA) through the Virtual InterNetwork Testbed (VINT) Project at the Lawrence Berkeley National Laboratory, Xerox Palo Alto Research Center, Inc., University of California, Berkeley, and the University of Southern California's Information Sciences Institute (NS-2.31, Accessed May 25, 2008).

Each simulation that was carried out consisted of twenty-five fully mobile nodes deployed in an environment of constant size. The use of twenty-five nodes was chosen based on the work and results presented in (Bruno, Conti, & Gregori, 2002). In their work, they showed that the Request To Send/Clear To Send mechanism provides better network performance over the Basic Access Method when the number of nodes present in the network reached an amount greater than 20 nodes. Every simulation had 10 transmitter/receiver pairs which were initially chosen and remained constant throughout this research. The other 5 mobile nodes were put into each simulation to serve purely as intermediate, or “hop”, nodes.

The method of sending packets was chosen to be Constant Bit Rate (CBR) packets. The ten transmitting nodes created new data packets to be sent every .2 seconds, or 5 packets per second, with each data packet being 512 bits in length. This method was chosen in order to give each transmitting node the same amount of opportunities to send data packets. This scenario also offers somewhat of a worst-case situation by creating a potential for high congestion within the network. Such a situation offers a thorough and rigorous form of investigation into the performance of the proposed dynamic mechanism.

Their movements were generated using the *setdest* function created by Carnegie Mellon University, which is embedded into NS-2.31. This function is useful when it is not feasible to explicitly control numerous nodes’ movements. It begins by having the user specify desired criteria and has the following layout:

./setdest -n \<number_of_nodes\> -p \<pausetime\> -s \<maximum speed\> -t \<simulation time\> -x \<maximum x coordinate\> -y \<maximum y coordinate\> \> \<output directory\>/\<scenario-file\>

The *setdest* function initially distributes the nodes in a uniform pattern throughout the environment stated in the function parameters. The maximum x and y coordinates were both set to 500 m, giving the 25 nodes a 500 m \times 500 m area within which to navigate around according to the *setdest* movement file.

In order to coordinate each node’s movement for the specified simulation duration, the function uses a Random Waypoint Model. In the Random Waypoint Model, each node travels in a ”zigzag” pattern from one waypoint to another. The waypoints are distributed throughout the stated environment size. When a node reaches a waypoint, the model also includes a pause, or “thinking” time, in which the node may stop at that point for a specified duration before continuing on to the next waypoint (*Random Waypoint Model*, Accessed June 3, 2008). The pause time in this paper's simulations was set to 0 seconds to ensure the worst-case scenario portion of constant movement. The simulation time was set to 350 seconds to allow plenty of data exchange opportunities without extending the simulations to a point where the simulation time becomes excessively large. After the node reaches the waypoint, its velocity is then changed based on a uniform distribution with the parameters [0 , Maximum Speed]. The maximum speed is changed between simulations, with values set at 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s, and 30 m/s. A movement scenario was created for each speed setting. After their creation, each movement scenario had different combinations of factors applied to it, including three fading/propagation effects and MAC-Layer setting adjustments which will be discussed later in the paper.

The Rayleigh and Ricean fading effects were applied using the small scale fading effects package documented in (Punnoose et al., 2000). The use of this fading package allows for high fidelity simulations that better signify the random nature wireless signals experience in real world settings. The Ricean-K factor was set to 6, which follows the settings in (Punnoose et al., 2000), where K is the Ricean parameter equal to the amplitude of a given signal. This means that in the simulations involving Ricean fading, the signal with the dominant path will provide a received power level of six times the amount of the average signal powers of the non-dominant signals. The maximum Ricean propagation velocity was set to 2.5 m/s. For the simulations involving the Rayleigh fading model, the Doppler frequency was eliminated from the Otcl simulation script, and the Ricean-K factor was set to $K = 0$. This is possible due to the fact that setting K to zero turns the Ricean distribution into a specialized case of the Rayleigh distribution (Punnoose et al., 2000).

The wireless card specifications employed in the simulations are that of the Orinoco 802.11b wireless card. This wireless card operates at a frequency 2.472 GHz with a bandwidth of 11 Mbps. The following is an example of the Otcl script using Ricean fading in which the Orinoco 802.11 wireless card specifications, as well as the routing protocol, environment size and MAC-layer type are set:

```
set val(chan) Channel/WirelessChannel ;# Channel type
set val(prop) Propagation/Ricean ;# Radio propagation model
# Values of the 802.11 card
Phy/WirelessPhy set L_ 1.0 ;# System Loss Factor
Phy/WirelessPhy set freq_ 2.472e9 ;# Channel-13. 2.472GHz
Phy/WirelessPhy set bandwidth_ 11Mb ;# Data Rate
Phy/WirelessPhy set Pt_ 0.031622777 ;# Transmit Power
Phy/WirelessPhy set CPThresh_ 10.0 ;# Collision Threshold
Phy/WirelessPhy set CSThresh_ 5.011872e-12 ;# Carrier Sense Power
Phy/WirelessPhy set RXThresh_ 1.15126e-10 ;# Received Power Threshold
set val(netif) Phy/WirelessPhy ;# Network interference type
set val(mac) Mac/802_11 ;# Mac Layer type
set val(ifq) Queue/DropTail/PriQueue ;# Interface Queue type
set val(ll) LL ;# Link Layer type
Antenna/OmniAntenna set Gt_ 1 ;# Transmit Antenna gain
Antenna/OmniAntenna set Gr_ 1 ;# Receiver Antenna gain
set val(ant) Antenna/OmniAntenna ;# Antenna Model
set val(ifqlen) 50 ;# Max number of packets in ifq
set val(nn) 25 ;# Number of Mobile Nodes
set val(rp) AODV ;# Routing Protocol
set val(x) 500 ;# x dimension of topography
set val(y) 500 ;# y dimension of topography
set val(stop) 350 ;# Time of simulation end
set val(move) "/home/ns-allinone-2.31/aaron/StatAnalysis/exp1ms15"
set val(traff) "/home/ns-allinone-2.31/aaron/StatAnalysis/exptraff"
set val(RiceanK) 6 ;# Ricean K factor
set val(RiceanMaxVel) 2.5 ;# Ricean Propagation MaxVelocity Parameter
```

All of the above *set val()* interpretations and purposes were understood from (Fall & Varadhan, 2002). The Otcl script above, as well as in Appendix B, provide comments throughout for a better understanding as well as for easier reading.

The variable *RXThresh*, which is set in the Otcl simulation script, helps the user customize the communication range of wireless nodes. The Received Power

Threshold, as well as other thresholds, can be computed using a C program that is packaged with NS-2.31. The input for this executable computation is as follows:

threshold –m <propagation-model> [other-options] distance

where *<propagation-model>* can be FreeSpace, TwoRayGround, Ricean and so on, and the *distance* variable is the desired communication range in meters. The variable *[other-options]* is used to specify parameters other than their default values. The other available options with the *threshold* function are as follows:

1. *-pl* (Path Loss Exponent)
2. *-std* (Shadowing Deviation)
3. *-Pt* (Transmit Power)
4. *-fr* (Frequency)
5. *-Gt* (Transmit Antenna Gain)
6. *-Gr* (Receive Antenna Gain)
7. *-L* (System Loss)
8. *-ht* (Transmit Antenna Height)
9. *-hr* (Receive Antenna Height)
10. *-d0* (Reference Distance)

The Received Power Threshold, *RXThresh*, value selected for the simulations involved in this paper is the value associated with the Orinoco 802.11b wireless card, which is 1.15126x10-10 W.

The Carrier Sensing Power Threshold, *CSThresh*, is the minimum amount of power a node must receive from overhearing other nodes in the network in order to "sense" them. If a node overhears a neighboring node's packets, and the received power is greater than the Carrier Sensing Power Threshold, then it acknowledges that the medium is not idle through physical carrier sensing. The *CSThresh* in the above Otcl script was not set using the *threshold* function as well. If the *threshold* function was used in setting the *CSThresh*, the user would then be able to decide how far away two nodes can be and still sense each others' transmissions in a Free Space propagation model. The *CSThresh* is held constant for all of the simulations, including when the Ricean and Rayleigh fading models incorporated in the network instead of the Free Space propagation model. This means that when the simulations in this paper depart from the Free Space propagation model and begin venturing into multipath fading models, the *CSThresh* value may not have as much control in alienating nodes that are a large distance from one another, and vice versa. The *CSThresh* value for the Orinoco card is 5.011872x10-12 W.

The Capture Threshold, *CPThresh*, is the capture threshold for a receiving node. This is used when multiple packets collide at the receiving node. When this situation occurs, only the first packet can be successfully received and processed under the condition that the received power of the first packet's signal is greater than any of the other colliding packets by a minimum value known as *CPThresh*. The Orinoco wireless card setting for this value is 10.0dB (Wang, 2003).

The Transmission Power, *Pt*, is the amount of power the signal has as it leaves the transmitter's device. The Orinoco wireless card setting for the transmission power is 3.1622777x10-2 W.

The variables *Gt* and *Gr*, which represent the transmitting and receiving antenna gains respectively, are set at 1. By setting these two variables at 1, the energy put into the antenna to send the signal is equal to the amount of energy the signal has when it leaves the antenna.

CHAPTER 6
RESULTS

The results represented in this chapter are presented using the approach of Designed Experimentation and as well as the use of statistical hypothesis testing . The theory behind and the proper use of the aforementioned techniques can be found in (Myers, Montgomery, & Vining, 2002) and (Devore, 2004) respectively.

6.1 Initial Linear Model Set-Up

In order to understand the importance of each factor being dealt with in this research, an *Initial Linear Model* was created. This model would help shed light not only on the significance of the effects created by the current factors, but also if other factors were being overlooked. An indicator that other factors were being overlooked would come from the R^2 value. If this value is low, that means the current model set-up is not adequately explaining deviations of the data from the linear regression line, which indicates that other influential factors need to be identified.

The Initial Linear Model equation is shown below:

$$Y_{ijklm} = \mu + \tau_i + \beta_j + \gamma_k + \delta_l + \alpha_{m(l)} + \tau\beta_{ij} + \tau\gamma_{ik} + \tau\delta_{il} + \tau\alpha_{im(l)} + \beta\gamma_{jk} + \beta\delta_{jl} + \beta\alpha_{jm(l)} + \gamma\delta_{kl} + \gamma\alpha_{km(l)} + \varepsilon_{ijklm(l)} \quad (6.1)$$

The Greek symbols from the above linear model represent the factors as follows:

- τ_i - Short Retry Limit (5 Levels)
- β_j - Long Retry Limit (4 Levels)
- γ_k - Fading Type (3 Levels)
- δ_l - Maximum Speed (6 Levels)
- $\alpha_{m(l)}$ - Route Nested Within Maximum Speed (60 Levels)

In this set-up, the Short Retry Limit, Long Retry Limit, Fading Type and Maximum Speed factors are fixed effects while the Route, which is nested within Maximum Speed, is a random effect. The Route effect is said to be random because each of them is based on the Random Waypoint Model previously discussed in an earlier chapter. Speed can be said to be fixed because it follows the Uniform Distribution with the parameters (a,b), where a in this case is zero and b is the selected Maximum Speed chosen for a particular simulation. From these parameters, an average speed for the nodes in the simulation can be easily obtained using the knowledge that it comes from the Uniform Distribution:

$$Speed_{avg} = \frac{a+b}{2} \tag{6.2}$$

Properly obtaining this speed average based on the correct distribution will be of great service later in this research for deciding which Retry Limit levels will be applied dynamically to the different speed ranges.

6.2 Initial Linear Model Findings

The Initial Linear Model was completed using the statistical software known as *Minitab 15*®. The response variables collected from the simulations are as follows:

- Constant Bit Rate Packets *Sent*
- Constant Bit Rate Packets *Lost*
- Constant Bit Rate Packets *Received*

Each response variable involved a collection of 3600 data points. This large amount helped ensure thorough analysis as well as lending a helping hand towards the data's normality. For the full results of the Initial Linear Model, please refer to Table 6.1 below.

From Table 6.1, it can be noted that for the responses of *CBR Packets Sent* and *CBR Packets Lost*, all of the factors, including their interactions with one another, are highly significant in their effect on those two response variables. It is also noticeable that for the response of *CBR Packets Received*, all of the factors and their interactions are significant with the exception of the interactions between the Long and Short Retry Limits as well as the interaction between Speed and the Long Retry Limit.

Table 6.1. Initial Linear Model p-Values

Initial Linear Model p-Values			
Source	CBR Packets Sent	CBR Packets Lost	CBR Packets Received
Speed	0.000	0.000	0.000
Route (Speed)	0.000	0.000	0.000
LRL	0.000	0.000	0.000
SRL	0.000	0.000	0.000
Fading Type	0.000	0.000	0.000
LRL*SRL	0.000	0.000	0.320
LRL*Fading Type	0.000	0.000	0.000
SRL*Fading Type	0.000	0.000	0.000
SRL*Route (Speed)	0.000	0.000	0.000
LRL*Route (Speed)	0.000	0.000	0.000
Fading Type*Route (Speed)	0.000	0.000	0.000
Speed*SRL	0.000	0.000	0.000
Speed*LRL	0.001	0.000	0.705
Speed*Fading Type	0.000	0.000	0.000
R-Squared Values	77.18%	88.69%	89.19%

The lack of significance between the interaction of the Short and Long Retry Limits for this response must be handled accordingly. The fact that these two factors do not interrelate means that when choosing the levels that produce the desired CBR Packets Received response, each Retry Limit must be looked at separately. To do so, this will mean that all of the different Short Retry Limit levels must first be analyzed without regard to the Long Retry Limit levels that are paired with them, and then the Long Retry Limit levels must be analyzed without consideration of the Short Retry Limit levels corresponding with them in each simulation.

The Linear Models (LMs) based on the CBR Packets *Sent, Lost* and *Received* all produced high R^2 values of 77.18%, 88.69%, and 89.19% respectively, which shows that the factors identified in the model, as well as their interactions, adequately describe the data represented by each model.

The Residual Plots for each of the responses are shown in Figures 6.1, 6.2, and 6.3. Figure 6.1 on the next page shows decent normality with a few outliers to the left, which is also indicated by the somewhat negative skewness of the histogram. This figure also shows that there is good constant variance and a definite independence of time for the observations.

Figure 6.2 shows a more solid normality plot aside from one semi-large outlier whose accuracy has been checked and verified. The other plots in this figure show

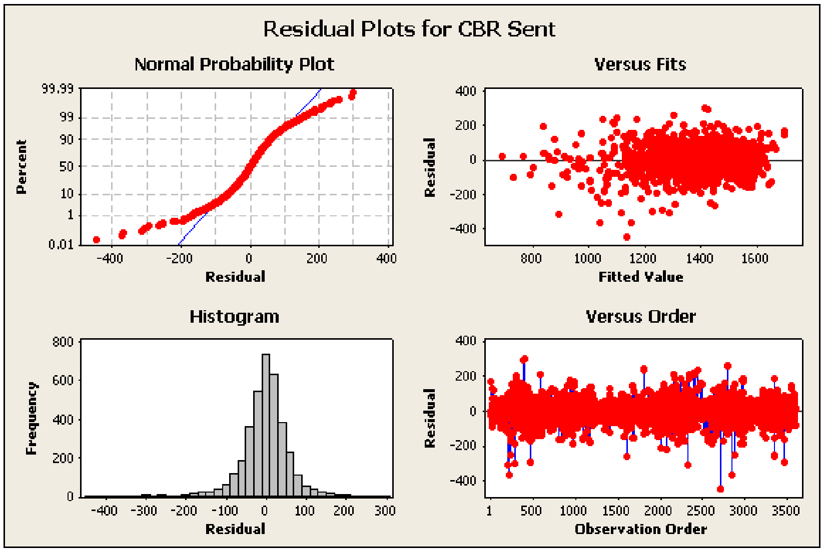

Figure 6.1. Residual Plots for CBR Packets Sent

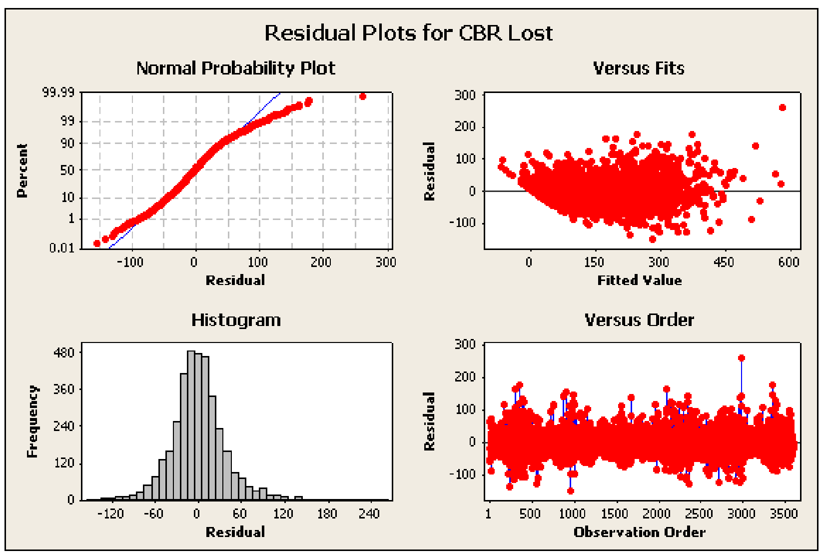

Figure 6.2. Residual Plots for CBR Packets Lost

that their is constant variance with no dependence of time for the data collected.

Figure 6.3 presents a few more outliers in its Normal Probability Plot than previous seen. This can somewhat be explained due to the many factors such as fading that may effect each simulation set-up differently. The rest of the figure shows that this response variable has solid constant variance with an independence

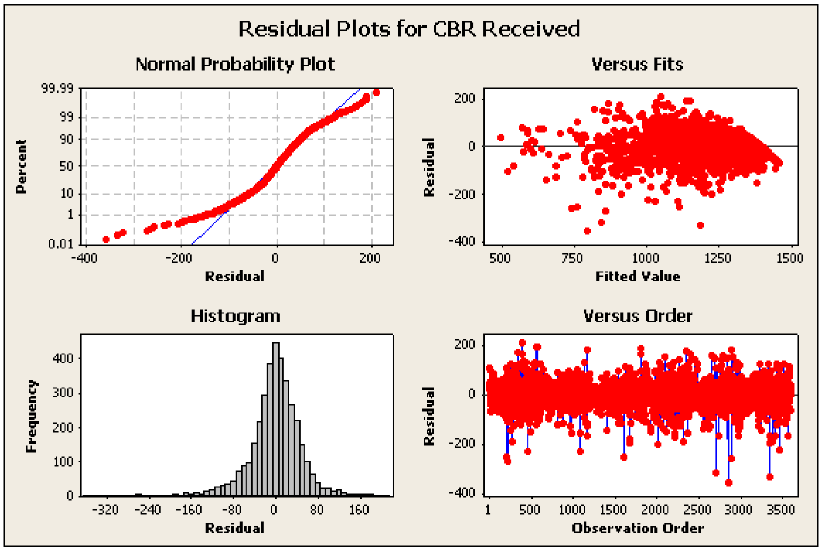

Figure 6.3. Residual Plots for CBR Packets Received

of time for the observations.

6.3 Focusing the LMs

The next step in this analysis involves the arduous task of focusing on the results within certain factors such as Maximum Speed and Fading Type. In order to begin doing so, the random effect of the *Route* had to be removed. This was done by taking the relative differences within each Maximum Speed and Fading Type setting combination within each route for each response. To gain a better understanding of what this means, it must be made clear that there are 10 routes for each of the 6 Maximum Speed scenarios. It must also be understood that, for instance, Route "1" of a Maximum Speed scenario of 5 m/s is not the same as Route "1" for the Maximum Speed scenario of 10 m/s and so on. This is because of the use of the *setdest* function that was previously explained in Chapter 5.2.

For a more in-depth example, supposed we were to look within the Maximum Speed level of 5 m/s, and within this speed level, we choose to look at only the simulations that only had a Fading Type level of 1 (Ricean Fading) applied to it. This means that we are only looking at simulations that are running at a maximum speed of 5 m/s under the Ricean Fading model. Doing this focuses the scope to 20 simulations which consist of the same 10 routes. The difference between the 20 simulations then becomes the Short and Long Retry Limits that were applied. Now, to get the desired transformed response, each observation is processed through the following formula:

$$Y_{ijklm(l)} = \frac{x_{ijklm(l)} - \left[\frac{\sum_{i=6}^{20} \sum_{j=2}^{10} x_{ijklm(l)}}{n}\right]}{\left[\frac{\sum_{i=6}^{20} \sum_{j=2}^{10} x_{ijklm(l)}}{n}\right]} \tag{6.3}$$

where $Y_{ijklm(l)}$ represents the relative difference for that observation, i signifies the Short Retry Limit level, j corresponds to the Long Retry Limit level, k symbolizes the Fading Type, l represents the Maximum Speed level, and *m(l)* stands for the number of the Route nested with a given Maximum Speed level.

Once Equation 6.3 is carried out, the relative difference response it yields can be easily interpreted. Equation 6.3 will produce a response of the nature $1 \geq Y_{ijm} \geq 0$. This decimal, when multiplied by 100, will now show how much of a percentage that particular response was above or below the average of all of the other routes of the same m value that had different Short and Long Retry Limit levels applied to them. This eliminates the randomness of the routes while also allowing the viewer to see which Retry Limit levels performed better for each Maximum Speed and Fading Type levels.

A visual of the tables that were created using the relative differences are available in the next subsection.

The "focused" Linear Model equation is of the form shown below:

$$Y_{ij} = \mu + \tau_i + \beta_j + \tau\beta_{ij} + \varepsilon_{ijk} \tag{6.4}$$

where τ_i represents the Short Retry Limit levels, β_j represents the Long Retry Limit levels, and Y_{ij} signified the the response which has been transformed into relative differences. For future sections, the interaction between the Short and Long Retry Limit levels has been omitted due to the lack of significance (p-values ¿ 0.5) found in these focused Linear Models.

6.3.1 Focused LM Findings

The tables provided in this subsection represent the outcome of the Linear Models that were performed using the relative difference responses calculated using Equation 6.3 and then the transformed responses are averaged together to give a percentage of how well each Short Retry Limit level did without considering its corresponding Long Retry Limit level, and then the same averaging method is performed for each Long Retry Limit level without regard for the Short Retry Limit level with which it is paired. This means that the data provided on Tables 6.2, 6.3, and 6.4 shows the average percentage (if you multiply the tables' numbers by 100) above or below the average for the Maximum Speed and Fading Type that the specified retry limit level performed at for the given response variable.

Table 6.2. The Relative Difference Analysis of Constant Bit Rate Packets Sent

CBR Data Packets Sent																		
Fading Type (γ_k)	No Fading						Rayleigh						Ricean					
Maximum Speed (δ_l)	5	10	15	20	25	30	5	10	15	20	25	30	5	10	15	20	25	30
SRL Levels																		
$\hat{\mu}$ (SRL = 6)	-0.016	-0.014	0.002	-0.013	-0.013	-0.010	-0.088	-0.060	-0.041	-0.072	-0.094	-0.056	-0.093	-0.067	-0.070	-0.109	-0.102	-0.089
$\hat{\mu}$ (SRL = 8)	0.001	0.009	0.003	0.002	0.001	0.009	-0.020	-0.005	-0.002	-0.015	-0.017	0.009	-0.025	-0.020	-0.013	0.031	-0.017	-0.016
$\hat{\mu}$ (SRL = 10)	0.000	0.004	0.005	0.016	0.005	0.009	0.024	0.013	0.010	0.012	0.011	0.028	0.024	0.013	0.014	0.019	0.021	0.025
$\hat{\mu}$ (SRL = 15)	0.002	-0.002	-0.006	0.002	0.009	0.005	0.031	0.021	0.024	0.038	0.057	-0.006	0.044	0.023	0.040	0.052	0.058	0.046
$\hat{\mu}$ (SRL = 20)	0.012	0.003	-0.004	-0.007	-0.001	-0.013	0.053	0.030	0.009	0.036	0.044	0.025	0.051	0.050	0.030	0.069	0.041	0.035
***p*-value**	0.000	0.000	0.000	0.004	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SE of Means	0.004	0.004	0.002	0.005	0.004	0.003	0.014	0.008	0.006	0.011	0.014	0.014	0.009	0.008	0.006	0.013	0.015	0.009
LRL Levels																		
$\hat{\mu}$ (LRL = 2)	0.000	-0.011	-0.005	-0.003	-0.011	-0.004	-0.074	-0.082	-0.050	-0.070	-0.077	-0.063	-0.032	-0.048	-0.024	-0.046	-0.057	-0.033
$\hat{\mu}$ (LRL = 4)	-0.002	0.001	0.002	0.002	0.000	0.001	0.000	0.007	0.003	0.008	0.022	0.023	0.010	0.017	0.010	0.004	0.017	0.016
$\hat{\mu}$ (LRL = 6)	0.001	0.005	0.002	0.001	0.006	0.001	0.037	0.039	0.019	0.034	0.029	0.032	0.011	0.017	0.006	0.021	0.020	0.011
$\hat{\mu}$ (LRL = 10)	0.001	0.005	0.002	0.001	0.006	0.002	0.037	0.027	0.028	0.027	0.026	0.009	0.011	0.014	0.008	0.021	0.190	0.006
***p*-value**	0.873	0.001	0.002	0.877	0.003	0.418	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SE of Means	0.003	0.003	0.001	0.005	0.004	0.003	0.012	0.007	0.005	0.010	0.013	0.013	0.008	0.007	0.005	0.011	0.013	0.008

Table 6.2 provides some very interesting observations when concerning the relative differences of the Constant Bit Rate Data Packets *Sent* response variable. It can be seen that for all levels of both the Maximum Speed and the Fading Type, the effects of the Short Retry Limit levels are significant regarding how many packets a node sends. On the other hand, the Long Retry Limit levels fail to provide a significant impact in three of the Maximum Speed levels (5, 20 and 30 m/s) under the Fading Type level of No Fading. With p-values of 0.873, 0.877 and 0.418, this would normally raise a red flag, but these values can be explained. The

No Fading scenario uses the Free Space Propagation Model, which, as previously discussed, represents a perfect world in which there are no factors to disrupt a signal except for the distance it travels. This being the case, the Long Retry Limit, which controls how many times a transmitting node can attempt to resend a data packet, would not be a factor because the chances of the data packet being lost are reduced dramatically when compared to the Rayleigh and Ricean Fading environments. Since the opportunity for a data packet to be lost drops, so does the opportunity to send the data packet again. For the Long Retry Limit to have a significant impact on the performance of the network, the scenario of losing and then sending the data packet again must occur.

Another trend to note in Table 6.2 lies predominantly within the Rayleigh and Ricean Fading environments. For the Short Retry Limit levels, it can be viewed that the higher levels such as *20* provide a greater number of packets sent at lower *Maximum Speeds* such as 5 and 10 m/s, but as the speed of the network begins to increase, the lower Short Retry Limit levels begin to provide a larger number of packets sent.

The same trend can also be applied to the Long Retry Limit levels. For both the Rayleigh and Ricean Fading environments, it can be seen that at a Maximum Speed level of 5 m/s, there is no difference between setting the Long Retry Limit to a value of either 6 or 10. This means that other factors must be considered to decide which one performs better, including the response variables of CBR Packets *Lost* and *Received*.

One common idea is that sending more data packets most likely leads to higher network performance measures such as throughput. The next two tables in this subsection will put that idea to the test.

The next table to analyze is Table 6.3. This table provides information on each Retry Limit level's performance concerning the amount of data packets lost.

In Table 6.3, whose response variable is the relative differences of the Constant Bit Rate Data Packets *Lost*, it is shown once again that the Short Retry Limit's effect is very significant throughout all of the Maximum Speed and Fading Type levels. Also, for a second time, the Long Retry Limit levels fail to have a significant impact for some of the Maximum Speed levels in the No Fading environment. This

Table 6.3. The Relative Difference Analysis of Constant Bit Rate Packets Lost

CBR Data Packets Lost																		
Fading Type (γ_k)	No Fading						Rayleigh						Ricean					
Maximum Speed (δ_l)	5	10	15	20	25	30	5	10	15	20	25	30	5	10	15	20	25	30
SRL Levels																		
$\hat{\mu}$ (SRL = 6)	-0.287	-0.317	-0.283	-0.442	-0.323	-0.317	-0.131	-0.167	-0.159	-0.279	-0.272	-0.093	0.0284	0.003	-0.047	-0.067	-0.094	-0.114
$\hat{\mu}$ (SRL = 8)	-0.322	-0.037	-0.249	-0.218	-0.269	-0.232	-0.087	0.027	-0.040	-0.130	-0.125	-0.096	-0.0539	-0.051	-0.044	0.130	-0.127	-0.072
$\hat{\mu}$ (SRL = 10)	-0.243	-0.015	-0.091	-0.211	-0.031	-0.162	-0.011	-0.018	-0.009	0.064	-0.027	-0.006	-0.0581	-0.077	-0.107	0.072	-0.090	-0.031
$\hat{\mu}$ (SRL = 15)	0.121	0.211	0.227	0.203	0.176	0.066	0.096	-0.001	0.208	0.130	0.157	0.011	0.0102	-0.021	0.083	0.065	0.075	0.084
$\hat{\mu}$ (SRL = 20)	0.731	0.157	0.397	0.669	0.448	0.645	0.134	0.160	0.000	0.216	0.268	0.184	0.0735	0.145	0.116	0.204	0.236	0.132
***p*-value**	0.000	0.000	0.000	0.000	0.000	0.000	0.087	0.000	0.001	0.000	0.000	0.000	0.034	0.000	0.000	0.000	0.000	0.000
SE of Means	0.114	0.069	0.086	0.055	0.046	0.057	0.079	0.047	0.058	0.051	0.036	0.035	0.034	0.029	0.029	0.027	0.026	0.025
LRL Levels																		
$\hat{\mu}$ (LRL = 2)	-0.070	-0.380	-0.136	-0.122	-0.169	-0.117	-0.405	-0.461	-0.391	-0.421	-0.390	-0.337	-0.158	-0.190	-0.130	-0.204	-0.196	-0.153
$\hat{\mu}$ (LRL = 4)	-0.108	0.029	0.052	0.006	-0.009	0.057	-0.081	0.074	-0.019	0.070	0.051	0.048	0.042	0.060	0.020	0.052	0.091	0.095
$\hat{\mu}$ (LRL = 6)	0.089	0.175	0.042	0.075	0.094	0.042	0.209	0.222	0.142	0.182	0.171	0.116	0.055	0.066	0.026	0.085	0.032	0.035
$\hat{\mu}$ (LRL = 10)	0.089	0.175	0.042	0.041	0.084	0.019	0.276	0.166	0.269	0.169	0.169	0.173	0.061	0.064	0.084	0.068	0.074	0.024
***p*-value**	0.373	0.000	0.245	0.032	0.000	0.064	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SE of Means	0.102	0.062	0.077	0.050	0.041	0.051	0.071	0.042	0.052	0.046	0.033	0.031	0.031	0.026	0.026	0.025	0.023	0.023

can be explained based on the same reasoning provided for the previous table's findings, which states that the No Fading scenario uses the Free Space Propagation Model, which, as previously discussed, represents a perfect world in which there are no factors to disrupt a signal except for the distance it travels. This being the case, the Long Retry Limit, which controls how many times a transmitting node can attempt to resend a data packet, would not be a factor because the chances of the data packet being lost are reduced dramatically when compared to the Rayleigh and Ricean Fading environments. Since the opportunity for a data packet to be lost drops, so does the opportunity for the Long Retry Limit to have a significant impact on the performance of the network.

This table provides information that suggests the lower Short Retry Limit levels provide fewer lost packets during all of the Rayleigh Fading environment runs, yet the lowest SRL level actually produces more lost packets than its two higher counterparts at lower speeds under the Ricean Fading conditions. As the Maximum Speed levels increase in the Ricean Fading environment, the Short Retry Limit levels of *8* and *10* switch off on providing the fewest data packets lost until at a level of 30 m/s, the SRL level of *6* finally regains supremacy.

Table 6.3 also provides observations that show under the Rayleigh and Ricean Fading conditions, the lowest Long Retry Limit level consistently provides the fewest number of lost data packets. This can be reasoned through referencing back to Table 6.2 in which the lower Long Retry Limit levels also produced the fewest number of packets sent. Due to that factor, it appears rational that a node cannot lose packets it is not sending.

Table 6.4. The Relative Difference Analysis of Constant Bit Rate Packets Received

CBR Data Packets Received																		
Fading Type (γ_k)	No Fading						Rayleigh						Ricean					
Maximum Speed (δ_l)	5	10	15	20	25	30	5	10	15	20	25	30	5	10	15	20	25	30
SRL Levels																		
$\hat{\mu}$ (SRL = 6)	-0.011	-0.007	0.006	0.008	0.005	0.002	-0.080	-0.047	-0.030	-0.046	-0.069	-0.055	-0.124	-0.080	-0.073	-0.121	-0.104	-0.088
$\hat{\mu}$ (SRL = 8)	0.005	0.011	0.007	0.012	0.016	0.015	-0.015	-0.004	0.002	0.003	0.003	0.018	-0.024	-0.014	-0.008	-0.011	0.006	-0.006
$\hat{\mu}$ (SRL = 10)	0.005	0.004	0.006	0.025	0.007	0.016	0.024	0.014	0.014	0.008	0.019	0.031	0.037	0.027	0.031	0.043	0.049	0.040
$\hat{\mu}$ (SRL = 15)	0.003	-0.007	-0.010	-0.010	-0.002	0.003	0.025	0.024	0.007	0.024	0.039	0.012	0.057	0.032	0.033	0.050	0.056	0.038
$\hat{\mu}$ (SRL = 20)	-0.002	-0.001	-0.009	-0.035	-0.026	-0.036	0.045	0.013	0.008	0.013	0.008	-0.006	0.053	0.034	0.017	0.039	-0.007	0.016
***p*-value**	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SE of Means	0.003	0.003	0.002	0.006	0.004	0.003	0.012	0.006	0.005	0.010	0.013	0.008	0.012	0.008	0.007	0.013	0.016	0.011
LRL Levels																		
$\hat{\mu}$ (LRL = 2)	0.001	-0.001	-0.002	0.002	-0.001	0.002	-0.038	-0.031	-0.021	-0.027	-0.028	-0.023	-0.015	-0.027	-0.010	-0.009	-0.024	-0.007
$\hat{\mu}$ (LRL = 4)	0.000	0.001	0.001	0.002	0.000	-0.001	0.007	0.010	0.004	0.003	0.020	0.015	0.005	0.010	0.009	-0.008	-0.001	-0.001
$\hat{\mu}$ (LRL = 6)	-0.001	0.000	0.001	-0.003	0.000	-0.001	0.020	0.014	0.008	0.015	0.007	0.012	0.007	0.010	0.003	0.006	0.019	0.006
$\hat{\mu}$ (LRL = 10)	-0.001	0.000	0.001	-0.001	0.000	0.001	0.011	0.007	0.009	0.009	0.001	-0.004	0.004	0.007	-0.002	0.011	0.007	0.002
***p*-value**	0.939	0.971	0.501	0.872	0.995	0.090	0.001	0.000	0.000	0.050	0.042	0.001	0.441	0.001	0.176	0.558	0.183	0.826
SE of Means	0.003	0.003	0.002	0.005	0.003	0.003	0.011	0.005	0.004	0.009	0.012	0.007	0.010	0.008	0.006	0.012	0.014	0.009

For Table 6.4, which deals with the relative difference of the Constant Bit Rate Packets *Received* response variable, it becomes very interesting quickly. Just to begin, the Long Retry Limit levels are only significant in one Maximum Speed level (30 m/s) under the No Fading environment and this limit also provides some very interesting results in the cases where it is significant. The Short Retry Limit levels are significant for all Maximum Speed levels in each Fading Type level.

The Short Retry Limit levels' observations show that the highest level of *20* provides the greatest number of received packets at the lowest Maximum Speed of 5 m/s under Rayleigh Fading conditions, and then begins to decline in performance as the Maximum Speed levels increase. The opposite can be said for the lower Short Retry Limit levels. Although some of their performances do drop somewhat, their CBR data packets received measures under the Rayleigh Fading environment still begin to rise above the higher levels until the Maximum Speed level reaches 20 m/s. This spike in the amount of received packets can most likely be explained by the increase of network speeds without the increase of the size of the environment. One can speculate that as the nodes began to increase their speed to higher levels, they began to more easily travel across the environment, possibly resulting in more direct contacts between the transmitting nodes and the data packets' final destinations. For the SRL levels under the Ricean Fading environment, the second highest level provided the most received data packets at the lowest Maximum Speed level of 5 m/s. Once again, as the speed began to increase, the higher Short Retry Limit levels began to decline in the throughput measures while the lower limits began to improve. Also, the same spike in received data packets that occurs in received packets around the level of 20 m/s can be reasoned the same way as the Rayleigh Fading data packet spike.

The Long Retry Limit has the most significant impact when considered under the Rayleigh Fading environment. The above table shows that at the lowest level of 5 m/s, the second highest Long Retry Limit level of *6* provides the greatest throughput. As the speed increases, the levels of *4* and *6* usually have only a small difference of 0.4% data packets received than the average. Under the Ricean Fading environment, the Long Retry Limit levels become fairly insignificant in their impact on the network performance. Once again, this can be explained through understanding that the environment size stayed the same, and once the speed of the network increased, the transmitting nodes were able to have more opportunities of making direct connections with the final destination nodes. Also, as previously explained, Ricean Fading is a "line-of-sight" fading scenario in which a dominant path or signal is present. This being said, and taking into consideration the increased probability of transmitter/received pairs making a direct connection, a case can be made that it is not unreasonable for the Long Retry Limit levels to not have as significant of a role in the Ricean Fading conditions as they would be in the Rayleigh Fading conditions.

6.4 Dynamic Level Selections

The next step in this research involves evaluating the charts presented in the previous section and making decisions on which Short and Long Retry Limit levels will work best for each range of speed. Part of making these decisions involves making trade-offs between difference response variables. These trade-offs are based in the need to keep *Quality of Service*(QoS) in mind while, at the same time, proceed towards creating a dynamic Retry Limit mechanism that increases the more valued measure of data packets received. Please note that these dynamic Retry Limit levels will not be fading specific, as it is not yet feasible to know the kind of fading you are experiencing in a real world setting. Instead, the Retry Limit levels will first be broken down into two fading types, which are Rayleigh and Ricean Fading. The No Fading or Free Space Propagation Model will no longer be included in this work as it has already been shown in (Guardiola & Matis, 2007) to not provide high fidelity simulations that properly emulate the randomness at which a signal can travel in the real world.

6.4.1 Dynamic Levels By Speed and Fading

The first part of selecting the dynamic levels is to record which limits did best under which Maximum Speed levels and for the performance measures of CBR Data Packets *Lost* and *Received*. The top Retry Limit performers in the Rayleigh Fading environment are shown below in Table 6.5 along with the average speed that the Maximum Speed level represents. Understanding the average speed aspect is important in assigning these Retry Limits to specific ranges of speed because a range may involve more than one Short or Long Retry Limit. Once these Retry Limit levels have been identified as potential suitors, the pros and cons must then be properly evaluated.

Table 6.5 shows that the lowest Short and Long Retry Limit levels provided the fewest amount of data packets lost under the Rayleigh Fading conditions. This tends to follow the previously mentioned hypothesis that the fewer a node sends, the fewer the node will lose, and a node cannot lose what it does not send. This is important to understand, because along with that line of thinking comes the fact that another node cannot receive a data packet that is never sent.

Table 6.5 also shows that the highest Short Retry Limit level, *20*, provided the most data packets received at the lowest Maximum Speed level of 5 m/s. As the Maximum Speed level is increased, it is important to note that the Short Retry Limit levels begin to decrease. It is also important to notice the Retry Limit boxes for the "Data Packets Received" portion in which more than one Retry Limit appears. This occurs when the difference between two limits is minimal and it is possible to choose a limit that has a minutely smaller amount of data packets

Table 6.5. The Best Retry Limits Under Rayleigh Fading

Highest Performing Retry Limits in Rayleigh Fading						
Maximum Speed	5 m/s	10 m/s	15 m/s	20 m/s	25 m/s	30 m/s
Average Speed	2.5 m/s	5 m/s	7.5 m/s	10 m/s	12.5 m/s	15 m/s
Data Packets Lost						
Short Retry Limit Level(s)	6	6	6	6	6	8
Long Retry Limit Level(s)	2	2	2	2	2	2
Data Packets Received						
Short Retry Limit Level(s)	20	15	10 , 8	10 , 8	8	10 , 8 , 6
Long Retry Limit Level(s)	6	4	4 , 2	4 , 2	4 , 2	4 , 2

received in order to receive the benefits of that limit's lower amount of lost data packets. This is a QoS decision that must be made while still keeping in mind the overall goal of improvement in the data packets received response variable.

The Retry Limits in Table 6.6 were chosen based on the same methodology as Table 6.5. It is interesting to note that the Short Retry Limit level of *10* provided the fewest lost data packets until the Maximum Speed level reached 15 m/s. The Long Retry Limit's lowest level, *2*, followed the trend of providing the fewest number of lost data packets.

Another attention-grabbing observation from Table 6.6 comes from the Short Retry Limit levels under the "Data Packets Received" portion of the table. Comparing these levels with the levels found for the Rayleigh Fading scenario, it can be seen that the Ricean Fading conditions prefer a little bit higher levels. Also, the Long Retry Limit levels in the "Data Packets Received" area of the table seem to stay somewhat stable as the speed of the network is increased, and then at the Maximum Speed level of 20 m/s, the LRL level jumps. Although this is peculiar, it is not of too much concern because the p-value of the Long Retry Limit levels for that scenario and response variable was 0.558. Such comparisons as these must be

Table 6.6. The Best Retry Limits Under Ricean Fading

Highest Performing Retry Limits in Ricean Fading						
Maximum Speed	5 m/s	10 m/s	15 m/s	20 m/s	25 m/s	30 m/s
Average Speed	2.5 m/s	5 m/s	7.5 m/s	10 m/s	12.5 m/s	15 m/s
Data Packets Lost						
Short Retry Limit Level(s)	10	10	6	6	8	6
Long Retry Limit Level(s)	2	2	2	2	2	2
Data Packets Received						
Short Retry Limit Level(s)	15	20 , 10	15 , 10	15 , 10	15 , 10	10 , 8
Long Retry Limit Level(s)	6	6	4	10	6 , 4	6

identified and noted in order to competently provide dynamic Retry Limit levels which satisfy both fading cases.

6.4.2 Dynamic Levels Based On Speed

Now that the dynamic Retry Limit levels have been identified within each fading scenario, they must now be assessed for possible trade-offs and combined to create a dynamic mechanism that can perform well in both fading environments. The assessments will be made using information from Tables 6.3 and 6.4. It is important to note that each movement scenario had an average speed. These average speeds will be used to break down possibilities for placing the correct Retry Limits with the right *Speed Ranges*. Once the speed ranges are set, the Maximum Speed levels which will have an influence on that speed range's Retry Limit levels will be grouped in *Decision Ranges*. A Maximum Speed level is determined to have an influence on that speed range if its average speed is within the speed range. Also, even though the amount of lost data packets will be considered, the amount of received data packets will carry more influence for the following two reasons:

1. The amount of lost data packets are in the magnitude of 100's while the amount of received data packets are in the magnitude of 1000's. This could mean, for example, although the data packets lost were decreased by 18%, the data packets received went down by 5%. In numerical terms and considering their magnitudes, this means that the network may have lost 18 or 20 fewer packets, but in the process, the network also failed to get 50 or 70 data packets from the transmitting node to the receiving node.

2. Increasing the amount of received packets is this research's main objective

and with those two rationales in mind, it is now time to begin selecting the Retry Limit levels for the dynamic settings.

The first speed range to consider is from *0.0 - 5.0 m/s*. With this being the speed range, the decision range becomes the Retry Limits of Maximum Speed levels of 5 and 10 m/s, with the level of 5 m/s being more heavily weighted because its average sits right in the middle of the speed range, and to use the Retry Limit at 10 m/s as more of a tie breaking method. First, the values from Table 6.4 are retrieved for the highest Short Retry Limit performances for the 5 m/s Maximum Speed level under both the Rayleigh and Ricean Fading environments. For the Rayleigh Fading environment with a Short Retry Limit of *20*, there are 4.53% more received data packets on average. For the Ricean Fading environment with a Short Retry Limit of *15*, there are 5.7% more received data packets on average.

The next step is to evaluate the alternatives. If the Ricean Fading's Short Retry Limit of *15* is chosen, then the network will lose 2.02% of its throughput when dealing with Rayleigh Fading. On the other hand, if the Rayleigh Fading's Short Retry Limit of *20* is chosen, then the network will only lose 0.36% of its throughput when dealing with Ricean Fading conditions. It should also be noted that one of the Ricean Fading Short Retry Limit levels chosen for its 10 m/s Maximum Speed level (which is in the decision range as well) is a level of *20*. This shows that the Ricean Fading scenario can comfortably have a SRL level of 15 or 20 and still perform at about the same level. Keeping in mind that the goal is to increase the amount of data packets received, it is decided that the range *0.0 - 5.0 m/s* will have the Short Retry Limit level of *20*.

The Long Retry Limit level for this speed range is chosen to be 6. This is chosen because 6 provides the best throughput performance in both fading scenarios.

For the second speed range from *5.1 - 10.0 m/s*, Rayleigh prefers the SRL of 15 while Ricean prefers either 20 or 10. 15 is chosen as the SRL for this speed range because it not only provides the Rayleigh environment with the best throughput, but it also does a good job of compromising for Ricean, in which it gets the fewer lost packets that 10 provides and allows for the higher throughput that 20 provides.

The LRL for this speed range is chosen to be 4 because it provides better throughput than 2. Ricean prefers 4, 6 or 10, but since the LRL is insignificant for the Ricean environment, these values are somewhat pushed aside.

For the third speed range of *10.1 - 15.0 m/s*, a Short Retry Limit level of 10 is selected because it provides better throughput than the choice of 8 with up to 4.37% more throughput. It also provides 9.233% fewer lost packets than an SRL of 15.

The Long Retry Limit for this range is chosen to be 2. Although an LRL level of 2 provides 2.86% less throughput, it is overshadowed by the fact that it also provides 33.85% fewer lost packets.

The final speed range is $\geq$ *15.1 m/s*. For this speed range, it is important to note that these speeds can only be reached through the use of an automobile. So, considering this fact, the main goal in choosing these limits must shift to coping with the amount of possible lost packets. With this in mind, the Short Retry Limit level chosen for this speed range is 6 because it provides the fewest lost packets overall when considering both fading environments.

The Long Retry Limit for the final speed range is chosen to be 2. This selection is based on the same principle of dealing with the Quality of Service measure instead of attempting to increase throughput performance.

Tables 6.4 and 6.3 were used in calculating the amount of network performance improvement for each Retry Limit level, and were then chosen accordingly. Another item to note in Table 6.4 is that for all but one of the Maximum Speed levels in the Ricean Fading environment, the Long Retry Limit appears to have an extremely small impact on the amount of data packets received. This being considered, the amount of lost packets will receive more merit in determining the desirable Ricean Fading Long Retry Limit levels, but more importantly, the response variables under

Rayleigh Fading will take more precedence over deciding what that Long Retry Limit level will be set at since these levels significantly influence the amount of data packets received in the Rayleigh Fading environment. Considering this along with the fact that a Long Retry Limit level of 6 provides the most received data packets for the Maximum Speed level of 5 m/s in the Rayleigh Fading scenario, the chosen Long Retry Limit level for the speed range of *0.0 - 5.0* will be set at 6.

As the speed of the network begins to reach speeds of upwards to 30 m/s, which converts into roughly 67 mph, the focus for deciding the Retry Limits is shifted into more of a goal of managing the situation. This means that a feasible middle ground must be chosen that helps combat lost packets while still maintaining a reputable amount of throughput. Speeds of this nature are that of an automobile on a highway. This being the case, as the size of an environment expands, the nodes that are traveling at this high rate of speed will not be dependable enough to be consistently counted on to be a route for data transmissions. The choice to simply manage the situation at these speeds is an effort to compensate and plan ahead for later work in which these environment sizes may be expanded to accommodate high speed nodes in the ad-hoc network. What this all means is that for the higher speed dynamic settings, especially for nodes moving faster than 15 m/s, the Short and Long Retry Limit levels will be chosen in a more conservative, less greedy fashion in which the amount of data packets lost becomes more highly regarded than the amount of data packets received.

Table 6.7. Retry Limits Chosen for Each Speed Range

	Dynamic Retry Limit Settings			
	0.0 - 5.0 m/s	5.01 - 10.0 m/s	10.01 - 15.0 m/s	≥ 15.01 m/s
Decision Range	5 - 10 m/s Max	10 - 15 - 20 m/s Max	20 - 25 - 30 m/s Max	30 m/s Max
Short Retry Limit	20	15	10	6
Long Retry Limit	6	4	2	2

Table 6.7 shows the final Dynamic Short and Long Retry Limit settings. It also shows the speed range for which the chosen Retry Limits will be applied as well as the decision range that the Short and Long Retry Limit selections were analyzed and chosen from.

6.5 The NS-2 Dynamic Retry Limits Code

The *C++* code written to implement the Dynamic Short and Long Retry Limits can be seen in Appendix A. This code was written in the file $mac - 802_1 1.cc$ which is located in the director $/ns2.31/mac/mac - 802_1 1.cc$. This code was validated by making the NS-2.31 simulator print out the current transmitting node's speed and its current Short and Long Retry Limits. These Retry Limits were then checked against the transmitting node's speed to verify that they had been adjusted correctly. The transmitting node's speed is attached to each packet that is processed by the MAC, so the SRL and LRL are adjusted on a per-packet basis to specifically suit the transmitter's current mobile state.

6.6 Dynamic vs. Default Settings Comparison

After implementing the new dynamic mechanism into the network simulator NS-2.31 as described in the previous section, the task now became pulling out the data provided by the simulations and then analyze it in its raw form to understand how much better or worse the dynamic mechanism did compared to the default settings. The following subsections represent how the data was analyzed as well as the findings that were a result of the simulations.

6.6.1 Paired t-tests

In order to truly understand if there is a significant difference between the results gained by running the Dynamic Short and Long Retry Limits mechanism and the static, default Short and Long Retry Limits, a paired t-test will be performed. The *Null and Alternative Hypotheses* are as follow:

$$H_0 : \mu_1 - \mu_2 = 0 \tag{6.5}$$

$$H_a : \mu_1 - \mu_2 \neq 0 \tag{6.6}$$

The three responses of CBR Data Packets *Sent*, *Lost*, and *Received* were analyzed

Table 6.8. Paired t-Test p-Values

	Paired t-Test Results											
	Rayleigh						Ricean					
Maximum Speed (m/s)	5	10	15	20	25	30	5	10	15	20	25	30
Packets Sent p-values	0.014	0.006	0.004	0.014	0.086	0.791	0.008	0.038	0.000	0.000	0.131	0.075
Packets Lost p-values	0.142	0.009	0.010	0.158	0.004	0.028	0.962	0.007	0.033	0.000	0.549	0.184
Packets Received p-values	0.018	0.018	0.056	0.027	0.449	0.247	0.030	0.129	0.000	0.000	0.075	0.125

Represents Statistical Significance

using a paired t-test for each set of data within each combination of the speed and fading type levels. The p-values from the paired t-tests can be seen in Table 6.8.

This table shows that many of the p-values are significant. This means that, for the significant p-values, the null hypothesis of the paired t-test is rejected. When the null hypothesis of a paired t-test is rejected, it means there is a significant difference between the two pairs of data. This is beneficial to know before continuing on with the analysis because if most of the p-values were insignificant, it would mean that the Dynamic Short and Long Retry Limit did not make any different of an impact on the performance of the system compared to the current default settings. Now that the differences have been verified, the next step is to see in what ways the Dynamic mechanism possibly improved or decreased the mobile ad-hoc network's performance.

6.6.2 Overall Network Performance Comparison

Before beginning the analysis of the Dynamic mechanism versus the current default settings, it must be noted that for accuracy purposes, the observations for both the default and Dynamic settings under Ricean Fading with Maximum Speed levels of 15 and 20 m/s had to be eliminated from the comparison. This was done due to problems with the NS-2.31 simulator, in which the simulator was unable to properly run the full simulations for these settings. Partial analysis of these observations would have been possible, but the accuracy of their findings would have been extremely questionable.

The first response variable that will be looked into is the *Percentage of Packets Lost*. This response variable is calculated using the following equation:

$$P_L = \frac{\sum_{m=1}^{10} X_{klm}}{\sum_{m=1}^{10} Z_{klm}} \tag{6.7}$$

where X_{klm} represents the amount of lost packets for a given route, m, which is encompassed by k which signifies the Fading Type, and l, which stands for the Maximum Speed level. The Z_{klm} in the above equation is the amount of packets sent for that route. Equation 6.7 produces the specified value of the averages with the range of $1 \geq P_L \geq 0$ with the response consisting of the units (The No. of Packets Lost)/(Packet Sent).

Table 6.9. Comparison of the Percentage of Lost Packets Per Packet Sent

Percentage of Packets Lost										
Fading Type	Rayleigh						Ricean			
Maximum Speed	5 m/s	10 m/s	15 m/s	20 m/s	25 m/s	30 m/s	5 m/s	10 m/s	25 m/s	30 m/s
Retry Limit Settings										
Default (74) Settings	10.28%	9.87%	5.45%	9.24%	10.41%	14.76%	17.71%	13.89%	21.29%	18.19%
Dynamic Settings	12.52%	14.11%	8.53%	10.10%	14.69%	12.97%	15.73%	15.40%	19.94%	18.83%
Differences	-2.24%	-4.23%	-3.08%	-0.87%	-4.29%	1.79%	1.97%	-1.51%	1.36%	-0.64%
Best Performance	Default	Default	Default	Default	Default	Dynamic	Dynamic	Default	Dynamic	Default

Table 6.9 shows the average amount of lost packets for every one packet sent. By multiplying the values from Equation 6.7 by 100, the average percentage of packets lost that were sent can be obtained, as shown in the table. Once again, this average comes from applying Equation 6.7 to each route within a given combination of Maximum Speed, Fading Type and Retry Limit settings. These averages are then averaged together to obtain an overall view of the network's performance regarding the stated response for that particular set-up. The differences presented in the table are absolute values.

To give a broader view, the Rayleigh Fading environment favored the Default Retry Limit settings, which provided an average of 2.15% fewer packets lost when compared to the Dynamic settings. Under the Ricean Fading conditions, the Dynamic settings allowed for an average of 0.29% fewer packets lost when looking at the general picture. The fact that the Dynamic settings allowed for a smaller percentage of packets lost in this setting is somewhat of a bonus because the main goal of the Dynamic mechanism was to increase throughput first and foremost, and improve on the QoS indirectly if possible.

The second measure of performance is the *Percentage of Total Throughput*. This measure calculates both Retry Limit Settings' abilities to reach perfect throughput performance. The perfect throughput performance value can be calculated as follows:

$$T_P = SimulationTime \times ConstantBitRate \tag{6.8}$$

In the case of this paper, all simulations times were *350 seconds*. The Constant Bit Rate was set to *5 packets per second*. Using Equation 6.8, the most received packets possible between a transmitting and receiving node is *1750 packets*. Being able to calculate this value is the precise reason why Constant Bit Rate data packets were used in the simulations.

Now that the perfect throughput value has been calculated, the average number of received packets must be computed. These averages are done simply by going within each Maximum Speed, Fading Type and Retry Limit Setting combinations and averaging the amount of received packets among the 10 routes. The equation looks like the following:

$$R_A = \frac{(\sum_{m=1}^{10} R_{klm})}{10} \tag{6.9}$$

where R_A is the *Average Number of Received Packets* within the given factor levels, R_{klm} is the *Number of Received Packets* for the given route, m is the *Route Number*, k is the Fading Type level, l is the Maximum Speed level, and *10* is the total number of routes for the set-up.

Once the average number of received packets has been calculated for each instance, they are then divided by the perfect throughput value of 1750 packets. The resulting decimal, when multiplied by 100, shows the average percentage of data packets that were able get from the transmitting node to the receiving node.

In Table 6.10, it can be seen that the Dynamic settings provide better network throughput in all Maximum Speed and Fading Type levels. The differences presented in the table are absolute differences.

A real bright spot that comes from this is that in the Ricean Fading environment,

Table 6.10. Comparison of the Percentage of Total Throughput

Percentage of Total Throughput										
Fading Type	Rayleigh						Ricean			
Maximum Speed	5 m/s	10 m/s	15 m/s	20 m/s	25 m/s	30 m/s	5 m/s	10 m/s	25 m/s	30 m/s
Retry Limit Settings										
Default (74) Settings	69.41%	72.22%	74.23%	68.27%	67.27%	70.81%	62.69%	68.27%	59.19%	60.93%
Dynamic Settings	77.30%	76.73%	76.93%	72.70%	69.35%	72.64%	71.37%	71.89%	64.82%	64.68%
Differences	7.89%	4.51%	2.70%	4.43%	2.09%	1.83%	8.68%	3.62%	5.63%	3.75%
Best Performance	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic

the Dynamic settings were able to both create a lower percentage of lost packets and increase total network throughput. For the Ricean Fading level, the Dynamic settings provided a total average of 5.418% more data packet throughput than the Default setting.

For the Rayleigh Fading environment, the Dynamic settings allowed for a 3.91% throughput advantage over the Default setting. This means that although the Default setting did produce 2.1% fewer lost packets per packet sent, the Dynamic settings were able to overshadow it with the 3.91% increased throughput. The term overshadowed is used because, as discussed previously, the magnitude of the throughput percentage is in the thousands when considering the raw data, while the magnitude of the percentage of lost packets remains in the low hundreds.

CHAPTER 7
CONCLUSIONS AND DISCUSSION

A Wireless mobile ad-hoc network is a complicated system. The idea of an autonomous, infrastructure-free, self-sustaining wireless network is very attractive to not only the military and disaster relief efforts, but also to those who use wireless devices every day. The road to reaching a fully functional and dependable mobile ad-hoc network will involve not only innovative thinking to expand on current ideas, but also inventive approaches that attack the shortcomings of a MANET from an entirely new angle. The work presented in this research is an innovation that may hopefully later lead to inventive techniques.

In this thesis, it was shown that creating static parameters such as the Short and Long Retry Limits in the Request To Send/Clear To Send mechanism can dampen, or hold back, the potential of an otherwise good idea. Using Linear Models, Paired t-Tests and upfront representations of the raw data collected through simulations, the benefits of a wireless device that can adjust its own settings based on the personal knowledge of its mobile state were demonstrated. It was shown that it is possible to identify factors that impact the network performance through the use of Designed Experimentation, and once the the scope is narrowed, the effective levels of those factors can be obtained to create a better option than the one that is currently available. The better option, the Dynamic Short and Long Retry Limits, was then proven through simulations in which both the Default and Dynamic settings were given the same simulation scenarios and then analyzed for significant differences. Through the final comparison analysis, it was shown that a wireless device that can adjust its own parameters based on its state of mobility consistently shows improved network throughput over the static (Default) settings, and in one fading scenario, also showed better general MANET performance when concerning the percentage of lost packets.

CHAPTER 8
FUTURE RESEARCH

The research presented in this thesis leaves many paths to be explored. For starters, it would be very worthwhile to test the Dynamic Retry Limits mechanism under the Nakagami Fading model. This would extend the exploration that fading effects can have on a mechanism such as this.

Another avenue that should be taken involves tying in the MAC-layer backoff timer as well as the Request To Send Threshold into the dynamic mechanism so that both may adjust themselves based on the transmitting node's mobility state like the Dynamic Retry Limits mechanism proposed in this work.

In order to explore scalability possibilities, the Dynamic Retry Limits mechanism should be applied to other protocols such as AODVjr which offer more flexibility than AODV when it comes to the density of the mobile ad-hoc network.

It would also be advantageous to apply this mechanism to TCP/IP traffic in order to step away from the worst-case scenario of Constant Bit Rate packets and instead step into the realistic realm of the extremely common TCP/IP traffic.

The last and most ambitious direction for this research to be taken is to apply the Dynamic Retry Limits mechanism in a real world MANET. One of the truly challenging points of this research would be trying to use Global Positioning System (GPS) coordinate approximations to attempt to estimate a mobile node's speed and adjust the Dynamic Retry Limits mechanism accordingly.

REFERENCES

Aad, I., & Castelluccia, C. (2001). Differentiation mechanisms for ieee 802.11. In *Proceedings of ieee infocom.*

Bhat, M., Chokshi, R., Desai, S., Panichpapiboon, S., Wisitpongphan, N., & Tonguz, O. (2003). Impact of mobility on the performance of ad hoc wireless networks. *IEEE.*

Bruno, R., Conti, M., & Gregori, E. (2002). Optimal performances: Rts/cts mechanism vs. basic access. In *Proceedings of pimrc.*

Crow, B., Widjaja, I., Kim, J., & Sakai, P. (1997). *Ieee 802.11 wireless local area networks* (Technical Report). The MITRE Corporation, Fujitsu Network Communications, University of Arizona and Cypress Semiconductor.

Devore, J. (2004). *Probability and Statistics for Engineering and the Sciences.*

Fall, K., & Varadhan, K. (2002). *The ns-2 manual* (Technical Report, The VINT Project). UC Berkeley.

Guardiola, I., & Matis, T. (2007). On the supplementary control overhead in mobile ad-hoc network protocols. In *The 2007 networking and electronic commerce research conference.* Riva Del Garda, Italy.

Gunes, M., Hecker, M., & Bouazizi, I. (2003). Influence of adaptive rts/cts retransmissions on tcp in wireless and ad-hoc networks. In *Proceedings of the eighth ieee international symposium on computers and communication.*

Howald, R. (Accessed May 27, 2008). *The fuss about fading.* (Communication Systems Design. http://www.commsdesign.com/main/art9804.htm)

Kim, S., Lee, J., & Yeom, I. (2008). Neighbor-aware adaptive retry limit for ieee 802.11-based mobile ad hoc networks. In *Proceedings of the ieee wireless communications and network conference.* Las Vegas, Nevada.

Linnartz, J. (Accessed May 27, 2008). *Wireless communication reference website. http://wireless.per.nl/reference/chaptr03/rayleigh.htm.*

Mckay, D., Mahmoud, N., Olsen, C. M., Rezvani, B., Kermani, P., & Hortensius, P. D. (n.d.).

Myers, R., Montgomery, D., & Vining, G. (2002). *Generalized Linear Models With Applications in Engineering and the Sciences.*

NS-2.31, T. N. S. (Accessed May 25, 2008). (Tech. Rep.). DARPA and The Vint Project.

Perkins, C., & Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings of the second ieee workshop on mobile computer systems and applications.* Washington, D.C.

Punnoose, R., Nikitin, P., & Stancil, D. (2000). Efficient simulation of ricean fading within a packet simulator. In *Proceedings of ieee vehicular technology conference* (p. 764-767).

Random waypoint model. (Accessed June 3, 2008). http://keskus.hut.fi/ esa/java/rwp/rwp-model.shtml.

Ware, V. O. C. (Accessed May 28, 2008). *Vietnam open course ware on wireless networks.*

Weinmiller, J., Woesner, H., Ebert, J., & Wolisz, A. (1995). Analyzing the rts/cts mechanism in the dfwmac media access protocol for wireless lans. In *Ifip tc6.*

APPENDIX A

MAC 802.11 NS-2 SOURCE CODE MODIFICATION

```
/* ======================================================================
   Retransmission Routines
   ====================================================================== */
void
Mac802_11::RetransmitRTS()
{


assert(pktTx_);
assert(pktRTS_);
assert(mhBackoff_.busy() == 0);
macmib_.RTSFailureCount++;


ssrc_ += 1; // STA Short Retry Count


//// ***Dynamic Short Retry Limit Coded By Aaron Phillips***


double SRL;
Packet *p = Packet::alloc();
hdr_cmn* ch = HDR_CMN(p);
struct rts_frame *rf = (struct rts_frame*)p->access(hdr_mac::offset_);
double sndspd_ = p->txinfo_.getNode()->speed();
cout<<"Speed = "<<sndspd_<<endl;
if(sndspd_ <= 5.00) {SRL = 20;}
else if (sndspd_ <= 10.00) {SRL = 15;}
else if (sndspd_ <= 15.00) {SRL = 10;}
else {SRL = 6;}
cout<<"SRL = "<<SRL<<endl;

if(ssrc_ >= SRL) {
```

```
discard(pktRTS_, DROP_MAC_RETRY_COUNT_EXCEEDED); pktRTS_ = 0;
/* tell the callback the send operation failed
   before discarding the packet */
hdr_cmn *ch = HDR_CMN(pktTx_);
//cout<<"RTS SRL BEING USED!"<<endl;
if (ch->xmit_failure_) {
                        /*
                         *  Need to remove the MAC header so that
                         *  re-cycled packets don't keep getting
                         *  bigger.
                         */
ch->size() -= phymib_.getHdrLen11();
                        ch->xmit_reason_ = XMIT_REASON_RTS;
                        ch->xmit_failure_(pktTx_->copy(),
                                          ch->xmit_failure_data_);
                }
discard(pktTx_, DROP_MAC_RETRY_COUNT_EXCEEDED);
pktTx_ = 0;
ssrc_ = 0;
rst_cw();
} else {
struct rts_frame *rf;
rf = (struct rts_frame*)pktRTS_->access(hdr_mac::offset_);
rf->rf_fc.fc_retry = 1;
//cout<<"RTS SRL BEING USED!"<<endl;
inc_cw();
mhBackoff_.start(cw_, is_idle());
}
}

void
Mac802_11::RetransmitDATA()
{
struct hdr_cmn *ch;
struct hdr_mac802_11 *mh;
u_int32_t *rcount, thresh;
assert(mhBackoff_.busy() == 0);

assert(pktTx_);
assert(pktRTS_ == 0);
```

```
ch = HDR_CMN(pktTx_);
mh = HDR_MAC802_11(pktTx_);

/*
 *  Broadcast packets don't get ACKed and therefore
 *  are never retransmitted.
 */
if((u_int32_t)ETHER_ADDR(mh->dh_ra) == MAC_BROADCAST) {
Packet::free(pktTx_);
pktTx_ = 0;

/*
 * Backoff at end of TX.
 */
rst_cw();
mhBackoff_.start(cw_, is_idle());

return;
}

macmib_.ACKFailureCount++;



//// ***Dynamic Short Retry Limit Coded By Aaron Phillips***

double LRL;
double SRL2;
Packet *p = Packet::alloc();
struct ack_frame *af = (struct ack_frame*)p->access(hdr_mac::offset_);
double sndspd_ = p->txinfo_.getNode()->speed();
cout<<"Speed2 = "<<sndspd_<<endl;
if(sndspd_ <= 5.00) {SRL2 = 10;}
else if (sndspd_ <= 10.00) {SRL2 = 8;}
else {SRL2 = 6;}
if(sndspd_ <= 5.00) {LRL = 6;}
else if (sndspd_ <= 10.00) {LRL = 4;}
else {LRL = 2;}
cout<<"LRL = "<<LRL<<endl;
```

```
cout<<"SRL2 = "<<SRL2<<endl;


if((u_int32_t) ch->size() <= macmib_.getRTSThreshold()) {
                rcount = &ssrc_;
               thresh = SRL2 ;
       //cout<<"ACK USED SRL!"<<endl;
        } else {
 //cout<<"LONG RETRY LIMIT BEING USED!"<<endl;
                 rcount = &slrc_;
                thresh = LRL;
        }

(*rcount)++;

if(*rcount >= thresh) {
/* IEEE Spec section 9.2.3.5 says this should be greater than
   or equal */
macmib_.FailedCount++;
/* tell the callback the send operation failed
   before discarding the packet */
hdr_cmn *ch = HDR_CMN(pktTx_);
if (ch->xmit_failure_) {
                        ch->size() -= phymib_.getHdrLen11();
ch->xmit_reason_ = XMIT_REASON_ACK;
                        ch->xmit_failure_(pktTx_->copy(),
                                          ch->xmit_failure_data_);
                }

discard(pktTx_, DROP_MAC_RETRY_COUNT_EXCEEDED);
pktTx_ = 0;
*rcount = 0;
rst_cw();
}
else {
struct hdr_mac802_11 *dh;
dh = HDR_MAC802_11(pktTx_);
dh->dh_fc.fc_retry = 1;

sendRTS(ETHER_ADDR(mh->dh_ra));
```

```
inc_cw();
mhBackoff_.start(cw_, is_idle());
}
}
```

APPENDIX B

FULL EXAMPLE OF OTCL SIMULATION SCRIPT

```
set val(proplog)        ricean-proplog.tr   ;#Set Log for RF propagation info
## Setting the basic parameters for the model.
puts "Beginning The 25 Node Simulation..."
puts "Defining Variables and Settings..."
# Variables
set val(chan)     Channel/WirelessChannel          ;# Channel type
set val(prop)     Propagation/Ricean             ;# Radio propagation model
# Values of the 802.11 card
Phy/WirelessPhy set L_ 1.0                 ;# System Loss Factor
Phy/WirelessPhy set freq_ 2.472e9          ;# Channel-13. 2.472GHz
Phy/WirelessPhy set bandwidth_ 11Mb        ;# Data Rate
Phy/WirelessPhy set Pt_ 0.031622777        ;# Transmit Power
Phy/WirelessPhy set CPThresh_ 10.0         ;# Collision Threshold
Phy/WirelessPhy set CSThresh_ 5.011872e-12  ;# Carrier Sense Power
Phy/WirelessPhy set RXThresh_ 1.15126e-10   ;# Recieve Power Threshold
set val(netif)    Phy/WirelessPhy                  ;# Network interference type
set val(mac)      Mac/802_11                       ;# Mac Layer type
set val(ifq)      Queue/DropTail/PriQueue          ;# Interface Queue type
set val(ll)       LL                               ;# Link Layer type
Antenna/OmniAntenna set Gt_ 1              ;# Transmit Antenna gain
Antenna/OmniAntenna set Gr_ 1              ;# Reciever Antenna gain
set val(ant)      Antenna/OmniAntenna              ;# Antenna Model
set val(ifqlen)   50                               ;# Max number of packets
                                                      in ifq
set val(nn)       25                               ;# Number of Mobile Nodes
set val(rp)       AODV                            ;# Routing Protocol
set val(x)        500                              ;# x dimension
                                                      of topography
set val(y)        500                              ;# y dimension
                                                      of topography
set val(stop)     350                              ;# Time of
                                                      simulation end
set val(move)  "/home/ns-allinone-2.31/aaron/StatAnalysis/exp1ms15"
set val(traff) "/home/ns-allinone-2.31/aaron/StatAnalysis/exptraff"
set val(RiceanK)        6                          ;# Ricean K factor
set val(RiceanMaxVel)   2.5                        ;# Ricean  Propagation
                                                      MaxVelocity Parameter

# Ricean  Propagation: Maximum ID of nodes (Total number of nodes) used to
# compute pairwise table offsets.
set val(RiceMaxNodeID)  [expr {$val(nn)-1}]        ;
set val(RiceDataFile)   "/home/ns-allinone-2.31/ns-2.31/mobile/rice_table.txt"             ;# Ricean Propagation Data File

#######Begin Simulation Description#######
puts "CREATING SIMULATION INTANCES................."

set ns_           [new Simulator]                  ;# Simulator instance
set tracefd       [open rid1ms15.tr w]         ;# Wireless trace
set namtrace      [open exp1ms5NF.nam w]        ;# Nam trace
$ns_ use-newtrace
$ns_ trace-all $tracefd                            ;# All traces saved
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
#######Set up Topography Model#######
set topo          [new Topography]
$topo load_flatgrid $val(x) $val(y)
####### Set GOD for simulation#######
set god_ [create-god $val(nn)]
```

```
puts "CONFIGURATION OF NODES......................."
####### Configuration of Nodes#######
$ns_ node-config -adhocRouting $val(rp) \
                 -llType       $val(ll) \
                 -macType      $val(mac) \
                 -ifqType      $val(ifq) \
                 -ifqLen       $val(ifqlen) \
                 -antType      $val(ant) \
                 -propType     $val(prop) \
                 -phyType      $val(netif) \
                 -channelType  $val(chan) \
                 -topoInstance $topo \
                 -agentTrace   ON \
                 -routerTrace  ON \
                 -macTrace     ON \
                 -movementTrace ON
# Set propagation settings
if { $val(prop) == "Propagation/Ricean"} {
    set prop_inst [$ns_ set propInstance_]

    $prop_inst MaxVelocity  $val(RiceanMaxVel);
    $prop_inst RiceanK      $val(RiceanK);
    $prop_inst LoadRiceFile  $val(RiceDataFile);
    $prop_inst RiceMaxNodeID $val(RiceMaxNodeID);

 }
####### Sets the configuration for ALL nodes#######
for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}
####### Set the movement and traffic model #######
puts "Loading The Node Movements..."
source $val(move)
puts "Loading Packet Transmission Method and Node Connections..."
source $val(traff)

#Setting the intial node position for nam
for {set i 0} {$i < $val(nn)} {incr i} {
    #30 defines the node size for nam
$ns_ initial_node_pos $node_($i) 30
}
puts "Telling NS-2 When To End The Simulation..."

#telling na the nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}
$ns_ at 350.01 "stop"
$ns_ at 350.01 "puts \"END OF SIMULATION\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
}
$ns_ run
```

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)

____ ________________
Student Signature Date

Disagree (Permission is not granted.)

__________Aaron Phillips___________________________ __7-17-08___________
Student Signature Date