

# CORRECT, EFFICIENT, AND REALISTIC WIRELESS NETWORK SIMULATIONS

A Thesis  
Presented to  
The Academic Faculty

by

Dheeraj S. Reddy

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Electrical and Computer Engineering

Georgia Institute of Technology  
May 2007

Copyright © 2007 by Dheeraj S. Reddy

# CORRECT, EFFICIENT, AND REALISTIC WIRELESS NETWORK SIMULATIONS

Approved by:

Dr. John Copeland, Committee Chair  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. George F. Riley, Advisor  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. Douglas Blough  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. Kiran S. Panesar  
Corporate Technology Group  
*Intel Corporation*

Dr. Henry L. Owen  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Date Approved: 9 January 2007

*To my mother and father who taught me perseverance. To my brother whose has  
always been there.*

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my Ph.D. advisor, Dr. George F. Riley, for his constant support, guidance, and inspiration. Dr. Riley's advising has allowed me to explore many other avenues in computer engineering which may or may not have contributed to my thesis. However, its a tribute to his excellent teaching skill that, I did not lose my way and have come this far. He has made sure that I do not lose my way wandering while exploring other systems. I have greatly benefitted from his teaching, preaching and technical skills. I have learnt immensely from his approach towards debugging and his persistence. His excellent technical advice as a researcher and a teacher have been invaluable, without which, I would never have finished. I look forward to continuing my association with him in the future.

I would like to thanks Dr. Douglas Blough with whom I have had many fruitful discussions regarding my research. His opinions and insight has been of immense help. I would also like to thank Dr. John Copeland who chaired my PhD. proposal committee and helped narrow my focus towards the final phases of my research. Thanks to Dr. Henry Owen, who helped me with his advice during my first few weeks at Georgia Tech to chose a PhD advisor.

I would like to thank Dr. Kiran Panesar for agreeing to serve on my Ph.D. dissertation defence committee and his mentorship during my stay at Intel Corporation. His advice and approach to problem solving have been of immense help.

I would like to thank all my colleagues in the MANIACS laboratory, especially Robby Simpson with whom I have had many a fruitful discussion. Without their help and support the research environment would not have been so conducive. I have also had very fruitful discussions with Young-Jun Lee, Julian Grizzard, Arun Subbiah. I

would like to thank them all.

Finally, I would like to express my sincere gratitude to my parents, Rama Devi and Jayaram Reddy, and Deepu without whose love and encouragement I could never have sustained my course. Special thanks to my mom who has spent endless hours listening to my rants and frustrations despite her being a computer illiterate.

# TABLE OF CONTENTS

|  |      |
|--|------|
| DEDICATION . . . . .   | iii  |
| ACKNOWLEDGEMENTS . . . . .   | iv   |
| LIST OF TABLES . . . . .   | viii |
| LIST OF FIGURES . . . . .  | ix   |
| SUMMARY . . . . .  | x    |
| I INTRODUCTION . . . . .   | 1    |
| 1.1 Background and Motivation . . . . .  | 1    |
| 1.2 Research Problems . . . . .  | 4    |
| 1.2.1 Differences in Wireless Simulation Models . . . . .                                  | 4    |
| 1.2.2 Realistic Wireless Link Models . . . . .   | 5    |
| 1.3 Research Contributions . . . . .   | 5    |
| 1.3.1 Wireless Simulation Models in GTNetS . . . . .                                       | 5    |
| 1.3.2 Identifying differences in Wireless Simulation Models and<br>their Effects . . . . . | 6    |
| 1.3.3 Realistic Wireless Link Models for Network Simulations . .                           | 6    |
| 1.4 Thesis Outline . . . . .   | 6    |
| II WIRELESS NETWORK SIMULATION IN GTNETS . . . . .   | 8    |
| 2.1 The Georgia Tech Network Simulator . . . . .   | 8    |
| 2.2 Wireless Simulation Models in GTNetS . . . . .   | 9    |
| III DIFFERENCES IN WIRELESS NETWORK SIMULATION MODELS                                      | 12   |
| 3.1 Introduction . . . . .   | 12   |
| 3.2 Related Work . . . . .   | 14   |
| 3.3 Simulation Experiment and Network Topology . . . . .                                   | 16   |
| 3.4 MAC Behavior under no contention . . . . .   | 17   |
| 3.4.1 Experiment Description . . . . .   | 17   |
| 3.4.2 Results and Discussion . . . . .   | 18   |

|       |  |    |
|-------|--|----|
| 3.5   | MAC Behavior during Contention Resolution . . . . .    | 22 |
| 3.5.1 | Experiment Description . . . . .                       | 22 |
| 3.5.2 | Results and Discussion . . . . .                       | 23 |
| 3.6   | Conclusions . . . . .                                  | 28 |
| IV    | REALISTIC WIRELESS LINK MODELS FOR NETWORK SIMULATIONS |    |
|       | 29   |    |
| 4.1   | Introduction and Related Work . . . . .                | 29 |
| 4.2   | Background on 802.11 Physical and MAC Layer . . . . .  | 43 |
| 4.2.1 | Physical Layer . . . . .                               | 44 |
| 4.2.2 | Medium-Access Control (MAC) Layer . . . . .            | 46 |
| 4.3   | Experimental Setup and Evaluation . . . . .            | 48 |
| 4.3.1 | Experimental Setup . . . . .                           | 48 |
| 4.3.2 | Measurement Environment . . . . .                      | 49 |
| 4.3.3 | Experiments . . . . .                                  | 50 |
| 4.4   | Results and Discussions . . . . .                      | 51 |
| 4.4.1 | Received Signal Strength Measurements . . . . .        | 53 |
| 4.4.2 | Retry Measurements . . . . .                           | 58 |
| 4.5   | Simulations . . . . .                                  | 63 |
| 4.5.1 | Simulation Models . . . . .                            | 64 |
| 4.5.2 | Simulation Results . . . . .                           | 66 |
| 4.6   | Conclusions . . . . .                                  | 68 |
| V     | CONCLUSIONS AND FUTURE DIRECTIONS . . . . .            | 71 |
| 5.1   | Thesis Summary . . . . .                               | 71 |
| 5.2   | Future Directions . . . . .                            | 73 |
|       | REFERENCES . . . . .                                   | 77 |
|       | VITA . . . . .   | 82 |

## LIST OF TABLES

|   |  |    |
|---|--|----|
| 1 | IEEE 802.11 parameters used in our simulations. . . . .                | 19 |
| 2 | Theoretical duration calculations (without forwarding delay) . . . . . | 19 |
| 3 | Theoretical duration calculations (with forwarding delay) . . . . .    | 20 |
| 4 | Distribution of backoffs in various bins. . . . .                      | 27 |
| 5 | Average number of rounds completed per run. . . . .                    | 27 |
| 6 | Modulation and Channel Coding for 802.11 Bitrates . . . . .            | 45 |



# LIST OF FIGURES

|    |  |    |
|----|--|----|
| 1  | Simulation Network Topology . . . . .  | 17 |
| 2  | Starting time vs. Round count for experiment one using default simulation parameters. . . . .                              | 21 |
| 3  | Starting time vs round count for experiment one. . . . .   | 23 |
| 4  | Starting Time vs Round Count for Experiment two. . . . .   | 26 |
| 5  | Experiment environment and mobility pattern for RSSI measurements  | 52 |
| 6  | Received Signal Strength Index variation with distance ignoring the antenna orientation and obstructions . . . . .         | 54 |
| 7  | Received Signal Strength Index variation with distance keeping the antenna orientation constant . . . . .                  | 55 |
| 8  | Received Signal Strength Index variation with distance keeping the antenna orientation constant (ralink chipset) . . . . . | 56 |
| 9  | Received Signal Strength Index variation with distance keeping the antenna orientation constant (ralink chipset) . . . . . | 57 |
| 10 | Frame Retries as a function of distance with varying modulation rates for 128 byte UDP packets . . . . .                   | 59 |
| 11 | Frame Retries as a function of distance with varying modulation rates for 256 byte UDP packets . . . . .                   | 60 |
| 12 | Frame Retries as a function of distance with varying modulation rates for 512 byte UDP packets . . . . .                   | 61 |
| 13 | Frame Retries as a function of distance with varying modulation rates for 1024 byte UDP packets . . . . .                  | 62 |
| 14 | Frame Retries probabilities as a function of distance with constant modulation and UDP packet sizes . . . . .              | 63 |
| 15 | Topology and mobility pattern used for the simulation experiments .  | 67 |
| 16 | Received signal strength variations as a function of distance(simulated)   | 68 |
| 17 | Percentage of frames successfully received during the simulation run in the first retry i.e retry count = 0 . . . . .      | 69 |

# SUMMARY

Simulating wireless networks accurately is a non-trivial task because of the large parameter space that affects the performance of such networks. Increasing the amount of detail in the simulation model increases these requirements by many times. Hence there is a need to develop suitable abstractions that maintain the accuracy of the simulation while keeping the computational resource requirements low. The topic of wireless network simulation models is explored in this research, concentrating on the medium access control and the physical layers.

In the recent years, a large amount of research has focussed on various kinds of wireless networks to fit various application domains. Mobile Ad-Hoc Networks (MANETs), Wireless Local Area Networks (WLANs), and Sensor Networks are a few examples. The IEEE 802.11 Physical layer (PHY) and Medium Access Control (MAC) layer are the most popular wireless technologies in practice. Consequently, most implementations use the IEEE 802.11 specifications as the basis for higher layer protocol design and analyses.

In this dissertation, we explore the correctness, efficiency, and realism of wireless network simulations. We concentrate on the 802.11-based wireless network simulations, although the methods and results can also be used for various other wireless network simulations too. While many simulators model the IEEE 802.11 wireless networks, almost all of them tend to make some abstractions to lessen the computation burden and to obtain reasonable results. A comparative study of three wireless simulators is made with respect to the correctness of their ideal behavior as well as their behavior under a high degree of load.

Further, the physical-layer abstraction in wireless network simulations tends to

be very simplistic because of the huge computational requirements that are needed to accurately model the various propagation, fading, and shadowing models. When mobility is taken into account several other issues like the Doppler effect should also be accounted for.

This dissertation explores an empirical way to model the physical layer which cumulatively accounts for all these effects. From a network protocol designer's perspective, it is the cumulative effect of all these parameters that is of interest.

Our major contribution has been the investigation of novel empirical models of the wireless physical layer, which account for node mobility and other effects in an outdoor environment. These models are relatively more realistic and efficient when implemented in a simulation environment. Our simulation experiments validate the models and provide simulation results which closely match our outdoor experiments. Another significant contribution is in understanding and design of wireless network simulation models.

# CHAPTER I

## INTRODUCTION

This chapter presents an introduction to the topic of wireless network simulations, an overview of this dissertation, and the motivations behind our research. It also specifies the research problems we have focussed on and identifies the contributions of our research. Finally, it provides an outline for the rest of the dissertation.

### *1.1 Background and Motivation*

Almost all branches of engineering science have experienced the need for modeling systems. Computer network engineering, being no exception, has followed suit and felt the need for modeling in the past few decades. Early approaches to modeling systems were analytical. The need for modeling real-world behaviors led to stochastic principles being used extensively in analytical modeling. However, it has been difficult to model the unpredictability inherent in real-world phenomena with analytical models. As more and more applications started being built on top of networks, the packet-centric view of a computer network became predominant. Over the past decade this approach has become predominant since it allows for the analysis of the network in a holistic manner.

Discrete-event simulations have been the method of choice when building packet-level network simulators. This has continued to be the case, as we have moved into wireless network simulations in recent times. Early network simulators were based on analytical models that provided essentially simulation-based validation of stochastic principles. However, from a network design and engineering perspective, analytical solutions, although being elegant and simple, are generally inflexible. Thus, they are constrained when a comprehensive study of a system is undertaken. Moreover,

analytical models often miss the environmental and real-world complications that are inherent in a deployed and operational network. Analytical models facilitate the study of systems under a simplified set of assumptions in a controlled environment. Network system design and engineering activity on the other hand often widely explore the design parameter space. The research area of packet-level simulators has emerged to fill this void between the analytical model and engineering requirements. Packet-level network simulators have been used extensively to evaluate the performance and scalability of wired networks. Pawlikowski [42] systematically analyzed the assumptions made in the stochastic simulations of telecommunication networks and pointed out systemic flaws. One of the observations he made was that analytical modelers typically fail to understand the limits of the analytical model. On the other hand, packet-level simulators begin with an engineer's view of the network.

With the widespread deployment of wireless networks and increased research focus on wireless network technologies in recent times, the need has increased for such packet-level simulations to evaluate the performance and testing the scalability of wireless networks. Many new classes of wireless networks have been proposed in the literature that use miniature low-power devices to create self-configurable networks for general purposes as well as for accomplishing specific tasks such as reconnaissance. Mobile Ad-Hoc Networks (MANETS) and Wireless Sensor Networks are two common examples of such networks. The applications that these networks are designed to address can often scale to thousands of nodes. Although evaluation of network protocols on a wireless network of a few nodes is a feasible approach, creating a wireless testbed of a few hundred wireless nodes to study various protocols and their performance is practically infeasible. Thus, packet-level network simulators are the tool of choice for evaluating network protocols, topology, and scalability issues in these environments.

Several wireless network simulators have been developed by the research community and in industry. Wireless network simulations are in many ways more complex

than wired networks and have larger requirements in terms of the computational resources and simulation time. This is because the parameter space that affects the performance of a wireless network is very large. The operating environment itself constitutes a significant part of this parameter space. Subtle variations in the parameter space can lead to very different and often misleading results when evaluating higher layer protocol performance and the scalability of the network. Simpler abstraction of wireless networks that can account for these variations accurately do not exist in the literature. For example, point-to-point links in wired networks can be described with bandwidth, delay, and a queue for reasonable results. On the other hand, such a simple description for wireless network links has not yet evolved.

In order to obtain useful results from a network simulator, it must be correct, efficient, and realistic. The network simulator should be *correct* in the sense that it should faithfully reproduce the behavior of protocol models that have been programmed into the hardware and firmware logic of real network interfaces. An example of this is the 802.11 MAC behavior conforming to the IEEE specifications. Thus, correct simulation models, when faithfully simulated, should provide results that are verifiable in a real network scenario. The correctness definition is not mathematical in this context. The simulation should also be *efficient* in that it should provide results in a reasonable time, consuming reasonable computational resources. Of course, the definition of reasonable changes as computing power becomes cheaper. However, the problems that we need to simulate also are becoming increasingly large. Last, the simulation should be *realistic*, in that the parameter space evaluated should reflect real-world conditions. Assumptions, that do not reflect the true nature of the problem domain limit the results of simulations to academic significance. The goals of correctness and efficiency have in some ways contradictory requirements. A correct, realistic simulation is likely to cause a large drain on resources. In such cases, reasonable compromises are needed so that the simulation results and the inferences

derived thereupon are not superficial.

Considerable research has been done examining the accuracy of wireless MAC algorithms and their simulation models. The relative performances of the various algorithms have been compared and contrasted using wireless network simulations. There has been comparatively very little work that focuses on how a given algorithm or a protocol (or both) performs on two or more different simulators. It is generally assumed that since IEEE 802.11 is a specification, all wireless MAC implementations based on the specification [22] will behave similarly, ignoring the physical layer differences.

## ***1.2 Research Problems***

### **1.2.1 Differences in Wireless Simulation Models**

It is common knowledge that various implementations of network protocols behave differently. This is especially true about stateful protocols such as TCP [45] and IEEE 802.11 MAC [22]. In wired network simulations, the physical and the data link layers have been mostly abstracted to be stateless layers. This problem concentrates specifically on the Medium Access Protocol models. The MAC protocol models for the IEEE 802.3 [23] specification are essentially stateless models in most simulators. However, such a simple abstraction is not possible for the wireless networks. The IEEE 802.11 MAC is a complex stateful protocol that tries to provide a 802.3 kind of interface for higher layer protocols. Since the MAC protocol is at the lowest layers of the protocol stack, a reasonably correct implementation of it is essential for the correct design and evaluation of protocols higher up in the stack.

Network simulators are used in research because of the ease they provide for evaluation of complex networks. Moreover, they facilitate scalability studies and allow exploration of extended parameter space. However, no single simulator satisfies the needs of all the research community. It has been found that simulations of a simple

flooding protocol differ substantially when simulated on different wireless network simulators [7]. This research provides a way to quantitatively describe the differences among three popular simulators, namely, ns-2, GloMoSim, and GTNetS. Using carefully designed experiments, we identify the differences and the reasons for these differences among the MAC models of the simulators under study.

### **1.2.2 Realistic Wireless Link Models**

Wired links have been very well understood in the research community. This has resulted in simplified models for wired links when simulating networks. This is done because the link characteristics and the effect of such abstractions on higher layer protocols are well documented. Moreover, the error rate in physical links is very small. On the other hand, the wireless links are quite difficult to model with simple abstractions. The parameter space that affects the behavior of wireless links is large. Typically, wireless links are characterized by their comparatively high noise, low bandwidth, and short propagation delays. Since these characteristics have temporal as well as spatial properties, it is hard to model them analytically.

In this work, we take an empirical approach to modeling wireless links for a given environment. We select a parameter space and conduct experiments in a selected open field. We develop empirical simulation models by analyzing the results of our experiments. Our experiments are conducted in a channel where there is no interference. Thus, the effects of MAC contentions are filtered. We implement these simulation models in the GTNetS simulation environment. In the end, we evaluate the models with a simple UDP file transfer.

## **1.3 Research Contributions**

### **1.3.1 Wireless Simulation Models in GTNetS**

Detailed protocol models for IEEE 802.11 specifications have been developed for the Georgia Tech Network Simulator as part of this research. The models in particular



support the Ad-Hoc mode of the 802.11 Medium Access Control Specification. The infrastructure mode, while functional, does not support any management and security functionality. Only the Distributed Coordination Function (DCF) mode of operation is implemented since it is the most prevalent method in practice.

### **1.3.2 Identifying differences in Wireless Simulation Models and their Effects**

This work quantifies the differences in the modeling of the 802.11 MAC protocol in the tested simulation scenarios and presents the reasons for diverging behavior. Our studies are based specifically on two sets of experiments designed to understand the behavior of the MAC protocol. We begin by looking at the baseline implementation of the 802.11 MAC without considering the effect of contention resolution. We then consider the contention resolution mechanisms in the 802.11 MAC protocol. We also present the reasons why different 802.11 contention resolution mechanisms behave differently under competing medium contentions.

### **1.3.3 Realistic Wireless Link Models for Network Simulations**

Empirical models for received signal strength have been obtained after careful experimentation in the field. The received signal strength is essentially the RSSI field as mandated in the PHY layer specification of the 802.11 specification [22]. Empirical models for packet error rates have also been derived from data obtained after another set of experiments for varying packet sizes, data rate, and distances. The above models allow for the development of more realistic and efficient simulation models for IEEE801.11 wireless links.

## **1.4 Thesis Outline**

This thesis is organized as follows.

Chapter 2 presents an introduction to the Georgia Tech Network Simulator (GT-NetS). It specifically concentrates on the lower layers of the protocol stack with emphasis on wireless network simulation models.

Chapter 3 addresses the reasons why various wireless network simulators behave differently even though they all model the same medium access control mechanism. It tries to bridge the lack of insight into these differences.

Chapter 4 presents an empirical approach to model wireless link behavior. This chapter has two major components. In the first part the experimental methodology used to gather link data is discussed. In the second part the construction and evaluation of the simulation model is addressed.

Finally, chapter 5 summarizes the thesis and discusses the future research directions.

### WIRELESS NETWORK SIMULATION IN GTNETS

#### *2.1 The Georgia Tech Network Simulator*

The Georgia Tech Network Simulator (GTNetS) is a full featured discrete-event network simulation environment that has been designed considering distributed simulation, simulation-scalability, and realism. Its object-oriented design leads to easy extensions for supporting new variations on existing network protocol models and methodologies. Apart from several existing routing protocols like DSR [27] and AODV [44] which have been discussed extensively in the literature and practice, experimental routing protocol models like DNVR [35] for wireless networks have been implemented. The three defining characteristic of GTNetS that differentiates it from other simulators is its emphasis on extensibility, scalability and realism.

GTNetS is written in the C++ programming language using object-oriented principles. This leads to a lot of code reuse. Consequently, a lot of new functionality can be added with little to moderate effort, thereby leveraging the existing design. An example of the ease with which GTNetS can be extended can be seen in the implementation of GTSNetS [41]. Several efforts have also been made in industry and academia to extend GTNetS for simulating various network protocols in various kinds of networks.

GTNetS has been designed from the ground up with scalability in mind. A detailed analysis of the design aspects of GTNetS that address the issues of scalability when simulating large scale networks is discussed in [50]. Memory requirements for a network simulation engine can grow quadratically with the size of the simulation topology. The efficiency properties of GTNetS with respect to GTNetS largely fall

into three categories :

1. Managing the simulation event list size
2. Managing the memory requirements
3. Fine-grained simulation tracing abilities

This property of GTNetS is especially useful in the simulation of wireless networks because most wireless networks are very large. Unlike wired networks, a rather large application domain of wireless network research has always been on autonomous nodes cooperatively operating a network. Applications envisioned for the ad-hoc and sensor network require networks to scale to hundreds and thousands of nodes. Simulation is the only viable way in which protocols and networks can be evaluated in such scenarios.

The abstractions in GTNetS closely follow real-life network behavior and implementations. The layered protocol abstractions, the host-peripheral interface of most network equipment, and the objects closely resemble real-life networks. While most simulators concentrate on the protocol models and ignore the way they are glued together to form a protocol stack, GTNetS follows the relevant protocols and RFCs to implement a network simulation. Consequently, if a person understands a network that one wants to simulate, GTNetS provides abstractions that can easily be glued to create a real-world network topology.

## ***2.2 Wireless Simulation Models in GTNetS***

Apart from several application layer, transport layer, network layer protocols, GTNetS also supports the IEEE 802.11 specifications-based Medium Access Control (MAC) and Physical (PHY) layer models. These models support both the Ad-Hoc and the Infrastructure mode of 802.11 specifications. The models implemented in

GTNetS use only the Distributed Coordination Function (DCF) for all contention resolution.

The IEEE 802.11 specification requires that this PHY and MAC layer appear to the higher layers of the protocol stack, namely, the Logical Link Control Layer (LLC) as existing IEEE 802 LAN. This requires that the IEEE 802.11 conform to the necessary glue layer. Unlike other simulators, the 802.x protocol models in GTNetS conform to the IEEE specifications. This often manifests as the link layer Service Data Unit (SDU) being larger than in other simulators, which assume ethernet framing.

The *WirelessLink* class is used to abstract the wireless PHY layer. It represents a broadcast domain (similar to an ethernet domain). This implementation makes certain assumptions that are valid only in the case of a wireless LAN. The wireless propagation models are implemented using the abstract class *Propagation*. Several propagation models can be easily implemented by extending this class that enables tuning the behavior of the wireless link to suit a given target environment. A detailed stateful model for the IEEE 802.11 protocol is implemented in the *L2Proto802\_11*. As in most real-life networks, the host's abstraction of the network device *class Interface* hands over the SDUs to the *L2Proto802\_11* object, which handles the MAC part of the protocol.

The IEEE 802.11 specification broadly specifies two ways in which wireless nodes may form a network: the independent basic service set (IBSS) LAN and the the basic service set (BSS) LAN. The latter can be extended to create an extended service set (ESS) LAN. However, researchers use a different nomenclature to address these networks. The former is called an ad hoc network and the latter an infrastructure network. Since most networking research is concentrated on the ad hoc network, most simulators implement only this mode of 802.11 networking. GTNetS on the other hand allows for the creation of infrastructure mode 802.11 local area networks, too. The IEEE 802.11 specification defines two methods for contention resolution, the

distributed coordination function (DCF) and the point coordination function (PCF). Only the distributed co-ordination function is implemented in GTNetS because it is the most widely used one in real networks.

In the extended service set mode, several BSS LANs coordinate via their access point (AP) nodes to provide a large wireless network. A handoff mechanism needs to be implemented for simulating mobility of nodes among different BSSes in a single ESS local area network. GTNetS implements a handoff scheme, that is a subset of the handoff scheme proposed in the IEEE 802.11F recommended practice for Access Point Interoperability [24]. A description of the implementation of this handoff scheme along with its use in implementing a higher layer routing protocols is discussed in [25].

## CHAPTER III

# DIFFERENCES IN WIRELESS NETWORK SIMULATION MODELS

### *3.1 Introduction*

Network simulation is a commonly used method for evaluating protocols and their behavior in topologies under specific conditions. The complexity of recent protocols, their implementations, and network elements make building accurate analytical models a hard task. In this chapter, we note that although 802.11 simulators are widely used, we find that they produce different results. We highlight the differences in three popular simulators, and explain the reasons for their divergent behavior. Along the lines of research in wired networks, various routing and application protocols for wireless networks use simulation models that concentrate less on the lower layers of the protocol stack, namely, the Medium Access Control (MAC) and the Physical (PHY) layer. Dependable wireless network simulations require that accurate models of these wireless PHY and medium-access-control protocol models be developed [55]. Such models allow researchers to network and protocol designers to evaluate their ideas in a realistic manner.

The IEEE 802.11 specification defines a Medium Access Control (MAC) layer that is one of the most popular in theory as well as in practice. It has become the de-facto standard for most wireless implementations. On the other hand, comparing and contrasting the behavior of various higher layer protocols is an essential part of wireless networking research. Since, Medium Access Control is the lowest layer of the protocol stack, it is essential to understand how various implementations behave during ideal

conditions as well as under heavy traffic. The common reference for all implementations is the IEEE 802.11 specification [22]. Identifying the reasons for differing behaviors also gives us insight into the behavior of the actual implementations of the protocols.

Although several wireless network simulators are available in the research community [37, 49, 58, 5, 12], not all higher layer protocols (routing, transport, application etc.) are implemented on all simulators. Thus, comparison is often made between the performances of higher layer protocols implemented on various network simulators. Despite the fact that most network simulators implement the same 802.11 MAC specification [22], unless all the implementations behave in a fairly consistent manner, these comparisons will be inconclusive at best. For our study we have restricted our research space to three simulators, namely, ns-2, GTNetS, and GloMoSim. These differences are due to various factors ranging from differing interpretations of specification to subtle details in the implementation that have been ignored for reasons of simplicity and computational complexity. Over the past few decades wired network models have evolved to an extent that we understand what parameters at the physical and data link layers affect the simulation results. As a result, suitable abstractions have been developed. In comparison, wireless models are newer and provide less guidance about what details can be ignored without affecting the accuracy of the simulation.

While ns-2 [37] and GloMoSim [58] are fairly well known, GTNetS [49] is a relatively new network simulation environment. GTNetS is a scalable network simulation environment designed to support large to very-large scale simulations. The design of the simulator closely matches the real network protocol stacks and hardware. This enables the users and developers to clearly identify issues that may cause the simulator to behave differently than a typical real network. The simulator is completely implemented in object-oriented C++, leading to easy extensions for new models or



modified behavior of existing models. A detailed description of the distinguishing features of GTNetS can be found in [48].

Our studies are based specifically on two sets of experiments designed to understand the behavior of the MAC protocol. We begin by looking at the baseline implementation of the 802.11 MAC without considering the effect of contention resolution. We then consider the contention resolution mechanisms in the 802.11 MAC protocol. We also present the reasons why different 802.11 contention resolution mechanisms behave differently under competing medium contentions.

### ***3.2 Related Work***

Considerable research has been done on the accuracy of wireless MAC algorithms and their simulation models. The relative performances of the various algorithms have been compared and contrasted using wireless network simulations. There has been comparatively little work that focuses on how a given algorithm or a protocol (or both) performs on two or more different simulators. It is generally assumed that since IEEE 802.11 is a specification, all wireless MAC implementations based on the specification [22] will behave similarly, ignoring the physical layer differences.

Heideman et al. [21] looked to answer the validity of their simulation models in ns-2. They strived to answer the question of what level of simulation validation is required. They list comparison of specifications and their implementations as a significant validation criterion. They explain that traditionally, protocols have been specified only to the level necessary to ensure successful communication between nodes and to obtain reasonable performance. This implies that many engineering decisions and optimizations are left for protocol implementers. In most cases, different decisions lead to differences in performance, but without compromising on the basic behavior encoded in the specification. The wireless network models are more prone to such differences because of the large parameter space that affects their performance.

Charles E. Perkins et al. [43] tried to compare routing protocols and tried to identify the various MAC protocol characteristics that might affect them. They observe that the interplay between the routing and MAC layers could affect performance significantly. Further, they observe that the additional MAC control packets cause significant difference when evaluating routing protocols qualitatively. They suggest that careful attention must be paid to the interlayer interactions when designing protocols for wireless ad hoc networks. These interactions are equally true when designing the simulation models of the protocols and when their performance is studied using simulation experiments.

In the context of wireless network simulations, Heideman et al. [20] were the first to look at the effects of detail in simulation. They discuss the various trade-offs in more detailed or abstract simulation models. They evaluate the effect of detail using four case studies of wireless simulations for protocol design. They suggest largely two approaches to cope with the varying levels of detail, first, by using robust networking algorithms that are stressed in similar ways by random error as by detailed models. Second, they suggest visualization techniques that can help pinpoint incorrect details. However, they fail to quantify the effects of these missing details in the network simulations.

Takai et al. [55] present a set of factors at the physical layer that are relevant to the performance evaluation of higher layer protocols. These factors include received signal strength, path loss, interference, noise computation and preamble length. The authors concentrate on ns-2 [37] and GloMoSim [58]. They conclude that the factors at the physical layer not only affect the absolute performance of a protocol, but because their impact on various protocols is non-uniform, physical layer considerations can even change the relative ranking among protocols for the same simulation scenario.

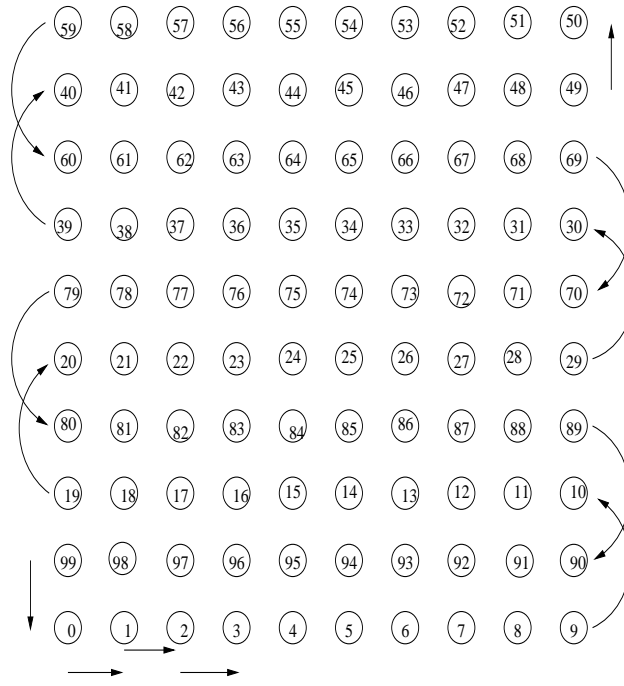
Cavin et al. [7] present the simulation results of a simple straightforward algorithm using ns-2, GloMoSim, and OPNET Modeler [5]. Although they show that significant

differences exist between the simulators, they do not attempt to explain the reasons for these differences. They speculate that this may be due to the mismatch of the modeling in the physical layer or due to the different levels of detail provided to implement the wireless models. They assume that this problem is rather intractable. Hence, they suggest that a hybrid model be used where the simulations run on an emulated testbed.

### ***3.3 Simulation Experiment and Network Topology***

In this section, we discuss our simulation experiments and the results. We also discuss the changes we had to incorporate to the various simulation tools as a result of our analysis of the results.

We constructed two wireless network simulation experiments to study how close various 802.11 implementations were to the published specification and to compare the model implementations in the three simulation tools studied. These experiments were designed to verify the ideal behavior of the 802.11 MAC under no media contention as well as the contention resolution behavior. For both sets of experiments, the topology is as shown in Figure 1. One hundred nodes are placed on a 1000 by 1000 meter grid, as shown, with 100 meter spacing between nodes. The nodes are assigned IP addresses, as shown, with node 0 in the lower left corner, node 50 in the upper right, and node 99 adjacent to node 0. Only the low-order eight bits of the IP address are shown, with the upper 24 bits being assigned an arbitrary, but common value for all nodes. There is no node mobility, and no wireless routing protocol at all in any of our experiments. The transmission range (or transmission power) value was set such that our maximum radio range was 250 meters.



**Figure 1:** Simulation Network Topology

### 3.4 *MAC Behavior under no contention*

#### 3.4.1 Experiment Description

The first experiment, “experiment one,” was designed to be as simple as possible, showing only the proper operation of the RTS-CTS-DATA-ACK exchange described by the 802.11 specification. The experiment works as follows. At simulation time 0, node 0 creates a data message of 512 bytes in length and sends it to node 1. All other nodes do nothing until a packet is received. When node  $n$  receives the data packet addressed to it, it immediately forwards the packet to node  $n + 1$ . As can be seen in our topology, the distance from node  $n$  to node  $n + 1$  is usually only 100 meters, with 200 meter distances occasionally. In all cases, the neighbor is only one hop away, and thus no multi-hop routing protocol is needed. When node 0 receives the packet from node 99, that denotes the end of a *round*, and the beginning of the next round. The starting time of each round is noted, and the experiment continues for 100 rounds.

When this experiment was designed, we anticipated no randomness at all and

completely reproducible and deterministic results, since there should never be any channel contention and thus no sampling of contention window random variable. In fact, this turned out to not be the case. Upon receipt of the DATA frame, all three simulators forwarded the received data packet up the protocol stack to the application layer, while simultaneously starting the transmission of the ACK frame. When the application received the packet, it immediately forwards it to the next hop and sends the packet down the protocol stack to layer two. Of course, the medium is busy at that time sending the ACK frame, resulting in a sampling of the contention window backoff variable and some randomness in the simulation. To circumvent this randomness, we implemented a variation of experiment one that delays the forwarding of the data packet by a fixed amount of time arbitrarily chosen to be 500 microseconds. Since the 500 microsecond delay is longer than the amount of time needed to send an ACK frame, this variation did indeed result in deterministic results, as shown later.

A summary of the parameters that we used in our experiments is given in Table 1.

### 3.4.2 Results and Discussion

First, we performed a pencil and paper analysis of the expected results for experiment one. Table 2 presents these calculations, showing the time needed for each frame during a unicast handshaking exchange. The analysis assumes an 11 Mbps rate for data frames, a 2 Mbps rate for control frames, and a 192-bit preamble sent at 1 Mbps.

As discussed in the previous section, we observed some randomness in the experiment one simulations because of the immediate forwarding of a data packet prior to the transmission of the corresponding ACK frame. We circumvented this randomness by including an artificial *forwarding delay*, which allowed time for the ACK frame to be transmitted before forwarding a new data packet. Table 3 shows the analysis without any forwarding delay. Without a forwarding delay, the backoff random variable is sampled in the range  $[0 \dots 31)$ , in units of a 20 microsecond slot time. Thus the

**Table 1:** IEEE 802.11 parameters used in our simulations.

| Parameter        | Value                                 |
|------------------|---------------------------------------|
| basic Rate       | 2 Mbps<br>(RTS/CTS/ACK rate)          |
| data Rate        | 11 Mbps (Data rate)                   |
| preamble Rate    | 1 Mbps (Preamble rate)                |
| RTS Size         | 20 bytes                              |
| CTS Size         | 14 bytes                              |
| Ack Size         | 14 bytes                              |
| DIFS             | 50 microseconds                       |
| SIFS             | 10 microseconds                       |
| Slot Time        | 20 microseconds                       |
| UDP Header       | 8 bytes                               |
| IP Header        | 20 bytes                              |
| LLC/SNAP Header  | 8 bytes                               |
| Preamble         | 24 bytes                              |
| Data Header      | 34 bytes                              |
| Payload          | 512 bytes                             |
| Forward Delay    | 500 microseconds (removes contention) |
| Initial CW       | 31 Slot times                         |
| Node Spacing     | 100 meters                            |
| Speed of Light   | 300 meters/microsecond                |
| Hops per round   | 100                                   |
| Number of Rounds | 10                                    |

**Table 2:** Theoretical duration calculations (without forwarding delay)

|                      | Duration    | Cumulative Time |
|----------------------|-------------|-----------------|
| RTS Rx Time          | 322 us      | 322 us          |
| CTS Rx Time          | 258 us      | 581 us          |
| Data Rx Time         | 620 us      | 1200 us         |
| Backoff Delay        | 360 us      | 1560 us         |
| Per Round (100 hops) | 0.1560 secs |                 |
| Time for 100 rounds  | 15.60 secs  |                 |

**Table 3:** Theoretical duration calculations (with forwarding delay)

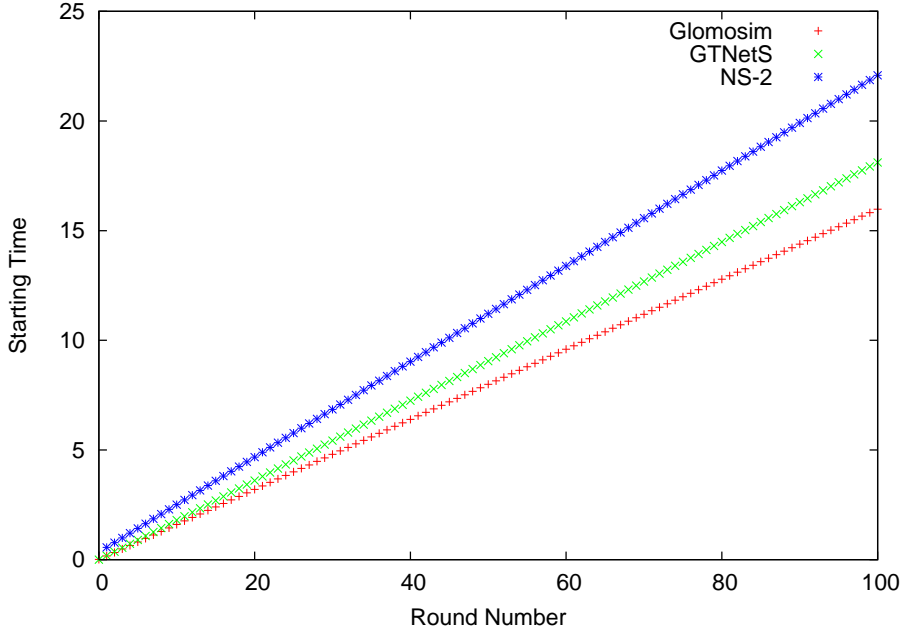
|                     | Duration       | Cumulative Time |
|---------------------|----------------|-----------------|
| RTS Rx Time         | 322 us         | 322 us          |
| CTS Rx Time         | 258 us         | 581 us          |
| Data Rx Time        | 620 us         | 1201 us         |
| Forward Delay       | 500 us         | 1701 us         |
| Per Round(100 hops) | 0.1701<br>secs |                 |
| Time for 100 rounds | 17.01 secs     |                 |

expected value of this backoff is 300 microseconds. The calculations show that the expected time for one round should be 0.1560 seconds without the forwarding delay and 0.1701 seconds with the forwarding delay.

For the initial attempt at experiment one, we simply set the payload size to 512 bytes, set the 802.11 data rate to 11Mbps, and set the transmission range to 250 meters for all simulators. We used default values for all other parameters and ran the experiment on all three simulation tools. The results are shown in Figure 2. The x-axis on the graph is the round number, and the y-axis shows the measured starting time of each round. The results presented are for the variation of experiment one using the forwarding delay of 500 microseconds.

It is easy to see that the results are quite different for the three simulators, with only *GTNetS* matching the value calculated by our theoretical analysis. Further investigation revealed the following differences in the protocol implementations between the various tools.

1. *GloMoSim* always sends the control frames (RTS, CTS, and ACK) at the same rate as the data frames. Both *GTNetS* and *ns2* allow for differing rates for control frames, and both default to sending control frames at 2Mbps. The protocol specification indicates that control frames should be sent at the slowest



**Figure 2:** Starting time vs. Round count for experiment one using default simulation parameters.

rate to allow for all nodes overhearing these frames and reacting accordingly.

2. *ns2* used the *Address Resolution Protocol (ARP)* during the first round to discover the *MAC* address of the next hop neighbor. For subsequent rounds, the cached value of the neighbor's *MAC* address is used. Neither *GTNetS* nor *GloMoSim* used *ARP*, but rather determined the neighbor's *MAC* address by using global knowledge. *GTNetS* has the option to use *ARP* for *MAC* address discovery, but this option is not enabled by default.
3. Neither *GloMoSim* nor *ns2* accounts for the *Logical Link Control – Service Next Access Point (LLC SNAP)* header required for protocol de-multiplexing at layer 2, while *GTNetS* includes this header.
4. *ns2* adds an additional random delay after the expiration of the *DIFS* timer and *SIFS* timers, presumably to account for noise in the clocks used in the hardware implementations.



Once these discrepancies were determined, we took corrective action as follows:

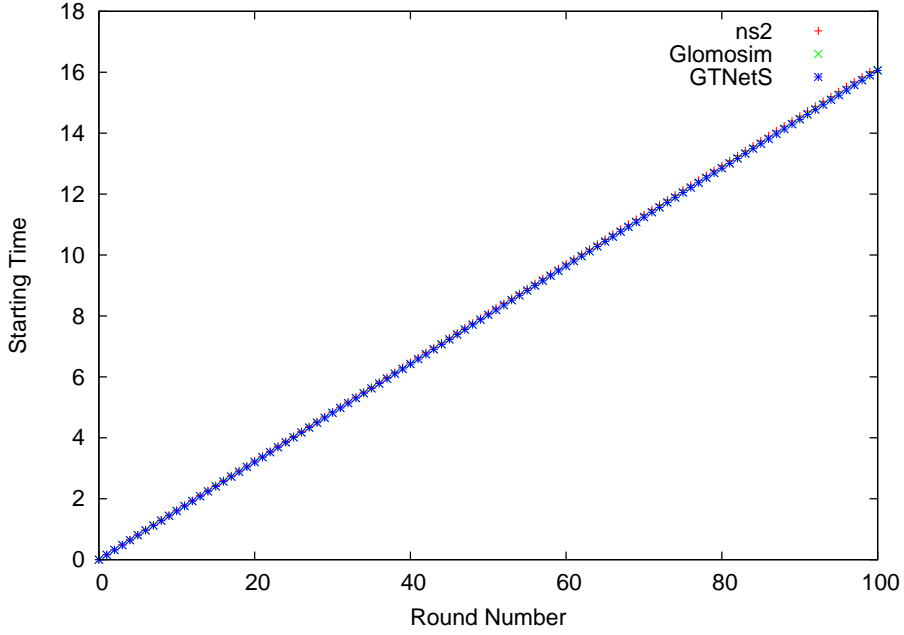
1. Set the rate for control frames (the so-called *Basic Rate*) to 11 Mbps in both *GTNetS* and *ns2* to match the basic rate used by *GloMoSim*. Ideally we would have modified *GloMoSim* to use the slower basic rate, but were unable to determine how to do this easily.
2. Ignore the results measured by *ns2* in the first round. For the *ns2* simulations, we ran 101 rounds and removed the first round data points. This removed the effects of the *ARP* packets.
3. Adjust the payload size for both *GloMoSim* and *ns2* slightly to account for the missing *LLC/SNAP* header.
4. The extra random delays were removed from *ns2*.

Once we made the adjustments listed above, all three simulators produced identical results, as can be seen in Figure 3. Analytical predictions similar to Table 3 with *basicRate* changed to 11 Mbps prove that the results are indeed correct.

### ***3.5 MAC Behavior during Contention Resolution***

#### **3.5.1 Experiment Description**

The second experiment, “experiment two,” was designed to exercise more of the features of the 802.11 protocol specification, specifically the proper management of the *Contention Window*, *backoff timers*, and the *Network Allocation Vector Timer* (*NAV Timer*). The topology and protocol parameters are identical to those in experiment one. Again, at time 0, node 0 creates a data message and forwards it to node 1, as in experiment one. Each node  $n$  forwards the packet to node  $n + 1$  as before, and rounds are measured and noted by node 0 as before. However, in experiment two, all nodes will spontaneously create a packet at a time randomly chosen in the interval



**Figure 3:** Starting time vs round count for experiment one.

[0 .. 10ms) and forward that packet to their next-hop neighbor. Thus, instead of a single packet in the network at any one time, we have 100 packets, all contending for channel bandwidth. Clearly there is significant randomness in this experiment as a result of random starting times for competing packets and random backoff times while waiting for the channel to become idle. This experiment was repeated 100 times for each simulation tool and average results and 90% confidence intervals recorded. Note that in this experiment, there is significant packet loss as a result of MAC layer retransmission limits, and thus frequently we do not complete 100 rounds.

### 3.5.2 Results and Discussion

The results of experiment two are plotted in Figure 4. Since we expected considerable randomness and variation in the results, we executed each simulation 100 times and show the average result and the 90% confidence intervals. For these experiments, all model parameters were the same as in the second experiment one, using 11 Mbps for the basic rate. One point to note is that, as each simulation progresses, some packets

are dropped because of the limits on the *MAC* layer retransmissions. Our application uses the unreliable *UDP* protocol, and therefore there are no retransmission attempts at the transport layer. Thus, there is less congestion in the network as more and more packet are dropped, and the rounds gradually take less time. Further, there is some chance that the data packet originated by node 0 is lost within a round. When this happens, there are no further rounds started (since the prior round never completes) and no additional data points plotted for that experiment.

It is easy to see that the results vary widely between the simulators. As can be seen from Figure 4. GTNetS seems to be the most aggressive of the three simulators in that it finishes its 100 rounds the fastest. On the other hand, ns-2 takes the longest to finish its 100 rounds. GloMoSim is a little slower than GTNetS but has a significant overlap. We spent considerable effort with testing and code inspection of the various protocol models and discovered a number of differences discussed below. In some cases, we were able to correct the differences and in some cases we were not. In cases where the difference was corrected, the corrected version of the simulator was used for the experiments shown in Figure 4.

1. When initiating a backoff, the *GTNetS* implementation doubled the value of the contention window and then sampled the random variable. The other two simulators sampled the random variable and then doubled the contention window. The specification indicated that both actions should be done, but does not state in which order. We changed *GTNetS* to sample and double, as is done by the other tools.
2. All three simulators initiate a backoff after sending an *ACK* frame if another packet is immediately available to be sent. This indicates that a data request from a higher layer was made while the medium was busy sending some other packet. However, the *GloMoSim* simulator samples the backoff timer without

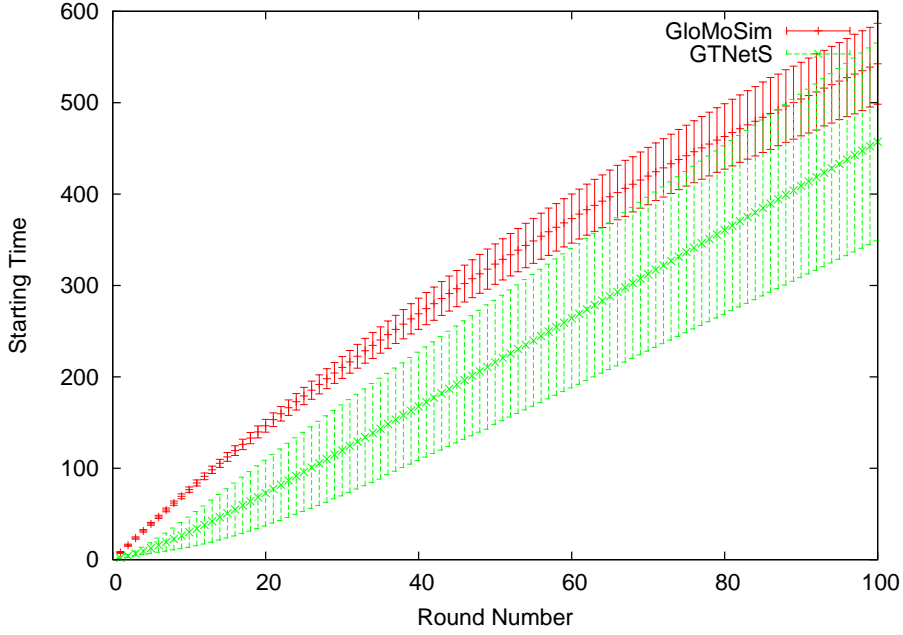
doubling the contention window in this case. We changed *GloMoSim* to double the contention window value in this case.

3. When a backoff timer is active and a new *NAV* timer is scheduled because of an *RTS* or *CTS* frame being overheard, the backoff timer should be suspended while the *NAV* timer is active. However, during channel idle periods, the backoff timer should count down even when the *NAV* timer is counting. This will occur during interframe spacing periods, for example between the *CTS* and *DATA* frames. This is clear in the IEEE 802.11 specification, section 9.2.5.2, as follows:

“A station performing the backoff procedure shall use the carrier-sense mechanism to determine whether there is activity during each backoff slot. If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff time by a SlotTime.”

Neither *GloMoSim* nor *ns2* allows the backoff timer to advance in this situation, resulting in longer backoff periods than the *GTNetS* implementation. We did not correct this difference because of the lack of clear physical channel state transition event interfaces in both *GloMoSim* and *ns2*. In general, the notion of the medium being busy is a combination of virtual carrier sense as well as the physical carrier sense. If either of them indicates that the medium is busy, then the backing off mechanism is paused. It is resumed only when both the carrier sense mechanisms indicate idleness. In any scenario, where the logical expression  $(VCSBusy() \parallel PCSBusy())$  is true because of the latter half of the expression, the actions taken during busy periods are likely to be incorrect.

To understand the backoff behavior better, we divide the interval between the minimum contention window ( $CW_{min}$ ) and the maximum contention window ( $CW_{max}$ )



**Figure 4:** Starting Time vs Round Count for Experiment two.

into six bins. We then instrument the simulator to record how many times the back-off mechanism sampled in each bin. This indicates to us how successful the backoffs were at resolving the medium contention. The backoff bin values for a typical run of 1000 seconds of experiment two are tabulated in Table 4. It can be seen that most of the contentions are resolved in the first backoff period itself. If the backoff mechanism extends to the last bin, then the chance of the contention being resolved is very low. Another interesting observation is that GloMoSim backs off a lot more (almost double) than GTNetS. This is because GTNetS samples the contention window only when it is actually backing off (the sampled value is actually used), whereas GloMoSim unconditionally samples it the first time it receives a packet from the network queue.

As is expected, not all packets finish the 100 rounds in our experiment. To see how the backoff mechanisms help in this regard, we tabulate the average number of rounds that are completed per simulation run. The results are shown in Table 5. It can be seen that in GTNetS, more simulation runs complete their 100 rounds than

**Table 4:** Distribution of backoffs in various bins.

| Bin      | GTNetS | GloMoSim |
|----------|--------|----------|
| 0-31     | 10287  | 21016    |
| 31-63    | 2719   | 3328     |
| 63-127   | 1840   | 1023     |
| 127-255  | 1325   | 510      |
| 255-511  | 867    | 274      |
| 511-1023 | 746    | 253      |

**Table 5:** Average number of rounds completed per run.

| Simulator | Average Completed<br>Runs |
|-----------|---------------------------|
| GTNetS    | 78.00                     |
| GloMoSim  | 62.18                     |
| NS-2      | 56.97                     |

NS-2 and GloMoSim.

To summarize the experimental results, we were able to get nearly exact agreement between all three simulation tools in the simple experiment with no channel contention and no backoffs. In the more complicated scenario, there is still considerable variation in the measured performance of the network. Further analysis is needed to determine if the differences we uncovered but were unable to easily correct can account for these discrepancies, or if there are other as yet undetermined variations in the implementations.

Differing implementations actually have substantial effect on the performance studies of various protocols above the MAC layer. Let us consider the example of a routing protocol that requires some form of controlled flooding to propagate route requests in a network. This is not an unreasonable assumption since many popular routing protocols like DSR require this. From the results of experiment two it is easy to see that in a heavily congested environment, GTNetS and GloMoSim provide better performance for the same protocol specification being implemented. While

studying the performance of such a routing protocol in a heavily congested network, it is easy to see that the same protocol will provide with considerably more encouraging results when implemented in GTNetS or GloMoSim than if it were to be implemented in ns-2.

### ***3.6 Conclusions***

We performed a comprehensive set of simulation experiments comparing the implementation of the wireless IEEE 802.11 protocol in three different simulation tools. Our experiments were designed to be as simple as possible, eliminating any variation resulting from node mobility, multi-hop routing protocols, or physical layer path loss computations. We showed that, after some small changes in the protocol implementations, the three simulation tools report nearly identical results in the simplest case with no channel contention. However, in the more complicated experiment with channel contention, there is still considerable variation in the predicted performance of the network. Since any realistic simulation-based study is likely to have channel contention, we believe that these simulation studies might produce differing results depending on which simulation tool is used. We are not claiming that any one tool is right and the others wrong, but rather that they are different.

We also expect that in a field experiment using wireless devices and 802.11, there may be considerable variation in measured results, depending on the hardware and firmware used for the experiments. Different interpretations of the specification can just as easily be present in hardware implementations as in simulation software. The important issue is not whether different experiments give different results, but in understanding the source and overall effects of the differences.

## REALISTIC WIRELESS LINK MODELS FOR NETWORK SIMULATIONS

### 4.1 *Introduction and Related Work*

Network simulations have been used extensively to evaluate the performance of wireless networks. With the advent of Wireless Ad-Hoc networks and Sensor networks, it has become a necessity because of the scale and complexity of a given network.

There are several popular network simulators [37, 49, 58, 5, 12] used in the research community as well as the industry. Almost all of them include protocol models of IEEE 802.11 specifications in considerable detail. However, most of these have fairly theoretical models of wireless PHY layer. In contrast to existing theoretical models, we present experimental results based on carefully crafted field experiments. These models are incorporated into an existing network simulator (GTNetS). In fact, most simulators provide a simple free space model which is a function of the inverse square of the distance ( $1/r^2$ ). The two models which are most prevalent are the *Friis free-space* model and the *Two-ray ground reflection* model. The *Friis free space* model assumes a flat ideal terrain without any obstacles. It ignores fading and shadowing effects. The *Two-ray ground reflection model*, on the other hand, considers both the direct and the ground-reflected propagation paths between the transmitter and the receiver. The latter has been proven to be reasonably accurate in the case of predicting signal-strengths over distances of several kilometers when the transmitter power is large and the transmitter is mounted at a large elevation. In contrast, wireless links in WLANs use low power transmitters and the distances involved are hardly more than a couple of hundred meters. Recently, shadowing models [34] have also been



incorporated into some simulators [37]. These models account for obstructions in indoor scenarios and outdoor shadowing via a statistical component. However, this shadowing model does not take into consideration any correlations that exist because of the proximity of the communicating nodes. Explicit models for the cross correlation function of the shadowing components affecting the links between communicating nodes should include both the autocorrelation mode for single components [19] and the cross-correlation for multiple components [18]. Ideally, RF (Radio Frequency) propagation models should be based on exact mechanics of RF propagation. They should account for factors that affect the connectivity of a radio link. It is worth noting that even though all the details of RF propagation are not essential for realistic wireless network simulations, the models used should offer realism in connectivity and the changes in connectivity as the topology and environment changes spatially and temporally. Although Per Johansson et al. [26] use the simple radio models, they explicitly state that they are simulating a given detailed realistic scenarios for a conference room, event coverage and a disaster area. It is necessary to note that unless the radio propagation models in wireless network simulations account for the topological constraints and the temporal effects of the surroundings, they will lead to less than realistic results.

For any simulation to be realistic, its physical layer models should take into account two important factors. First, the Packet Error Rate (PER) in the channel which has a strong correlation with the Signal to Noise Ratio (SNR) or the Signal to Interference and Noise Ratio (SINR). Second, it should also address the fading, shadowing (large-scale fading), and path loss effects. Fading is the variation of signal power at the receiving nodes, caused by the node mobility that creates varying path conditions from transmitters. Fading Models with Rayleigh and Ricean distributions have been commonly used in wireless network simulators. The Additive Gaussian White Noise (AGWN) is the noise in an ideal channel where no signal fading occurs.

Path loss defines the average signal loss of a path on a given terrain. Free space and Two ray models have been used in most network simulations.

Pawlikowski et al. [42] were one of the first to address this issue of realistic simulation models questioning the credibility of the simulation results obtained from stochastic simulation models. They then surveyed a large volume of published network simulation results to point out systemic flaws in them. They suggest that simulation results must also be accompanied by explicit mentions of the assumptions made and the stochastic methods used. They should also mention the bias, if any of their parameters might cause, when presenting the results. On similar lines Kurkowski et al. [33] list significant shortfalls in their survey of MANET simulation studies. They conclude that, ‘while the use of simulation has increased, the credibility of the simulation results has decreased.’ They list simulation setup, simulation execution, output analysis and publishing as significant areas where various results lack in credibility.

The impact of different channel propagation models on simulation results, and the primary techniques implemented in the GloMoSim library to achieve high performance simulation are described in [54]. They show that a detailed propagation model allows the effects of different environments (topography and building type) to be explored via simulation. Further, they suggest that such detailed models should be used even in the performance evaluation of higher layer protocols. However, this increases the computational cost for the simulation dramatically. To alleviate this, they suggest network griddling and parallel simulation techniques which contribute towards better runtime performance. In particular, network griddling technique implemented in GloMoSim/PARSEC reduces the computational cost even in sequential simulations.

A measurement based analysis of the error characteristics of an in-building wireless network was done by David Eckhardt et al [13]. They conducted their experiments using the AT&T WaveLan (pre-cursor to the IEEE 802.11 network) hardware. They

categorized the errors into three groups according to the feasibility of their measurements and investigations.

1. Measurable and can be investigated.
  - (a) Attenuation
  - (b) Front end overload
  - (c) narrowband interference
  - (d) spread-spectrum interference
2. Possible impact but cannot be measured
  - (a) Natural background noise
  - (b) Multipath interference
3. Error causes that were ignored
  - (a) Path loss due to dispersion
  - (b) Motion of the communicating nodes
  - (c) Data dependent effects

They infer that distance alone has little effect in a fairly large indoor environment. In general, they observed relatively few bit-errors and the receive threshold was effective in shunting out distant sources. The worst errors were caused by the spread spectrum cordless devices operating in the same frequency band. They also concluded that self-interference is substantial enough to impede building a robust cellular infrastructure. This is because the hardware lacks transmitter power control and multiple spreading sequences necessary to completely isolate adjacent cells.

Nguyen et al. [40] did the first trace based modeling approach for wireless networks. Similar to our approach they also conduct experiments to obtain traces and

then model error rates. Finally, they create simulation models and discuss the validity of their results. However, they use higher level metrics to characterize the network behavior. Their experiments were conducted in the 900 Mhz ISM band. Even though their trace collection metrics are located at the device driver level, their error characteristics are essentially UDP transmission failures. Their study includes packet size and distance variations but does not include the effect of rate modulations. It is unclear if the CSMA/CA protocol implemented in the AT&T WaveLan hardware had retry mechanisms. To validate their models, they compare their trace based approach against the uniform error probability model and an improved two-state Markov model [4]. They infer from the simulation results that their trace driven approach is the closest to the results obtained. They conclude that the results they show are suitable only for evaluating higher-level network protocols like TCP.

Shih et al. [52] advocate a physical layer driven approach to protocol and algorithm design for wireless sensor networks. Wireless sensor networks are power constrained networks. They discuss techniques at various levels of system hierarchy that take advantage of the underlying hardware to produce more energy efficient solutions. Specifically, to protocol design, they show how to take advantage of the hooks and knobs in the physical layer to build more energy-efficient protocols and algorithms. They conclude that if protocol designers treat the physical layer as a black box, system designers may design protocols that are detrimental to energy consumption. Their techniques make use of many existing analytical models with raw data taken from device specifications. However, an empirical model with a sufficiently large parameter space can provide all the necessary hooks and knobs necessary to design an energy efficient protocol.

A different kind of trace-driven wireless channel model was developed by Konrad et al. [31]. They propose and evaluate a Markov based Trace Analysis algorithm. Their approach is to derive a statistical constant from the wireless traces and use

this constant to divide the previously non-stationary trace into stationary subtraces representing lossy and error-free segments of transmission. By analyzing the length distributions of these segments, they can effectively characterize the transitions between them, and create a model that more accurately represents the original trace. They validate the benefits and accuracy of the Markov Trace Analysis algorithm by applying it to a 215 minutes of GSM data traces collected at the Radio Link Layer [14] to generate the MTA GSM channel model. They then show that artificial traces generated by their model has the same statistical properties as the traces collected from the actual network.

Clark et al. [9] explored the feasibility of designing an outdoor cellular network based on the IEEE 802.11 standard. (These IEEE specs were developed for wireless local-area networks) For channels that are typical in cellular networks, they study the radio link power budget and the bit-error performance of three kinds of receivers: (1) the constrained RAKE; (2) the full RAKE; and (3) the ideal equalizer. They conclude that an 802.11-based cellular network with a cell radius of a few km is feasible. From a wireless simulation perspective, their simulation models are explicit analytical models of RAKE receivers and they predict bit-error-rate BER. However, such a model is expensive in terms of computational complexity if we want to incorporate a complete network stack on top of that model. They also conclude that they need to take special care to provide equalization since multipath fading is predominant in outdoor 802.11b networks.

Takai et al. [55] were one of the first ones to address the effects of physical layer modeling in mobile adhoc networks. They focused on the effects of physical layer modeling on their performance evaluation of higher layer protocols and devices. They demonstrate by means of simulations the importance of physical layer modeling even if the evaluated protocols do not directly interact with the physical layer. The details of physical layer modeling in ns-2, GloMoSim and OPNET are discussed. They list

the following physical layer modeling parameters as the reasons for differing higher protocol behavior in their simulation experiments.

1. The presence of 192 bits of the Physical Layer Preamble in the 802.11 PLCP header.
2. Computation of interference and noise at each receiver as this is the basis for SINR(Signal to Interference and Noise Ratio) or SNR (Signal to Noise Ratio) that has a strong correlation with FER (Frame Error Rate)
3. While propagation models such as fading, shadowing and path loss are not part of the radio models, they control the input given to the physical layer models and have great impact on their performance.

They systematically study the effect of these factors on the packet delivery ratio (PDR) of the commonly used AODV and DSR routing. In the latter part, They closely study the differences in the physical layer modeling in simulations on ns-2 and GloMoSim on the overall network performance for scenarios typically used for the evaluation of ad hoc routing protocols. They also try to bridge the differences among them such that the higher layer protocol behavior is similar.

Kotz et al. [32] addressed many of these ideal assumptions made in modeling wireless links. They conducted experimental studies to evaluate several assumptions made in prevalent simulation models. They surveyed a set of research publications over a period of 7 years and concluded that most of the works had either very ideal PHY models or very simple radio propagation models. Based on their survey they listed six axioms upon which most MANET simulation studies and results explicitly and implicitly relied. These axioms, not all of which are orthogonal, affect how higher layer protocol models behave.

1. The world is flat. i.e, simulations consider mostly two dimensional terrain characteristics.

2. A radio's transmission area is circular. i.e, A transmitting node's radio reception area is essentially a circle, whose radius is the radio range.
3. All radios have equal range.
4. If I can hear you, you can hear me. i.e, all radio links are symmetrical in nature. This property is often referred to as link symmetry.
5. If I can hear you at all, I can hear you perfectly. i.e, the error rates are fairly constant over a given radio distance.
6. Received Signal Strength at the receiving nodes is a simple function of distance.

They proceed to examine these axioms and compare it with the data they gather from their outdoor routing experiment. All the above listed axioms are proven to be strongly contradicting to what was observed in reality. Thus they suggest that their results cast doubt on published simulation results that implicitly rely on these assumptions. They conclude with a series of recommendations for the research community. Salient recommendations are:

1. Choose a target environment carefully and list the assumptions about that environment.
2. Use a realistic stochastic model when verifying a protocol
3. Consider a three dimensional terrain model
4. The scenario being simulated must include asymmetric links and links with temporal variations.
5. Use a range of propagation models which are suited to the environment being simulated.

6. Protocol designers must carefully consider all assumptions about the lower layers in their design.
7. Develop standard terrain and mobility models which can be representative of common experimental scenarios.

John J. Lemmon's report to the NTIA [36] provides a simple three-parameter model (Gilbert model) with parameter determination. This enables accurate simulation of the error processes for the wireless links. Land mobile radio and wireless local area networks were investigated using these models. They show that the error distributions derived generally agree with those derived from waveform simulations. From their simulation results, they conclude that the model parameters appear to be well-defined, deterministic functions of signal-to-noise and signal-to-interference ratios. They suggest that the dependence of model parameters on the link conditions could therefore be represented as functional relationships determined by empirical curve fitting. Such relationships would obviate the need to carry out additional waveform simulations.

Deepak et al. [16] conducted comprehensive measurement studies on the Berkeley motes. They illustrate the link layer characteristics are fairly complex and assess their affect on protocol design. Their contour maps of signal strengths around the radio is neither circular, nor convex, nor even monotonically decreasing with distance. They concentrate on the simple flooding protocol to illustrate the complexity involved, since it is one of the most widely used and well studied protocol. A message initiated from a source is rebroadcasted by neighboring nodes and extends outward, hop by hop, until the entire network is reached. Their experiments reveal interesting effects related to the link layer.

1. Highly irregular packet reception contours.
2. Nodes in deep fades.



3. Directionality in transmission resulting in longer links
4. Large degree of asymmetry in long links.

Some of these effects can be modeled by prevalent RF-propagation models like a shadowing model with a high noise variance. Other effects such as directionality in transmission which result from spatially correlated behavior (where nodes along certain directions consistently have better throughput to a transmitter) cannot be captured by Gaussian models of noise. Current models, while sufficient for protocol designs in sparse, mobile networks, might need to be expanded to include a larger range of parameter while testing network protocols. Several empirical studies have also similarly concluded that radio range varies with direction and that the percentage of asymmetric links in a system varies depending on the average distance between the nodes. [59, 8]

A method to estimate signal to noise plus interference levels has been proposed by Mullen [39, 38] to improve the performance of Mobile Ad Hoc network routing protocols. The basic premise is that random variations in signal-to-noise plus interference ratio (SNIR) profoundly affect mobile ad hoc network performance. Hence, good estimates of mean SNIR are essential for reliable routing. This work takes a stochastic modeling approach to modeling signal levels by assuming that noise plus interference (N+I) level is independent of it. This independence is the result of the following:

1. N+I levels can be estimated when the signal is absent
2. Signal estimates must allow for missing values when the transmitter cannot be identified
3. Signal and N+I levels vary for different reasons
4. Separate estimates are less variable than direct estimates of SNIR

Yeung et al. [57, 56] have designed a method to incorporate detailed physical layer models in network simulation of mobile ad hoc networks. They begin by showing that the physical layer variables, including path loss, shadowing, multipath, and Doppler, have significant effects on the predicted overall network performance as inferred from the simulation results. They then propose an approach to simulate details of wireless propagation and radio characteristics in wireless networking studies without compromising on the simulation execution time. The novelty in this approach is the use of two different simulators to do the job that they best can. They use MATLAB/Simulink to obtain detailed orthogonal frequency division multiplexing (OFDM) models and the Qualnet simulator [51] to simulate the network protocol models. However, a link-level OFDM model to simulate every bit in the network is too computationally expensive and leads to unacceptably long execution times. Their proposed approach delineates a method in which simulators of dramatically different time granularities are combined using simple APIs. Through, runtime performance studies, they show that their proposed approach can improve simulation run-time performance by three to four orders of magnitude without any effect on the fidelity of simulation results. Their simulations do not include detailed 3-D terrain models and does not account for the mobility of communicating nodes.

Gang et al. [60], investigate the impact of radio irregularity on wireless sensor networks and try to model it for stationary node scenario. They conducted experiments to study the directionality of radio irregularity on the MICA2 platform. The results demonstrated that although the radio pattern is largely random, it exhibits a continuous change with incremental changes in direction. Based on their experimental results, they formulated a radio model for simulations called the Radio Irregularity Model (RIM). RIM takes into account both the anisotropic properties of the propagation media and the heterogeneous properties of devices. Using the RIM model, they explore the impact of radio irregularity on MAC, routing, localization and topology

control performance. They find that while radio irregularity has a significant impact on routing, localization and topology control performance, its impact on MAC protocols is relatively small. They also find that location based routing protocols, such as Geographic Forwarding [29] perform worse in the presence of radio irregularity than on-demand protocols, such as AODV and DSR. They conclude by presenting eight potential solutions to alleviate the problems caused by radio asymmetry.

Gregor Gaertner et al. [15] looked at the problem of link quality in 802.11 mobile ad hoc networks and show that the communication quality of current 802.11 networks is low. They also postulate that users can experience strong fluctuations in link quality as a result. They identify key factors that cause these fluctuations and derive implications for the developments of higher protocols. Their experiments were conducted in an outdoor environment. These key factors are:

1. Users shadowing (blocking) node links due to their own body/node orientations.
2. Other people shadowing node links
3. Automobiles shadowing node links
4. The chipset or the make of the wireless card
5. The height of the nodes

They also note that the type of ground surface, small-scale movements, message length, payload patterns, and communication load have little to no effect. They conclude that higher layer protocols must tolerate frequent disconnections, network partitioning, latency variations which are far more severe than conventional networks.

Measurement studies carried out at the Roofnet Project [1, 6, 2] indicate that theoretical models do not accurately reflect the characteristics of a wireless local area network. This work starts with the understanding that real wireless links differ from their abstracted models in a number of ways. They question the very principles

on which the widely prevalent 802.11 MAC protocols are based. They observe that many routing and link-layer protocols assume the validity of a “neighbor” abstraction that partitions all the pairs of nodes into pairs that can communicate directly, and pairs that cannot. This leads to the design of MAC protocols such as 802.11 which assume that a pair of nodes either hear each other’s control packets (RTS/CTS) or will not interfere each other’s communication. Neighbor abstraction is supported by assumptions about the relation ship between signal-to-noise ratio and bit error rate. Empirical measurements have not conclusively proved whether neighbor abstraction holds or not. There have been measurements that indicate either ways. It further tries to provide an insight into which differences are important enough to worry about, and to draw conclusions relevant to the design of MAC and routing protocols. To illustrate the effect of multipath they use a wireless emulator that enables both realistic and repeatable experimentation [28]. Their experiments suggest that multi-path fading due to reflections in the radio environment are an important cause for intermediate loss rates. The paper presents the following major conclusions for the wireless MAC and routing protocol design:

1. Links with intermediate loss rates are common, with no sharp transition between high and low packet loss rates.
2. Inter-node distance is not strongly correlated with whether nodes can communicate.
3. Most links have non-bursty loss patterns.
4. Links with very high signal strengths are likely to have low loss rates, but in general signal strength has little predictive value.
5. A link is likely to have a significant loss rate at its optimum 802.11b bit-rate.

6. Multi-path fading greatly affects outdoor links and helps explain intermediate loss rates.

It can be concluded from their work that simulation models of wireless links which are based on these theoretical models will not accurately reflect the actual network conditions are likely to yield misleading results.

Cavin et al. [7] present the results of the simulation of a simple flooding algorithm using three popular mobile ad-hoc network simulators. Their simulations show important divergences between the simulators used. The divergences were not only in terms of the numerical results obtained from their experiments, but also in terms of the general behavior of their protocol. These observations led them to remark that “This observation makes the simulation phase less credible as it is difficult to tell which simulator describes the reality better”. They conclude that standalone simulations do not really fit the actual needs of wireless application developers. Instead of simulations, they suggest a hybrid approach where only the lowest layers and the mobility model are simulated and higher layers are executed on dedicated hosts. They also sense that there is a lack of real experiments to prove the feasibility of wireless network protocols.

In this work we present experimental results from several field experiments we conducted. Using the results obtained we created empirical models for a wireless link when the nodes are mobile. Further, we incorporate these models to calculate *packet error probability* and received signal strength in the Georgia Tech Network Simulator. (*GTNetS*). The *packet error probability* indicates whether a packet sent from a mobile node will be a successful transmission.

Most analytical calculations derive *packet error rate* from *bit error rate* using the simple relation

$$PER = 1 - (1 - BER)^n \quad (1)$$

where  $n$  is the number of bits in a packet. However this relationship is hardly accurate in practice, because bit errors are not completely independent and often occur in bursts. Moreover, the errors do not exactly conform to a uniform distribution. When creating a abstracted network simulation model, it is more useful to think in terms of packet errors than bit errors. When a packet is sent in a medium which does not have interference from any other nodes, the only parameters that affect the successful delivery are the modulation and coding rate used, size of the packet, the distance between communicating nodes apart from the environmental factors like weather. Our intuition is based on the principle that when the physical layer conditions are relatively less conducive for a MAC data frame to be transmitted, it will be retried more number of times. If the frame is dropped after the number of retries mandated by the specification (which is 7), then the physical layer characteristics were not good enough for the frame to be transmitted. If a packet is received successfully, we use the received signal strength model to calculate the received strength. This model is limited to the environment in which we have done our experiments and the methodology we have followed to conduct the experiments. These are described in the subsequent sections.

This work also is different from previous such efforts in that all our measurements were made when the node was mobile without any fixed mobility pattern.

## ***4.2 Background on 802.11 Physical and MAC Layer***

In this section, we describe the aspects of the IEEE 802.11 specification [22] that are relevant to this work. The core IEEE 802.11 specification was proposed by the IEEE 802.11 WLAN working group in 1997 and an addendum was added in 1999. These standards describe both the Physical (PHY) Layer and the Medium Access Control (MAC) layer for fixed, portable and mobile stations within a Local Area Network. The IEEE specification supports both mobile as well as portable nodes.

### 4.2.1 Physical Layer

Wireless networks have some fundamental characteristics that make them significantly different from traditional wired Local Area Networks. Some of them are:

1. Unlike a wired Local Area Network (LAN) an address is not equivalent to a physical location. In wireless LANs, an addressable unit, which is a message destination, is not a fixed location.
2. Wireless LANs use a medium that does not have readily observable boundaries beyond which nodes with conformant PHY transceivers are known to be unable to receive network frames.
3. The logical medium (PHY layer) is unprotected from outside signals. This can include cross-channel interference from nearby frequency bands.
4. This communication medium is significantly less reliable than their wired counterparts.
5. The IEEE 802.11 PHYs have dynamic topologies that change over time because of mobility as well as because of propagation characteristics.
6. The nodes lack full connectivity. Thus, the assumption normally made that every node can hear every other node is incorrect.
7. These PHY media have time-varying and asymmetric propagation properties.

It is also important to understand the modulation techniques and channel coding aspects of the physical layer because they have a significant impact on the error rates. The first 802.11 standard specified Direct Sequence Spread Spectrum (DSSS) radios that operate at 1Mbps in the 2.4 GHz Industrial-Scientific-Medical (ISM) radio bands. The 802.11b specification added additional higher bit-rates (upto 11 Mbps). Later, the 802.11g version of the specification added bit-rates that use Orthogonal Frequency

Division Multiplexing (OFDM). Table 6 presents a summary of the modulations and channel coding for the bit-rates used in 802.11. Each bit-rate uses some form of forward error correction (FEC) with a coding rate expressed by  $k/n$ , where  $n$  coded bits are transmitted for every  $k$  bits of data. The bit-rate is the product of the coding rate, bits per symbol and the number of symbols transmitted per second. While DSSS bit-rates send one-symbol at a time, OFDM bit-rates can send 48 symbols in parallel. This explains why DSSS 5.5 Mbps has a lower bit-rate than OFDM 6 Mbps even though 6 Mbps sends fewer bits per symbol and has a longer symbol length. The particular modulation and the coding techniques chosen implicitly denote the bit-rate that is selected. The bit-rate choice affects the amount of data that is being transmitted across the channel. As mentioned above, the 802.11 wireless PHY has significant time-varying and asymmetric propagation properties that include effects of fading, multi-path interference or any other interference that is not additive while Gaussian noise. Thus, predicting bit-errors and thus packet-errors is a non-trivial task when dealing with the 802.11 PHY for a given data rate.

**Table 6:** Modulation and Channel Coding for 802.11 Bitrates

| Bitrate | A/B/G | DSSS/OFDM | Modulation | Bits/Sym | Coding rate | MSyms/sec |
|---------|-------|-----------|------------|----------|-------------|-----------|
| 1       | B     | DSSS      | BPSK       | 1        | 1/11        | 11        |
| 2       | B     | DSSS      | QPSK       | 2        | 1/11        | 11        |
| 5.5     | B     | DSSS      | CCK        | 1        | 4/8         | 11        |
| 11      | B     | DSSS      | CCK        | 2        | 4/8         | 11        |
| 6       | A/G   | OFDM      | BPSK       | 1        | 1/2         | 12        |
| 9       | A/G   | OFDM      | BPSK       | 1        | 3/4         | 12        |
| 12      | A/G   | OFDM      | QPSK       | 2        | 1/2         | 12        |
| 18      | A/G   | OFDM      | QPSK       | 2        | 3/4         | 12        |
| 24      | A/G   | OFDM      | QAM-16     | 4        | 1/2         | 12        |
| 36      | A/G   | OFDM      | QAM-16     | 4        | 3/4         | 12        |
| 48      | A/G   | OFDM      | QAM-64     | 6        | 2/3         | 12        |
| 54      | A/G   | OFDM      | QAM-64     | 6        | 3/4         | 12        |



All 802.11 packets contain a small preamble before the data payload which is sent at a low bit-rate. The preamble contains the length of the packet, the bit-rate of the accompanying data payload, synchronization bits and a CRC-16 frame check sequence calculated over the contents of the preamble. This preamble is sent at 1Mbps in 802.11b and 6Mbps in 802.11g.

#### **4.2.2 Medium-Access Control (MAC) Layer**

The MAC layer specification for 802.11 has similarities to the 802.3 Ethernet wired line standard. The protocol for 802.11 uses a protocol scheme known as carrier-sense, multiple access, collision avoidance (CSMA/CA). This protocol avoids collisions instead of detecting a collision like the algorithm used in 802.3. It is difficult to detect collisions in an RF transmission network and it is for this reason that collision avoidance is used.

The MAC layer operates together with the physical layer by sampling the energy over the medium transmitting data. The physical layer uses a clear channel assessment (CCA) algorithm to determine if the channel is clear. This is accomplished by measuring the RF energy at the antenna and determining the strength of the received signal. This measured signal is commonly known as RSSI. If the received signal strength is below a specified threshold the channel is declared clear and the MAC layer is given the clear channel status for data transmission. If the RF energy is above the threshold, data transmissions are deferred in accordance with the protocol rules. The standard provides another option for CCA that can be alone or with the RSSI measurement. Carrier sense can be used to determine if the channel is available. This technique is more selective sense since it verifies that the signal is the same carrier type as 802.11 transmitters. The best method to use depends upon the levels of interference in the operating environment.

Nodes conforming to the 802.11 MAC also listen for any packet being sent on

the network. If they decode the physical header of the packet, they will mark the medium busy for the duration of the transmission. Each node uses a back-off window to track how long it should defer sending packets after the medium has become idle. When a node is required to transmit a packet, it waits for a Distributed Co-ordination Function (DCF) InterFrame Spacing (IFS) when the channel is determined to be idle. After this the node shall pick a random time within its backoff-window and waits till it expires. This timer decrements only when the medium is determined to be idle. If at the end of the backoff timer, the medium is determined to be not idle, the back-off window is doubled and the whole process of waiting for a DIFS starts again.

To shield the packet losses in the PHY and MAC layer from the higher network layers when transmitting unicast packets, 802.11 uses link-layer retransmissions. After each data packet is received, the recipient sends an acknowledgement to the transmitter indicating that the packet was received correctly. The lack of a receipt of an acknowledgement indicates a packet loss and hence a larger backoff-window for the subsequent retry.

The CSMA/CA protocol allows for options that can minimize collisions by using request to send (RTS), clear-to-send (CTS), data and acknowledge (ACK) transmission frames, in a sequential fashion. Communications is established when one of the wireless nodes sends a short message RTS frame. The RTS frame includes the destination and the length of message. The message duration is known as the network allocation vector (NAV). The NAV alerts all others in the medium, to back off for the duration of the transmission. The receiving station issues a CTS frame which echoes the senders address and the NAV. If the CTS frame is not received, it is assumed that a collision occurred and the RTS process starts over. After the data frame is received, an ACK frame is sent back verifying a successful data transmission. A common limitation with wireless LAN systems is the "hidden node" problem. This can disrupt 40% or more of the communications in a highly loaded LAN environment. It

occurs when there is a station in a service set that cannot detect the transmission of another station to detect that the media is busy.

We concentrate on nodes with continuous mobility in this work, although it can be argued that propagation effects blur the distinction between portable and mobile nodes. For example, even stationary nodes appear to be mobile due to propagation effects. Further, we limit our work to the 802.11b specification where the maximum data rates obtainable are 11 Mbps. Since most of the real-world applications tend to use the Distributed Co-ordination Function (DCF) and because we do not have Point Co-ordination Function (PCF) implemented in *GTNetS* we restrict our measurements and analysis to DCF.

### ***4.3 Experimental Setup and Evaluation***

In this section, we provide an overview of methodology and the measurement setup used to conduct our experiments. We also discuss how we evaluated the methodology.

#### **4.3.1 Experimental Setup**

We used wireless adaptors based on the Atheros chipset [3] and the RaLink [46] Chipset in our experiments. For accurate location measurements, we used a Garmin V [17] GPS device. The laptops use the NetBSD-current operating system. We wrote an NMEA (National Marine Electronics Association) line discipline to obtain the raw NMEA data from the GPS device. We instrumented the RaLink(ral) and the Atheros(ath) drivers with ring buffers to store the retry statistics that we needed for our measurements. When the Atheros and the RaLink cards signal the completion of the transmission of a MAC frame, they indicate to the host if the transmission was ACKed. If the transmission was successful (ACK was received) the descriptor also indicates how many retries were needed. NetBSD also provides a *radiotap* interface which allows us to obtain per packet signal and noise measurements as the driver stamps it. We also disabled the bit-rate adaptation algorithms that are enabled by

default in both the drivers so that we can operate the card in the desired mode. The RTS threshold was set to a sufficiently high value to disable the RTS/CTS exchanges.

The measurement application is multi-threaded. While the main thread keeps up with the job of reading the actual statistic, the slave thread keeps up with the GPS device for the current location readings. The accuracy of the GPS device was accurate for upto 10 feet in our experimental environment. Each entry in our logs was stamped with the time as well as the location when it is stored.

Since we are conducting essentially link-level measurements, it does not matter whether we are operating in ad-hoc mode or the infrastructure mode. We set the transmit power of the radios to 20 dBm and it was kept constant during the entire set of experiments.

#### **4.3.2 Measurement Environment**

All experiments were carried out in the sports field adjacent to the Campus Recreation Center at Georgia Tech. A number of precautions were taken to make sure that the interference from miscellaneous sources is minimal. We took care so that, the orientation of the network adaptors do not change during the course of measurements. We also made sure that there is always line-of-sight maintained between the two communicating stations. For this reason, all experiments were conducted when the football field was unoccupied. The reason for this is that once we get measurements that are as ideal as possible, we can inject interferences and disturbances into our model as needed. On the other hand, the reverse may not be always possible.

It is important to shield the experiments from various interferences. This enables us to constrain the number of parameters affecting our measurements. Control and Data frames in another Service Set Identifier (SSID) can cause our candidate transmission to falsely retry. The resulting retries will not be representative of the path

characteristics but of the congestion in the medium. Interference from communications occurring in neighboring channels and in the same channel can provide false readings especially when retries are being measured. For this reason, we operated in channel 16 of the ISM band. We insured that this channel is spatially separated enough that cross-channel interference is not a concern by measuring the *physical errors* as reported by the cards over an extended period.

When operating in the access point mode, we made sure that the spacing between successive beacons is greater than 1 second. Though this is not configurable in the cards that we were using, we modified the interrupt handler which indicates when a beacon is supposed to be sent. This makes sure that our data transmissions are not affected significantly by the periodic beacons that our own access point sends in the channel in which we are operating.

We verify the data rates at which the radio is operating by checking ping latencies of large sized packets. The data sizes that are actually being sent into the radio are verified by looking at the retry statistics which gather this information from the DMA descriptors.

### **4.3.3 Experiments**

We conducted two sets of experiments for measuring the received signal strengths of successfully delivered packets and the number of retries associated each successful transmission of a data frame.

#### *4.3.3.1 RSSI Measurements*

In this experiment, we stationed an access point 2 meters above the ground level and 10 meters away from any obstruction on a pole. It was set to send beacons every 100 milliseconds. The transmit power of the radio was set to 20 dBm. We used a laptop with Atheros AR5212 chipset based card as well as the RaLink RT2570 chipset based cards to move around in the field in different directions with random

mobility. Throughout the experiment we kept the orientation of the antenna of the laptop constant even while the laptop was mobile. We also made sure that there is always a line-of-sight maintained during the course of an experiment. This was because previous runs of the experiment resulted in widely varying signal strengths. We gathered four sets of measurements with two different cards so that we can be sure that our measurements are not affected by any chipset quirks. The mobility pattern that we followed can be seen from Figure 5. We traversed the same pattern thrice to get a larger sample size.

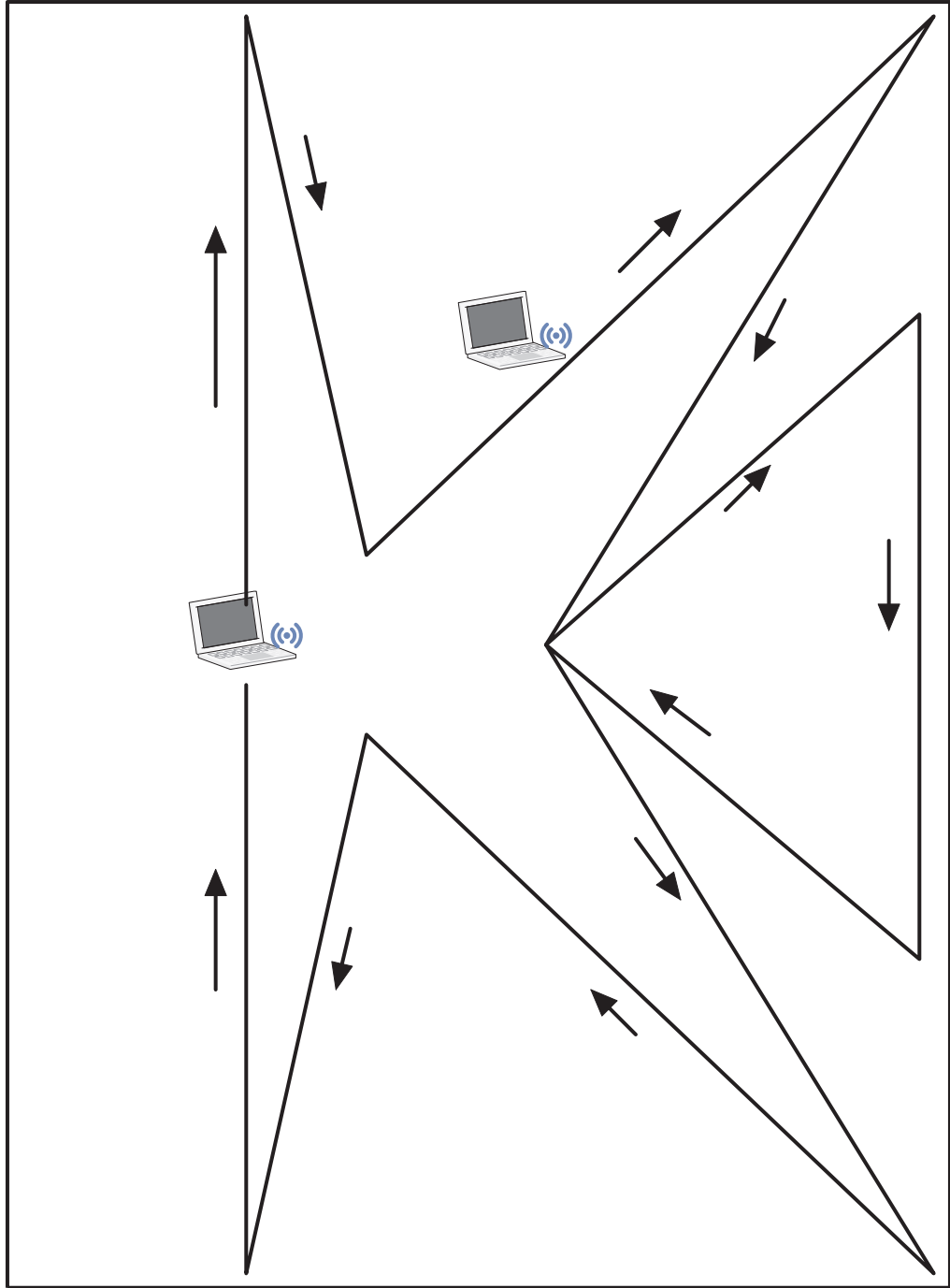
#### 4.3.3.2 *Retries Measurements*

In this experiment, we stationed the laptop acting as an access point 1 meter above the ground such that we can easily access it for varying the bitrates and the mode in which it operates. From the client laptop we then send continuous UDP frames of constant size at the rate of 10 per second to the access point. We leave a certain time interval (10 seconds) in the beginning so that the *probe frames* and *association frames* are exchanged. Since the RTS/CTS exchanges were disabled, the only data that's being transmitted during the course of our experiment are our UDP packets. We verified that this is indeed the case in a laboratory setup.

Now, for each of the 802.11b data rates (1, 2, 5.5 and 11 Mbps) we do one run of mobility for a given packet size. The packet sizes that we chose for our experiments was 128, 256, 512 and 1024. In these 16 experiments we traverse our mobility pattern as shown in Figure 5 twice. The kernel logger logs the retry statistic and the packet size in each transmit complete interrupt handler. Our userspace logger then copies them into a logfile using the *kvm* interface.

## 4.4 *Results and Discussions*

In this section we discuss the lessons we learnt when doing the experiments as well as the results we obtained.



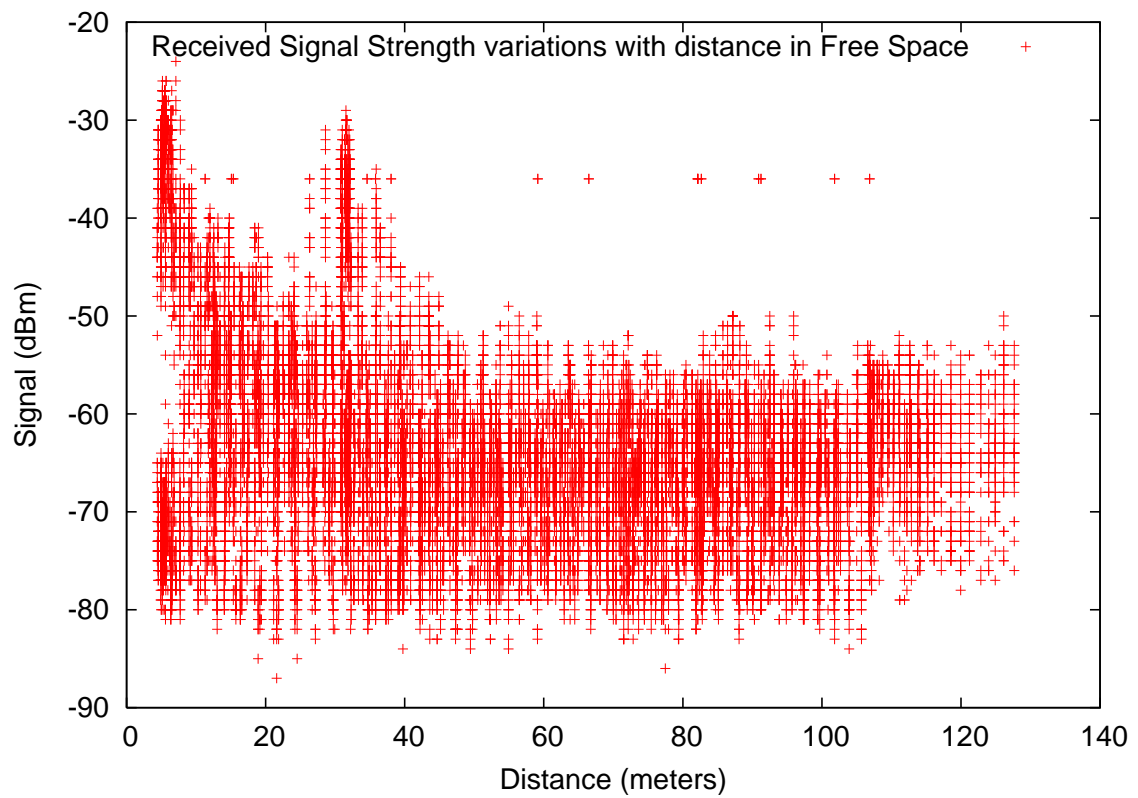
**Figure 5:** Experiment environment and mobility pattern for RSSI measurements

#### 4.4.1 Received Signal Strength Measurements

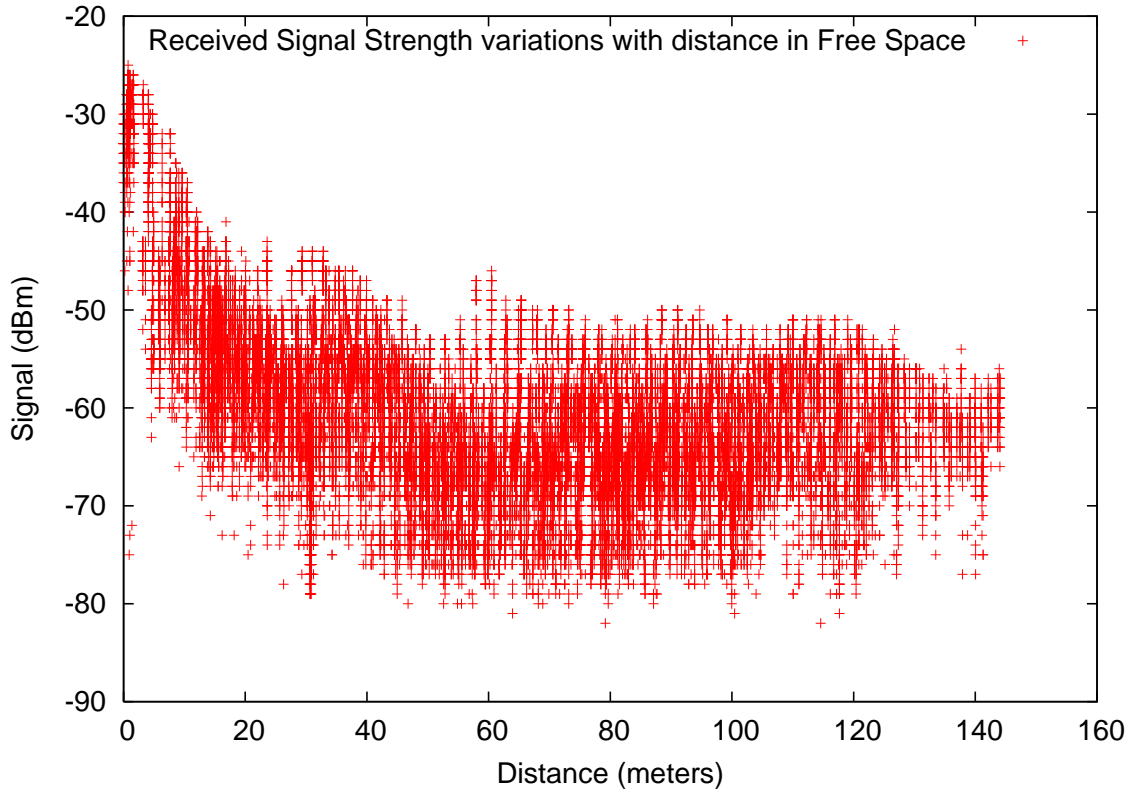
Ideally in free space, the average signal strength should fade with distance according to a power-law model [47]. This relationship holds if we consider ideal conditions that the physics behind the relation holds true. However, in practice a real environment is a poor match to the ideal conditions assumed. Variations due to obstruction, reflection, refraction, scattering, fading and shadowing in practice cause substantial deviation from ideal behavior.

In the beginning, we did not understand the effects of antenna orientation on our measurements. Moreover, we did not take into account that while moving we might ourselves be obstructing the line of sight path between the mobile node's antenna and the access point's. So, we had an access point sending 10 beacons per second and we moved along the field in the mobility pattern as shown in the previous section. This resulted in the signal strength decay graph in Figure 6. It is clear from Figure 6 that the signal strength variations are widely varying from low values to rather large values. This result is similar to that of Kotz. et al. [32] when relating signal strength to distance, except that the measurements made in [32] are averaged signal strengths and their logs contain per-second values. Thus, their signal log contains entries of the most recently received packet on each laptop (mobile node). On the other hand, our measurements are per packet measurements that are obtained when a packet is received by the network adaptor. The DMA descriptors from the chipset contain the signal strength information along with the frame contents. This frame along with other radio specific information is used in our measurements. Kotz also observe that the mean signal strength observed follows the power curve as illustrated by Rappaport [47]. However, they use the mean values of dBm as they measured for each distance bucket. This may be misleading since mean values of received power in mW is not the same as mean values of received power in dBm which is a logarithmic scale.



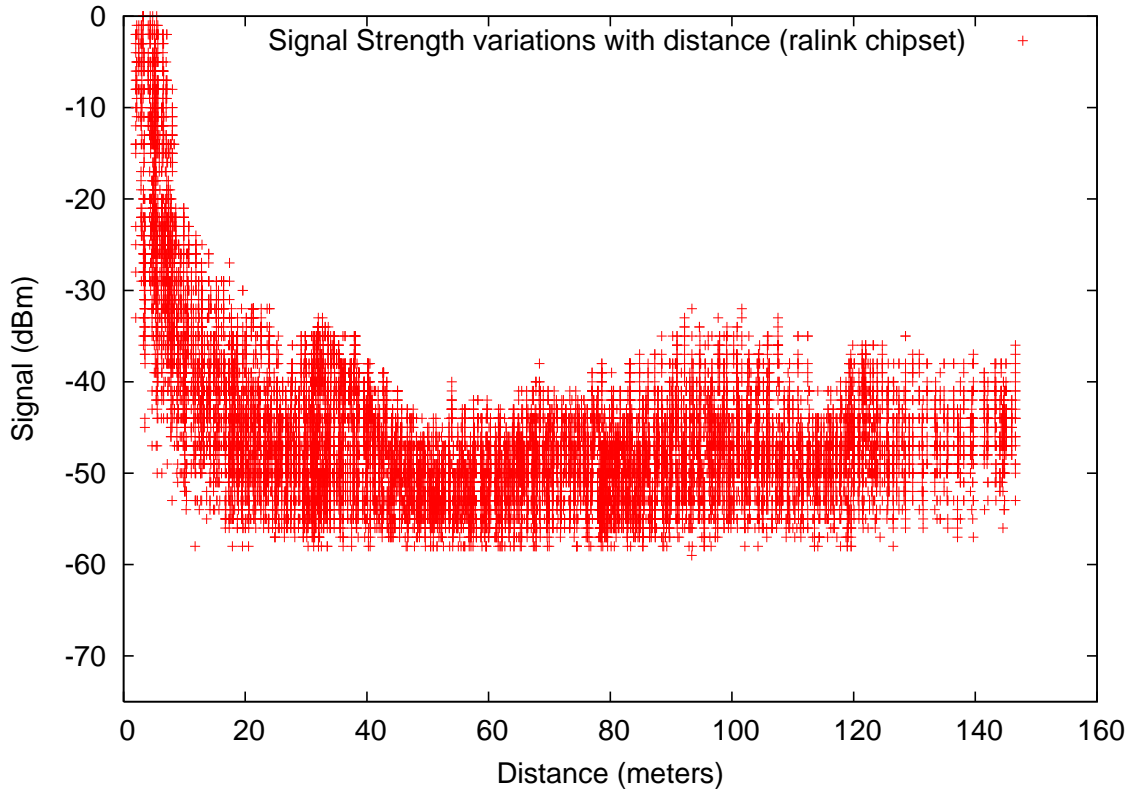


**Figure 6:** Received Signal Strength Index variation with distance ignoring the antenna orientation and obstructions



**Figure 7:** Received Signal Strength Index variation with distance keeping the antenna orientation constant

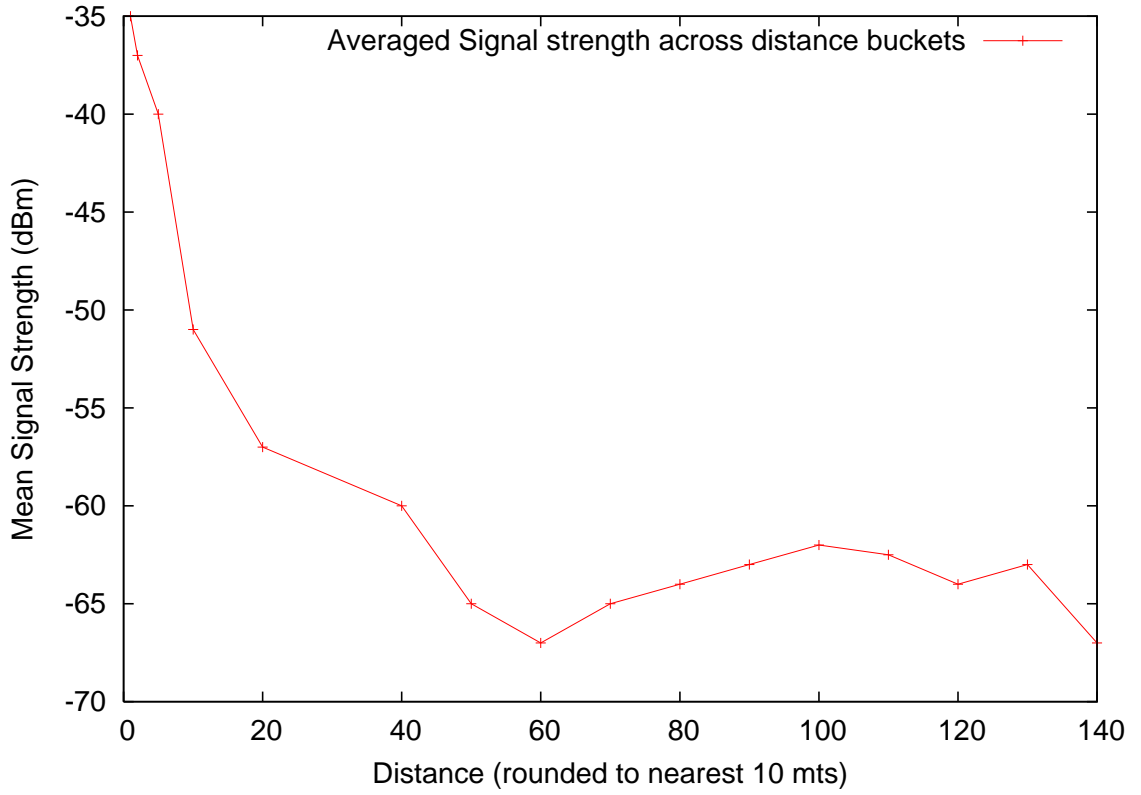
In the next, run of the experiment, we tried to keep the mobile node's orientation with respect to the access point constant. We also made sure that the orientation of the network card remains the same throughout the mobility path. The distance vs. signal strength plot of the resulting values is shown in Figure 7. It is clear from Figure 7 that if we create ideal conditions, we can see that at smaller distances, the signal strength will be higher and at larger distances the signal strength will be weaker. The slope of the curve is linearly decreasing in the initial portion of the curve indicating that in that range the decay in signal strength follows a power law. However, after a certain distance, the signal strength remains almost constant with about 20 dBm variation. It is also clear from Figure 7 that at a given distance the variation (difference between the maximum and the minimum value) of the signal is almost 20 dBm.



**Figure 8:** Received Signal Strength Index variation with distance keeping the antenna orientation constant (ralink chipset)

In order to make sure that the variations and the patterns we see are not because of some chipset quirks, we changed the roles of the nodes acting as accesspoint and as mobile nodes. This allowed us to use the card based on the RaLink chipset as the mobile node. Again, we repeated the experiment making sure that we keep the antenna orientation and the line-of-sight. The variations in signal strength with varying distance when the node is mobile is for the Ralink card is shown in Figure 8. It can be concurred from this plot that the decay in signal strength in the initial part of the graph follows a power law. However for the significant latter portion of the distance range father apart, the signal strength remains constant.

To see the decay patterns more clearly, we rounded the distances to the nearest 10 meter multiples and plotted the mean signal strength in each bucket as a function of the distance. It can be seen concluded from Figure 9 that there is a linear decrease in



**Figure 9:** Received Signal Strength Index variation with distance keeping the antenna orientation constant (ralink chipset)

the initial portion of the plot and for the latter portion it remains relatively constant.

There is large variation in the signal strengths that we see at any given distance. This is due to the fact that the signal strengths were measured irrespective of the direction. Hence, these variations can be concluded to be the directional variations in signal strength. [60]

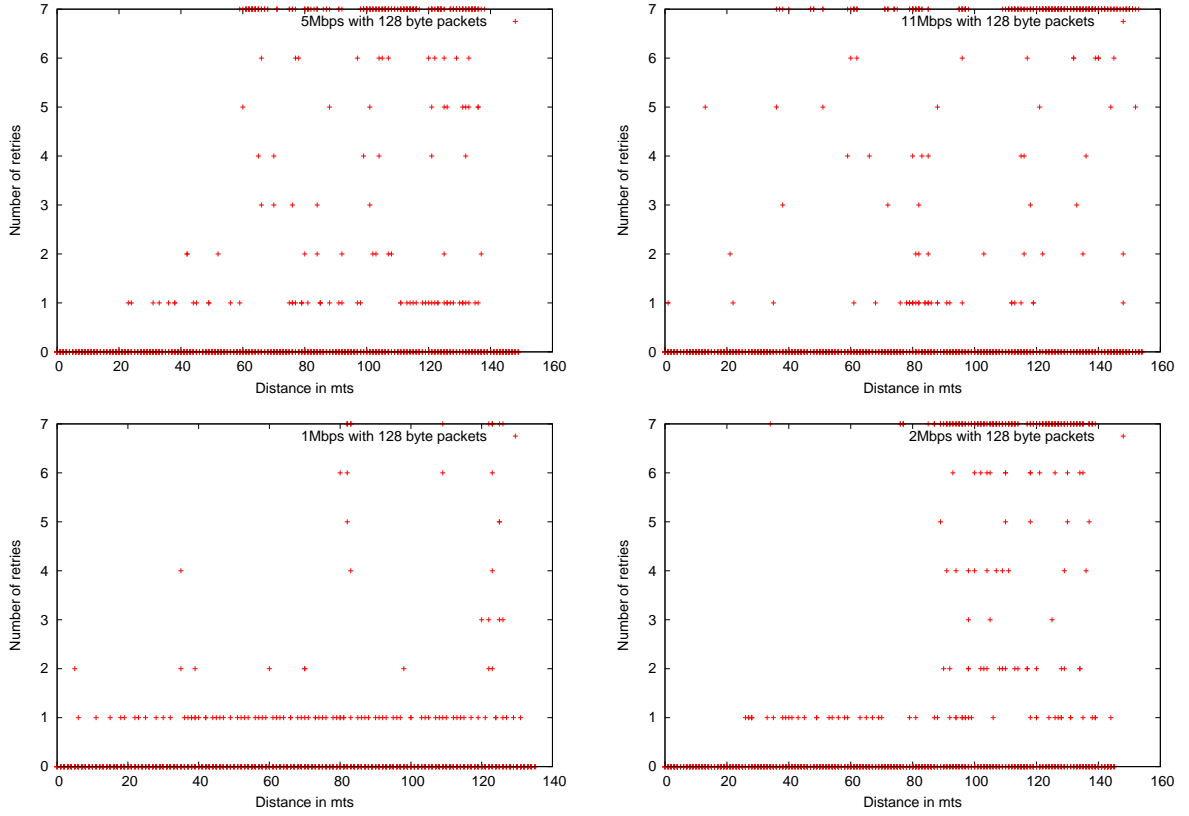
It must be noted that this signal strength is measured only for the successfully received beacons. Thus the variation in signal strength that we see in the plots above cannot be concluded to be the absolute variations in the signal. These are signal variations for successfully received packets. Thus, these can only denote variations from a receiver node's perspective.

#### 4.4.2 Retry Measurements

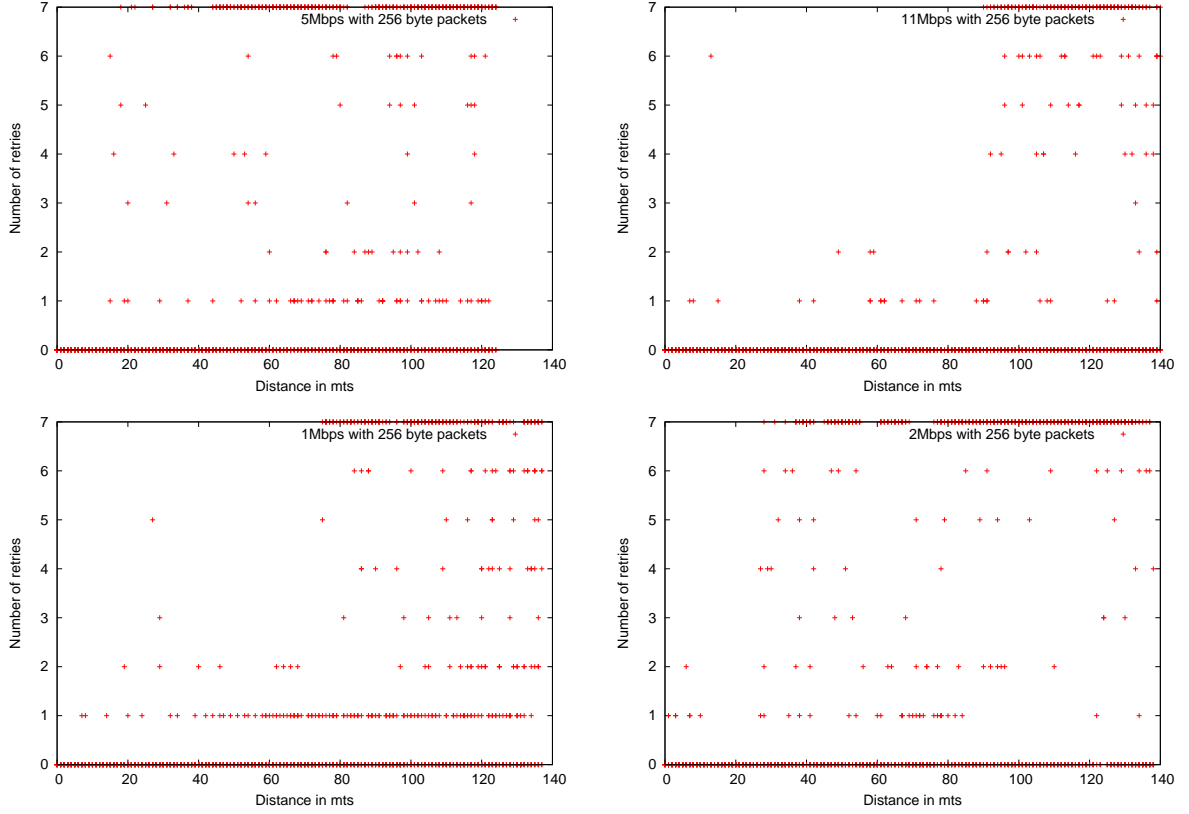
The second set of experiments were designed to understand the variation in packet error rate varies with respect to changing distance, modulation and packet sizes. Sixteen runs of the experiment were done for each combination of data rate (DS1, DS2, DS5, DS11) and packet sizes(128, 256, 512, 1024). For each of the combinations, we measured the number of retries as the mobile node moves across the field in a given mobility path. The results for each are shown in Figures 10, 11, 12, and 13 respectively. The distances in the plots were rounded off to the nearest meter.

It can be easily concluded from the above graphs, that as the distance increases, the number of retries needed to send a particular frame increases. At locations farther from the access point node, the retries are more pronounced than at nearer locations. As a retry is attempted only when a previous try has failed, it can be only because of a PHY error. It is worth noting that we are doing our experiment in a frequency band (channel 15) where there was no MAC layer interference from other wireless nodes operating. Also, the beacons being generated from the access point node were way too sparse to cause any noticeable effect on our measurements. It can also be seen from the plots that, for a given distance range, there are more retries for a packet to be sent at the highest rate (DS11) than at the lowest rate (DS1).

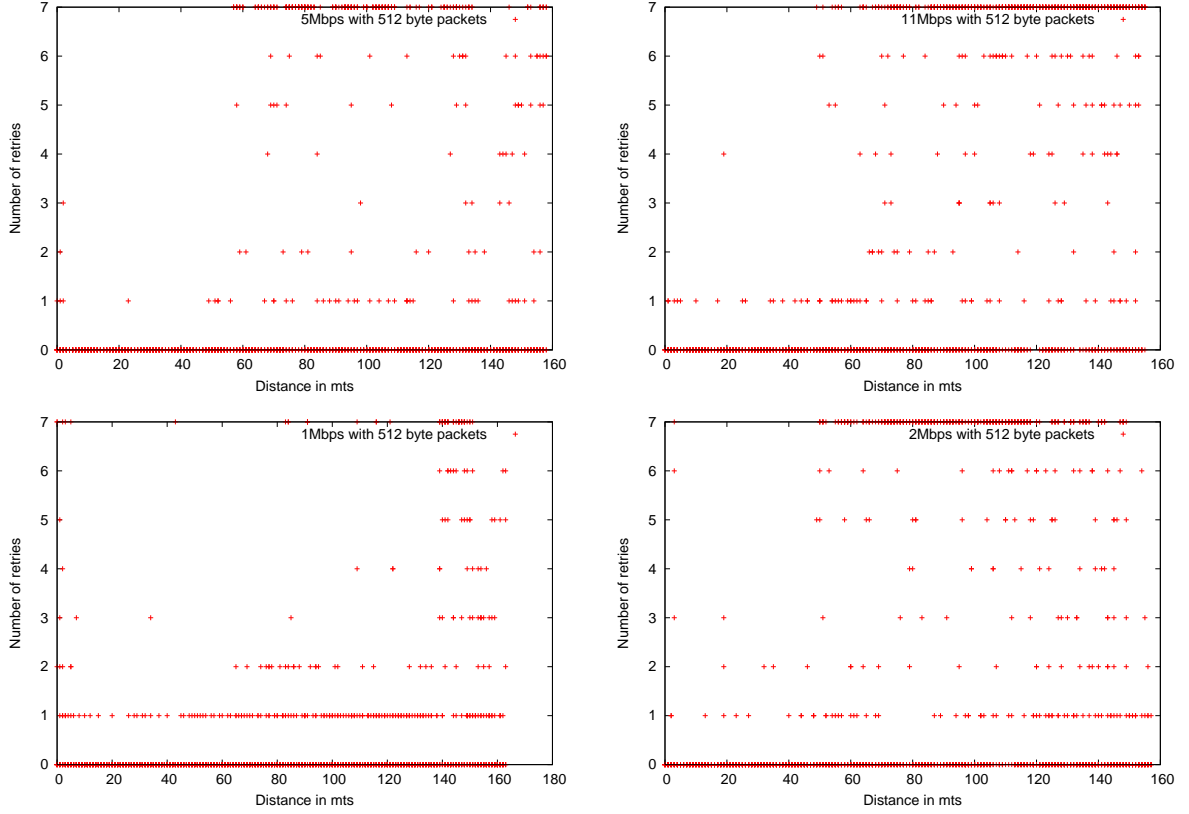
To understand this clearly, we present a few representative plots to show what is the probability that a given packet sent to a destination node at a given distance reaches the destination successfully. These plots plot the distance versus the retry probabilities. It is clear from the Figure 14 that as the distance increases, the probability that a packet requires larger number of retries to achieve a successful transmission increases. We calculated that of all the packet transmissions that went to 7 retries, only a small fraction succeeded (0.001). Thus, we can conclude that as the distance increases, the probability that a packet is delivered without an error



**Figure 10:** Frame Retries as a function of distance with varying modulation rates for 128 byte UDP packets

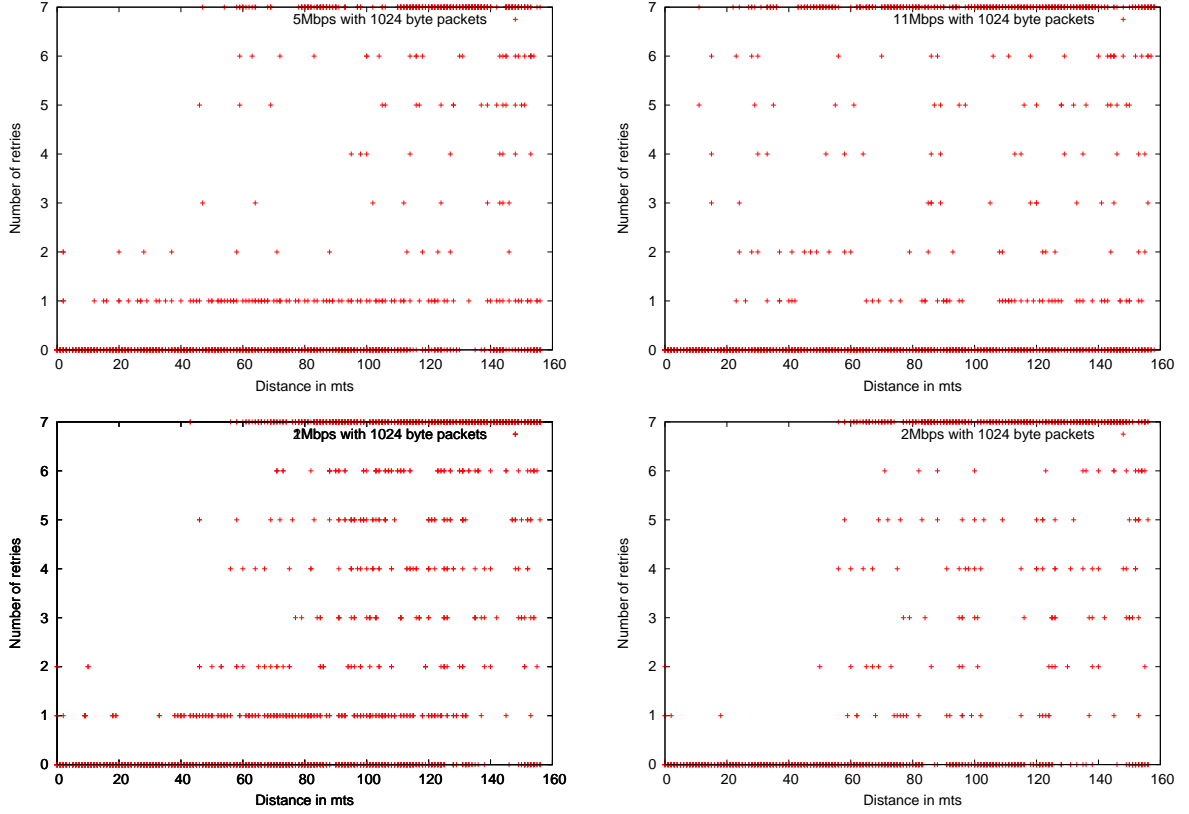


**Figure 11:** Frame Retries as a function of distance with varying modulation rates for 256 byte UDP packets

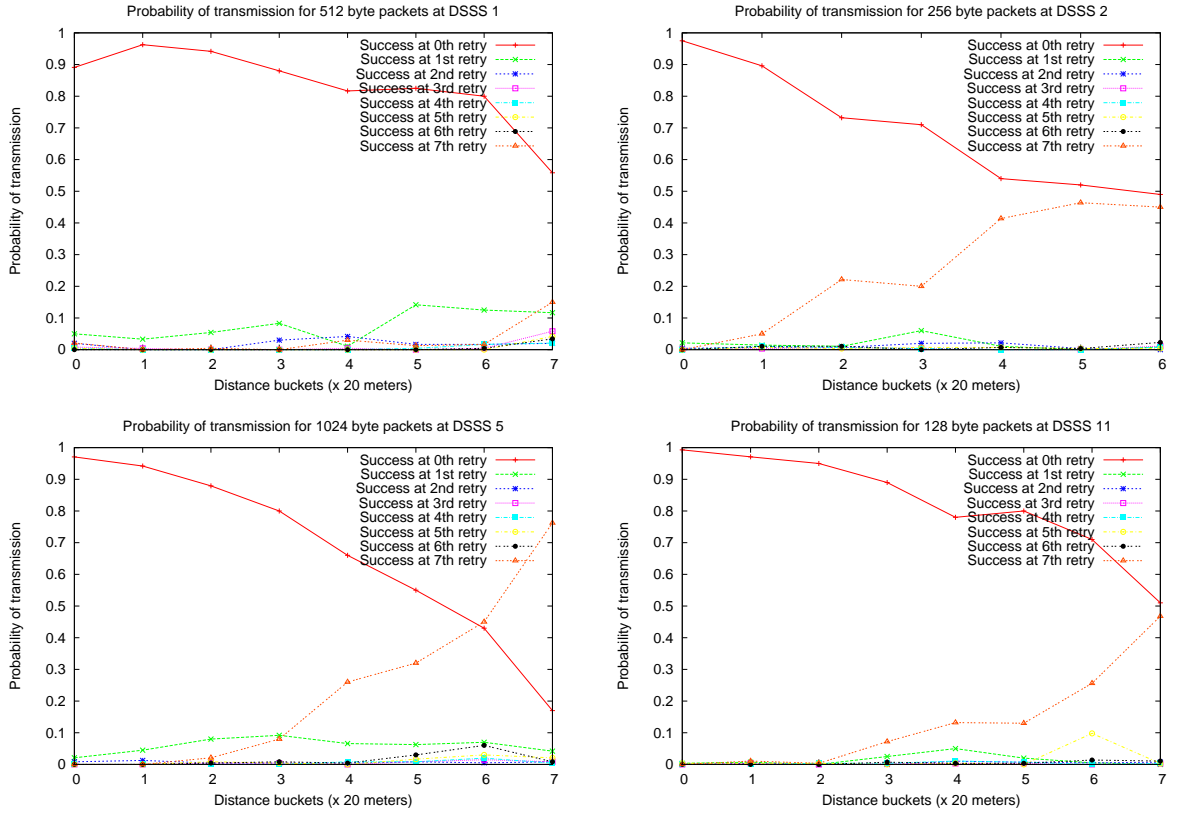


**Figure 12:** Frame Retries as a function of distance with varying modulation rates for 512 byte UDP packets





**Figure 13:** Frame Retries as a function of distance with varying modulation rates for 1024 byte UDP packets



**Figure 14:** Frame Retries probabilities as a function of distance with constant modulation and UDP packet sizes

decreases. It is also clear that the number of retries have relatively little, but significant presence in the plots. Thus, we can conclude that although a significant number of packets do have errors while being transmitted at a given distance, they do get eventually delivered after a few retries.

## 4.5 Simulations

In this section, we describe the simulation models, the way we implemented them, and the way a typical simulation would use them. We also discuss a typical simulation run we conducted using these models.

### 4.5.1 Simulation Models

#### 4.5.1.1 Radio Propagation Model

We use the received signal strength measurements to create radio propagation models for the receive paths. Although, these models are independent of the topology, they are definitely a characteristic of the environment in which the experiments were conducted. All the RSSI are measurements into buckets of 10 meters each. For each bin, we then make a CDF of the received signal strength variation. The model has the following assumptions:

1. Signal strength is a characteristic of the environment and the distance between the signal source and the measurement point.
2. Signal strength that could not be detected by the measurement apparatus does not cause any interference.

The second assumption can be discounted if the measurements were made using a spectrum analyzer. However, the measurements as exported by the 802.11 physical layer convergence layer is what is available for the higher layers if they want to make any decisions based on signal strength. A good example of using this information is when mobile nodes operating in infrastructure mode decide when to disassociate with an access point and associate with another. In our investigations of hardware, we could not find any commodity 802.11 hardware that exports the signal strength values that the CCA (Clear Channel Assessment) algorithm uses to determine. Only the prism chipset [11, 10] exports that value, but it does so only when it successfully receives an 802.11 frame.

We extended the existing propagation models in GTNetS to provide an empirical propagation model. We use this model to determine the signal strength at a given distance. The model essentially calculated the bin from which the signal strength has to be sampled. As discussed earlier, each bin has an associated CDF of RSSI values.

#### 4.5.1.2 *Packet Error Rate Model*

We use the packet retries to derive a packet error rate model for the transmit paths. The measurements from the second experiment have the number of retries for each combination of datarate(DS1, DS2, DS5 and DS11) and packet size with varying distance. At each distance we note the number of retries required for a packet transmission to succeed. The methodology we followed to derive the models is as follows :

We categorize all our distance measurements into 20 meter buckets. Since the maximum distance that we measured is about 155 meters, all our measurements fit into 8 buckets (0-7). Using the retry counts, we then calculate the probability of success of a packet transmission for each retry count at the given datarate, packet size and distance. The probability of failure is the reverse of the success probability. This probability of failure is therefore the packet error rate. Since, it is the packet error rate only that we are concerned in discrete-event network simulation when transmitting the packet over a link, we need not decipher the bit error rate from this. It must be noted that this is the probability of success of a transmission. And a successful transmission in our case is the exchange of a DATA frame as well as an ACK. Therefore, when creating the simulation model, we assume that the data frames are transmitted with this packet error rate, but the ACK frames have zero probability of an error.

An alternate method would have been to simply use broadcast frames in our experiments so that we could accurately calculate the error probabilities of ACK frames too. However, the hardware we used in our experiments did not send the retry counts or the success indications in the processed DMA descriptors. Thus we had to use unicast frames for our measurements.

It is obvious from Figure 14 that there are very few packets which are sent during retries counts greater than 1. This gives us an indication of the observation made in

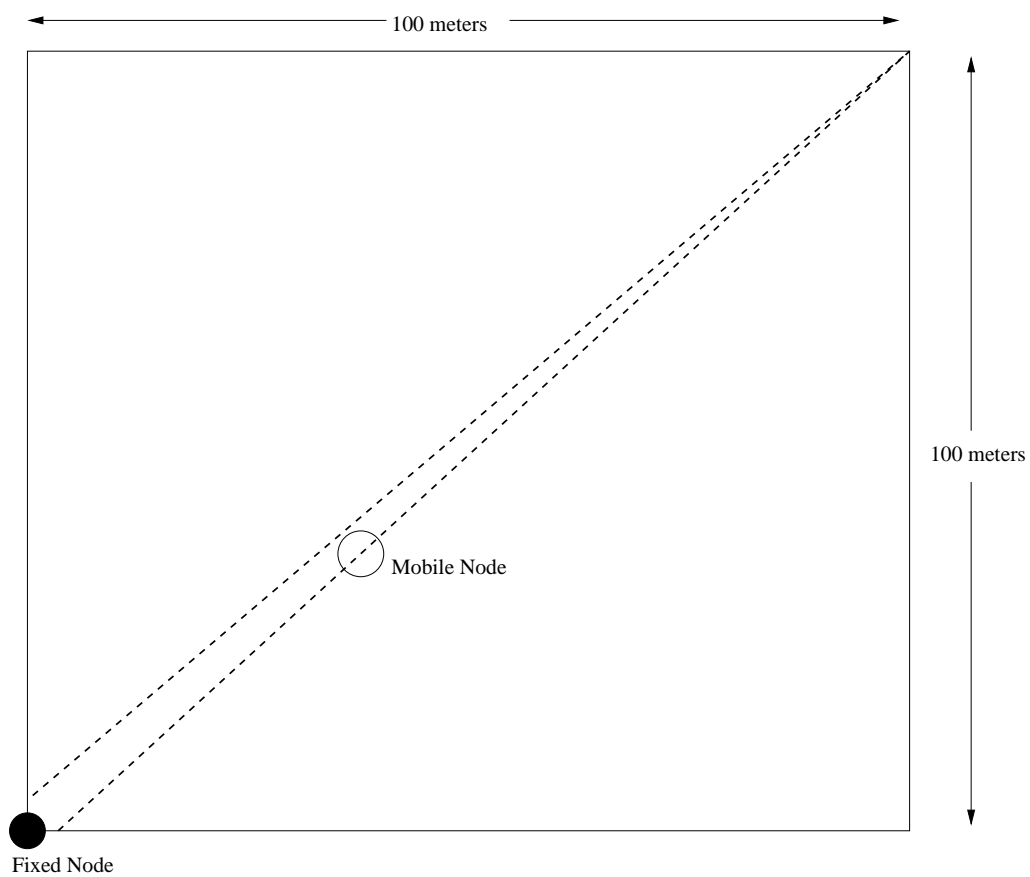
previous studies that errors in wireless networks are correlated. If the first retry is unsuccessful because of a PHY error, a significant proportion of the subsequent retries also experience the same error. However, correlations that exist between different 802.11 frames being sent are not captured by the proposed model. A more sophisticated model which can establish correlations between retries of successive packets will be able to capture those correlations.

#### 4.5.2 Simulation Results

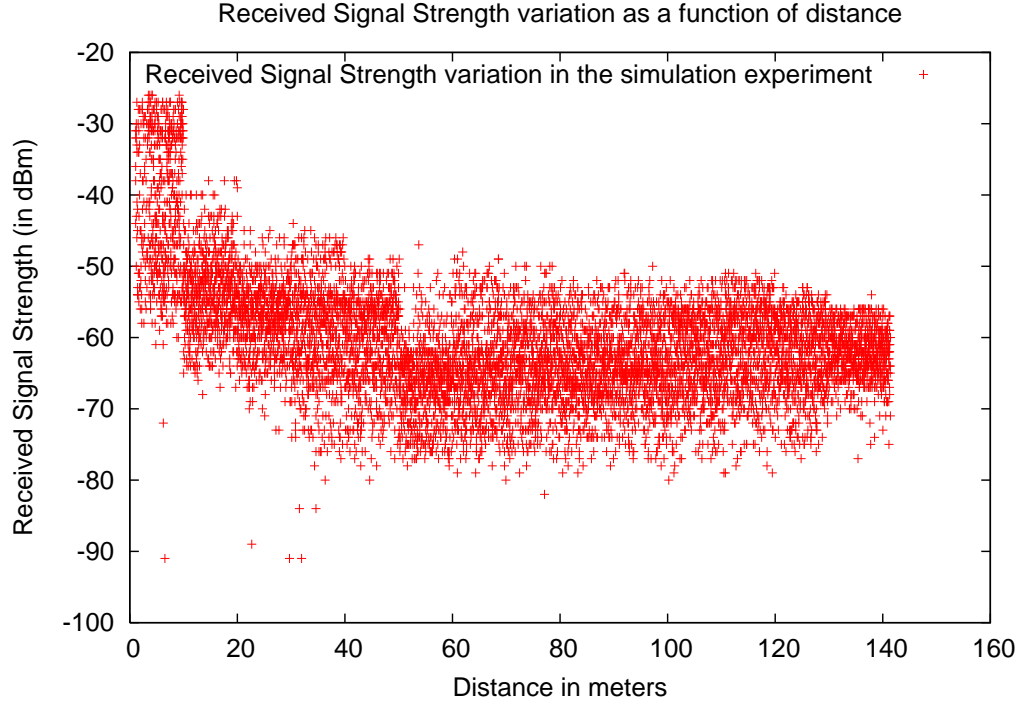
In this section we discuss the simulation experiment we conducted to validate our simulation models described in the previous sections.

In the first simulation experiment we simulate received signal strength. The topology consists of two nodes, one of which acts as the access point and the other mobile. The mobility pattern is as shown in Figure 15. The fixed node (access point) transmits beacon frames at 10 frames per second. The received signal strength at the mobile node is measured continuously for each received packet. The signal strength measurements obtained in the simulation as a function of varying distance are shown in Figure 16.

The second simulation experiment simulates throughput and packet loss. The topology is identical to that used in the first experiment. One of the node is stationary and the other moving. Both the nodes are initially positioned 1 meter apart. The mobility pattern is still the same as in Figure 15. The moving node moves away from the static node for a radial distance of 141 meters and then returns back to its original position. Since our model does not take into account the directional variations in the propagation and assumes that the speed is constant at 1.5 meters per second, our simulation experiment uses the same scenario. We send UDP frames at 10 packets per second from the mobile node to the static node. We sent 128, 256, 512 and 1024 byte frames at various datarates in our simulation experiment.



**Figure 15:** Topology and mobility pattern used for the simulation experiments

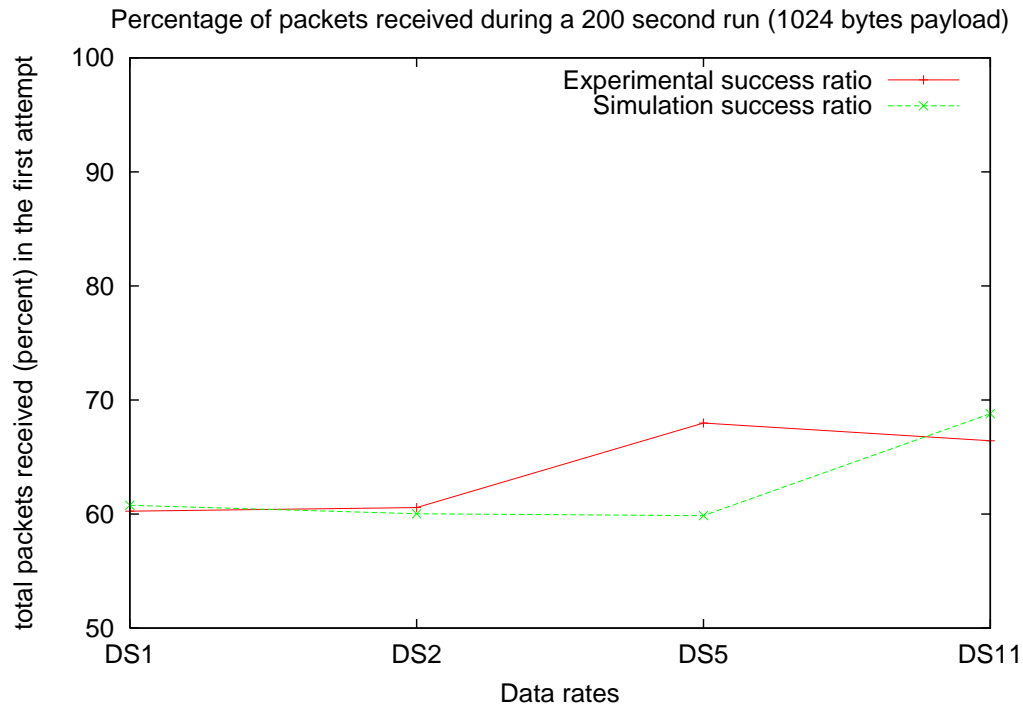
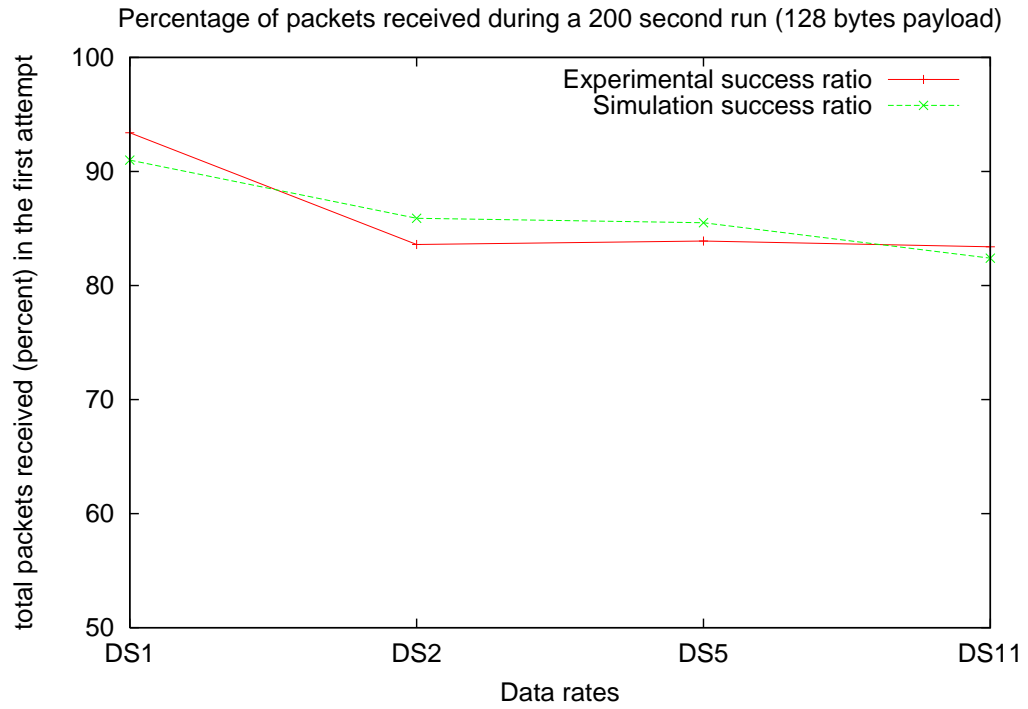


**Figure 16:** Received signal strength variations as a function of distance(simulated)

We present the results of 128 as well as 1024 byte transmissions in Figure 17. It can be inferred from the plots that the throughput measurements from the simulation model closely reflect the results of the field experiment. For the same mobility, the increase in payload size decreases the number of packets successfully transmitted in the first attempt by almost two-thirds. We believe that the differences in the model are due to experimental randomness that cannot be fully reproduced in the model. A much larger dataset might help bridge the graphs further.

## 4.6 Conclusions

This work is a study of an IEEE 802.11 wireless link behavior in a obstruction free outdoor environment. Our work mostly concentrates on low-speed mobile nodes. We define two sets of measurements to separate out the transmit and receive path characteristics. While the receive path characteristics are defined by the radio propagation losses, the transmit path characteristics are defined by the packet error rate when a



**Figure 17:** Percentage of frames successfully received during the simulation run in the first retry i.e retry count = 0



packet of a given size is transmitted at a given power level and modulation mechanism. We derived this packet error rate from the number of retries it takes to send a given packet. We infer that the decay of signal strength with distance is not strictly an exponential decay but a rather long tailed decay. Thus, we use an empirical distribution to model it. We model the transmit path as a packet error probability derived from the retry statistics. We incorporate these models in the Georgia Tech Network Simulator. To validate our models, we conduct a experiments with two nodes without any contention resolution (of the 802.11 MAC) involved. Our simulation results are a close match to those that were measured during the experiments. It must be noted that the radio propagation models have been constructed using only the RSSI values of successfully received frames.

### CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, we summarize our research work and directions for future research in this area.

#### *5.1 Thesis Summary*

This thesis focuses on the modeling and simulation aspects of the IEEE 802.11 WLAN specifications. It provides insight and experience into how one can do correct and realistic wireless network simulations. This is essential for deriving useful results from simulation experiments that are applicable to network engineering practice.

1. Wireless network simulations which resemble the real network behavior were incorporated into the GTNetS simulation environment. Both infrastructure as well as ad-hoc modes of network simulations are possible in GTNetS. A subset of the 802.11F based inter-access protocol and handoff mechanisms have also been implemented for extended basic service set operations.
2. The implementation of the IEEE 802.11 MAC protocol was studied in-depth in GloMoSim, ns-2 and GTNetS simulation environments. We designed two simple experiments for understanding the correctness and the sources of differences in these simulators. These experiments eliminate the effects of mobility, path-loss and modulation choices. To eliminate the effects of any stateful protocols at the higher layers of the protocol stack, we chose a simple stateless routing protocol to conduct our experiments. We categorize our study of the protocol models into two parts, namely, ideal behavior and behavior under channel contention.

- (a) The pristine code showed considerable variations in the ideal contention-less case. After some changes to the sources and making the default parameters uniform across all the simulators, the simulation results were identical.
- (b) In the experiment which had an extreme case of all nodes contending for the medium, we found that the variations in the network performance is considerable. Our experiments helped identify some of the differences in the protocol model implementations, but variations still exist.

Because of this behavior we conclude that different network simulators will produce different results. It is hard to claim that a particular simulation tool is correct or otherwise. However, it is important to understand that significant differences exist and one must take them into account when inferring from the simulation results. Just as there are differences in simulation models, it is likely that any real-world implementations of 802.11 MAC protocol will have differences. Depending on the hardware and platform, network performance results are likely to vary.

3. In chapter 4, we presented a method of deriving wireless simulation models from measurements. The basic method that we use consists of the following three steps :
  - (a) Conduct experiments and obtain measurements without any interference.
  - (b) Derive simulation models by analyzing the measurement data.
  - (c) Validate the models by implementing them in a network simulator and comparing these results with the measured ones.

Our measurements were carried out in a local outdoor field. We could concentrate only on modeling the link layer behavior by operating on channel 15 where

there is no contention due to cross-channel interference or due to other transmissions in the same channel. Our measurements show that while modulation-rate and packet size have a significant effect on the packet error rate, the effect of distance is much more pronounced in the outdoor environments. We then construct simulation models for radio propagation losses and packet error probability using the measurements made. These simulation models were implemented in the GTNetS to tune the behavior of the wireless PHY layer to accurately represent our experimental environment. Simulation experiments were conducted on a simple two node network to verify that the simulation results thus obtained actually match the real world experiments.

## 5.2 *Future Directions*

In this section, we describe the research topics which will generalize protocol modeling using measurements obtained from real-world experiments. Design of experiments and the analysis of the experimental data can provide useful insight in fine tuning empirical protocol models and expand their applicability and scope.

1. Even though we have quantitatively tried to characterize the impact of MAC modeling in various simulation environments, its effects on the performance on the higher layer protocol modeling has not been defined quantitatively. This problem requires that the same stateful protocol be implemented on all the simulation environments and its effects evaluated. These effects should then be characterized in an abstract way so that they may be applicable to various other protocol models and scenarios.
2. An area of considerable interest is to see how the real world MAC implementations differ from simulation environments. This area has not been explored because of the proprietary nature of MAC implementations. Moreover, most

MAC implementations have been offloaded to the micro-controller of the network peripherals resulting in almost no incentive for the manufacturers to open up their implementations. Recently, there has been a trend to offload the MAC computations over to the host machine to improve cost competitiveness. A number of softmac implementations have been developed for the open source operating systems [30, 53]. This opens a window for researchers to investigate ways of bridging the gap between simulation models of MAC protocol and the protocols implemented in practice.

3. Modeling the wireless link layer using stochastic models has an advantage that they can be applied to a large class of environments. On the other hand, measurements based empirical models are distinctly a characteristic of the environment in which the measurements have been made. Thus, even though the methodology we adopted provides a fairly accurate model, its applicability is limited. It is applicable only in outdoor environments which are similar to those where the experiments were conducted. Hence, there is a need to characterize environments with respect to their radio propagation properties. This is necessary so that measurement based models which have been developed with measurements or traces in a particular environment can be used for other environments also, which satisfy the same characterization.
4. For coarse simulation of IEEE 802.11 MAC/PHY based networks, simulation models that can abstract the whole of MAC/PHY layer are helpful. One way of achieving it is to have an estimate the amount of traffic in a particular location and derive the IEEE 802.11 MAC retries as a function of the traffic at a given location. This is not very easy at this time because, it is very hard to get all the 802.11 frames that the network peripherals' radio sees in the air. For reasons of efficiency and time criticality the network peripheral passes only a subset

of frames to the host (even in the monitor mode operation). The GNU Radio project provides for a software radio implementation that may provide all the necessary requirements for such an estimation to be possible. However, it is still in its infancy to be of much practical use for network protocol models.

5. The radio propagation models of wireless channels are based on the measurements of only successfully received packets. Thus, we are measuring signal strengths of only packets that have been deciphered by the radio's carrier tracker as a valid frame. However, this means that signals whose energy levels could not be tracked by the carrier sensing were ignored by our measurements. This effectively means that our average received signal strength is biased to being larger than the actual mean received signal strength. A more accurate propagation loss model can be done if the measurements were done using a sophisticated spectrum analyzer. Alternatively, if any peripheral could provide an interface into the digitized values of energy that the carrier tracking loop uses to sense the presence of a signal, it could provide for a more accurate propagation model. The prism chipset [10, 11] provides for such a value but it is exported only when a successful packet is received, thus its use is limited.
6. Our work does not parameterize the effects of the direction of motion and the direction in which the antenna is facing. However, it has been proposed in other measurements that the difference in direction of the orientation of the receiving and transmitting antenna have a finite effect on the propagation properties of the radio signals. Also, the characteristic of mobility of the communicating nodes has an effect on the propagation characteristic. For example, the Doppler effect can be more pronounced in relatively high speed mobile environments. The effect of nodes moving closer to each other or moving farther away

from each other has also not been studied. Incorporating all the above factors as parameters into a simulation model will make the model comprehensive. This should enable evaluation of network performance under a wider range of operational characteristics.

## REFERENCES

- [1] AGUAYO, D., BICKET, J., BISWAS, S., COUTO, D. S. J. D., and MORRIS, R., “MIT roofnet implementation.” <http://pdos.csail.mit.edu/roofnet/doku.php>, December 2006.
- [2] AGUAYO, D., BICKET, J., BISWAS, S., JUDD, G., and MORRIS, R., “Link-level measurements from an 802.11b mesh network,” in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 121–132, ACM Press, 2004.
- [3] ATHEROS. <http://www.atheros.com>, November 2006. Atheros Communications.
- [4] B, F., “A binary channel characterization using partitioned markov chains,” *IEEE Transactions on Information Theory*, pp. 221–227, April 1967.
- [5] BERTOLOTTI, S. and DUNAND, L., “Opnet 2.4: an environment for communication network modeling and simulation,” in *Proceedings of the European Simulation Symposium*, October 1993.
- [6] BICKET, J., AGUAYO, D., BISWAS, S., and MORRIS, R., “Architecture and evaluation of an unplanned 802.11b mesh network,” in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 31–42, ACM Press, 2005.
- [7] CAVIN, D., SASSON, Y., and SCHIPER, A., “On the accuracy of manet simulators,” in *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, (New York, NY, USA), pp. 38–43, ACM Press, 2002.
- [8] CERPA, A., WONG, J. L., KUANG, L., POTKONJAK, M., and ESTRIN, D., “Statistical model of lossy links in wireless sensor networks,” in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, (Piscataway, NJ, USA), p. 11, IEEE Press, 2005.
- [9] CLARK, M. V., LEUNG, K. K., MCNAIR, B., and KOSTIC, Z., “Outdoor iee 802.11 cellular networks: Radio link performance,” in *Proceedings of the IEEE International Conference on Communications, ICC 2002*, pp. 512–516, 2002.
- [10] CORPORATION, I., “ISL3873: Wireless LAN Integrated Medium Access Controller with Baseband Processor,” 2000.
- [11] CORPORATION, I., “HFA3863: Direct Sequence Spread Spectrum Baseband Processor with Rake Receiver and Equalizer,” December 2001.



- [12] COWIE, J., OGIELSKI, A., and NICOL, D., “The SSFNet network simulator.” Software on-line: <http://www.ssfnet.org/homePage.html>, 2002. Renesys Corporation.
- [13] ECKHARDT, D. and STEENKISTE, P., “Measurement and analysis of the error characteristics of an in-building wireless network,” in *SIGCOMM '96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 243–254, ACM Press, 1996.
- [14] ETSI, “GSM Radio Link Protocol for data and telematic services, Version 5,” *ETSI GSM TECHNICAL SPECIFICATIONS 04.22*, December 1998.
- [15] GAERTNER, G. and CAHILL, V., “Understanding link quality in 802.11 mobile ad hoc networks,” *IEEE Internet Computing*, vol. 8, no. 1, pp. 55–60, 2004.
- [16] GANESAN, D., KRISHNAMACHARI, B., WOO, A., CULLER, D., ESTRIN, D., , and WICKER, S., “Complex behavior at scale: An experimental study of low-power wireless sensor networks,” 2002. Technical Report USLA/CSD, TR 02-0013.
- [17] GARMIN. <http://www.garmin.com>, November 2006. Garmin International Inc.
- [18] GRAZIOSI, F. and SANTUCCI, F., “A general correlation model for shadow fading in mobile radio systems,” *IEEE Communications Letters*, vol. 6, pp. 102–104, March 2002.
- [19] GUDMUNDSON, M., “Correlation model for shadow fading in mobile radio systems,” *Electronic Letters*, vol. 27, pp. 2145–2146, November 1991.
- [20] HEIDEMANN, J., BULUSU, N., ELSON, J., INTANAGONWIWAT, C., CHAN LAN, K., XU, Y., YE, W., ESTRIN, D., and GOVINDAN, R., “Effects of detail in wireless network simulation,” in *Proceedings of the SCS Multiconference on Distributed Simulation*, (Phoenix, Arizona, USA), pp. 3–11, USC/Information Sciences Institute, Society for Computer Simulation, January 2001.
- [21] HEIDEMANN, J., MILLS, K., and KUMAR, S., “Expanding confidence in network simulation,” *IEEE Network Magazine*, vol. 15, pp. 58–63, Sept./Oct. 2001.
- [22] IEEE, “IEEE Standard 802-11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification,” *Institute of Electrical and Electronic Engineers*, 1997.
- [23] IEEE, “Ieee standard 802-3 carrier sense multiple access with collision detection(CSMA/CD) access method with physical layer specifications,” *Institute of Electrical and Electronic Engineers*, 2000.

- [24] IEEE, “IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation,” *Institute of Electrical and Electronic Engineers*, 2003.
- [25] JAAFER, T., REDDY, D., RILEY, G. F., and BLAIR, D., “Simulation-based routing protocol performance analysis - a case study,” in *WSC '06: Proceedings of the 2006 Winter Simulation Conference*, INFORMS Simulation Society, December 2003.
- [26] JOHANSSON, P., LARSSON, T., HEDMAN, N., MIELCZAREK, B., and DEGERMARK, M., “Scenario-based performance analysis of routing protocols for mobile ad-hoc networks,” in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 195–206, ACM Press, 1999.
- [27] JOHNSON, D. B. and MALTZ, D. A., “Dynamic source routing in ad hoc wireless networks,” *Mobile Computing*, pp. 153–181, 1996.
- [28] JUDD, G. and STEENKISTE, P., “Repeatable and realistic wireless experimentation through physical emulation,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 63–68, 2004.
- [29] KARP, B. N., “Geographic routing for wireless networks. phd thesis, harvard university, 2000.,” 2000.
- [30] KETRENOS, J. <http://ieee80211.sourceforge.net>, November 2006. An ieee80211 subsystem for Linux.
- [31] KONRAD, A., ZHAO, B. Y., JOSEPH, A. D., and LUDWIG, R., “A markov-based channel model algorithm for wireless networks,” *Wireless Networks*, vol. 9, no. 3, pp. 189–199, 2003.
- [32] KOTZ, D., NEWPORT, C., GRAY, R. S., LIU, J., YUAN, Y., and ELLIOTT, C., “Experimental evaluation of wireless simulation assumptions,” in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, (New York, NY, USA), pp. 78–82, ACM Press, 2004.
- [33] KURKOWSKI, S., CAMP, T., and COLAGROSSO, M., “Manet simulation studies: the incredibles,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.
- [34] LEE, W. C. Y., *Mobile Communications Engineering*. McGraw-Hill, 1982.
- [35] LEE, Y.-J. and RILEY, G. F., “Dynamic nix-vector routing for mobile ad hoc networks,” in *IEEE Wireless Communications and Networking Conference*, 2005.

- [36] LEMMON, J. J., “Wireless link statistical bit error model,” *NTIA Report, U.S. Department of Commerce, 02-394*, June 2002.
- [37] MCCANNE, S. and FLOYD, S., “The LBNL network simulator.” Software online: <http://www.isi.edu/nsnam>, 1997. Lawrence Berkeley Laboratory.
- [38] MULLEN, J. P., “Efficient models of fine-grain variations in signal strength,” in *OPNETWORK 2004, Washington, DC*, OPNET Technologies, 2004.
- [39] MULLEN, J. P., “A proposed method to estimate signal to noise plus interference levels in order to improve the performance of mobile ad hoc network routing protocols,” in *IPSI 2005, Amalfi, Italy*, International IPSI-2005 Amalfi Conference, February 2005.
- [40] NGUYEN, G. T., KATZ, R. H., NOBLE, B., and SATYANARAYANAN, M., “A trace-based approach for modeling wireless channel behavior,” in *Winter Simulation Conference*, pp. 597–604, 1996.
- [41] OULD-AHMED-VALL, E., RILEY, G. F., HECK, B. S., and REDDY, D., “Simulation of large-scale sensor networks using gtsnets,” in *MASCOTS '05: Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, (Washington, DC, USA), pp. 211–218, IEEE Computer Society, 2005.
- [42] PAWLIKOWSKI, K., JEONG, J., and LEE, R., “On credibility of simulation studies of telecommunication networks,” *IEEE Communications Magazine*, vol. 40, pp. 132–139, January 2002.
- [43] PERKINS, C. E., ROYER, E. M., DAS, S. R., and MARINA, M. K., “Performance comparison of two on-demand routing protocols for ad hoc networks,” *IEEE Personal Communications*, vol. 8, pp. 16–28, Feb 2001.
- [44] PERKINS, C. E. and ROYER, E. M., “Ad hoc on-demand distance vector routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, Feb 1999.
- [45] POSTEL, J., “Internet RFC793: Transmission control protocol.” Network Working Group, Sep 1981.
- [46] RALINK. <http://www.atheros.com>, November 2006. Ralink Technology Corp.
- [47] RAPPAPORT, T. S., *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, New Jersey, 1996.
- [48] RILEY, G., “Large-scale network simulation with GTNetS,” in *Proceedings of the 2003 Winter Simulation Conference (WSC'03)*, pp. 676–684, Dec 2003.
- [49] RILEY, G. F., “The Georgia Tech Network Simulator,” in *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pp. 5–12, ACM Press, 2003.

- [50] RILEY, G. F., “Large-scale network simulations with gtnets,” in *WSC '03: Proceedings of the 2003 Winter Simulation Conference*, INFORMS Simulation Society, December 2003.
- [51] SCALABLE NETWORK TECHNOLOGIES, I., “QualNet Simulator,” October 2006. <http://www.qualnet.com>.
- [52] SHIH, E., CHO, S.-H., ICKES, N., MIN, R., SINHA, A., WANG, A., and CHANDRAKASAN, A., “Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks,” in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 272–287, ACM Press, 2001.
- [53] SOURCE, D. O. <http://www.devicescape.com/opensource/>, November 2006. Advance Datapath Driver.
- [54] TAKAI, M., BAGRODIA, R., TANG, K., and GERLA, M., “Efficient wireless network simulations with detailed propagation models,” *Wireless Networks*, vol. 7, no. 3, pp. 297–305, 2001.
- [55] TAKAI, M., MARTIN, J., and BAGRODIA, R., “Effects of wireless physical layer modeling in mobile ad hoc networks,” in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 87–94, ACM Press, 2001.
- [56] YEUNG, G., TAKAI, M., and BAGRODIA, R., “Effects of detailed ofdm modeling in network simulation,” *Simulation*, vol. 81, no. 4, pp. 241–253, 2005.
- [57] YEUNG, G., TAKAI, M., BAGRODIA, R., MEHRNIA, A., and DANESHRADE, B., “Detailed ofdm modeling in network simulation of mobile ad hoc networks,” *pads*, vol. 00, pp. 26–34, 2004.
- [58] ZENG, X., BAGRODIA, R., and GERLA, M., “GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks,” in *Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, May 1998.
- [59] ZHAO, J. and GOVINDAN, R., “Understanding packet delivery performance in dense wireless sensor networks,” in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 1–13, ACM Press, 2003.
- [60] ZHOU, G., HE, T., KRISHNAMURTHY, S., and STANKOVIC, J. A., “Models and solutions for radio irregularity in wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 221–262, 2006.

## VITA

Dheeraj Reddy was born in Chittoor, India on December 8th 1976. He studied in half a dozen schools before completing his AISSCE from Kendriya Vidyalaya, Uppal Hyderabad in 1993. He received his Bachelors in Technology (B.Tech) in Electrical Engineering from Sri Venkateswara University in May 1997 and Masters in Technology (M. Tech) in Electrical Engineering in January 1999 from Indian Institute of Technology, Madras. His undergraduate studies were funded by an armed forces scholarship and graduate study was funded by the University Grants Commission (UGC). He received a gold medal for his academic achievements from Sri Venkateswara University. He worked for Tata Consultancy Services and Nortel Networks for the next three years in telecommunication signalling protocol development and deployment. In 2002, he joined the doctoral program at the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta.

His research interests lie in wireless networks, network simulation, measurements and protocol modeling. His non-academic interests lie in abstract philosophy, music, trivia, badminton, racquetball, cricket and hiking. Apart from his academic pursuits, he likes to work on the Linux and the BSD operating systems especially the networking and the memory management subsystems.