UNIVERSITY OF CALIFORNIA

Los Angeles

Detailed OFDM Modeling in Network Simulation of Mobile Ad Hoc Networks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Ka Ki Yeung

2003

The thesis of Ka Ki Yeung is approved.

Mario Gerla

Babak Daneshrad

Rajive Bagrodia, Committee Chair

University of California, Los Angeles

2003

To my father, mother, sister, and Cathy.

Table of Contents

List of Figures

List of Tables

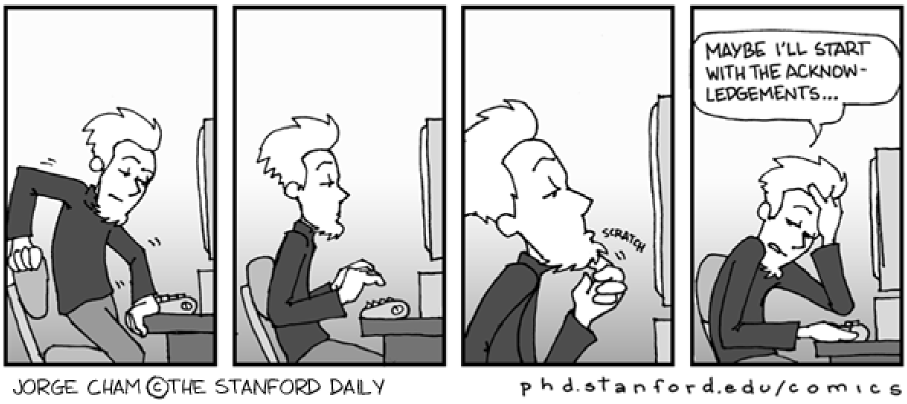
List of Equations

Acknowledgments

I would like to thank my advisor, Dr. Rajive Bagrodia, for helping me through this research with his guidance, wisdom, and knowledge. I am very grateful to Dr. Mineo Takai for all of his help, encouraging words, and constructive criticisms of this research path and other ideas. I gratefully acknowledge Dr. Babak Daneshrad and Mr. Alireza Mehrnia for providing our research group with the detailed OFDM simulator, without which this thesis would have never materialized. I would also like to thank Mrs. Junlan Ji and Mr. Zhengrong Ji for spending many late nights and weekends with me in the PCL Lab.

In addition, I would like to express my deepest gratitude to Dr. Allen Moshfegh for his visionary concepts and generous research funding.

This work was supported in part by the Office of Naval Research through the MINUTEMAN project under contract number N00014-01-C-0016.



“Piled Higher and Deeper” by Jorge Cham (www.phdcomics.com)

ABSTRACT OF THE THESIS

Detailed OFDM Modeling in Network Simulation of Mobile Ad Hoc Networks

by

Ka Ki Yeung

Master of Science in Computer Science
University of California, Los Angeles, 2003
Professor Rajive Bagrodia, Chair

Typical studies of mobile ad hoc networks (MANET) use network simulation for the performance evaluation of protocols running at high layers. The need for developing highly detailed device models for accurate network simulation is clear; detailed models for the device and the wireless channel are crucial for prediction of higher layer network protocol performance. However, current network simulators substitute accurate physical layer models with generic abstract models for simplicity and speed. One approach to bridge this gap is by integrating a MATLAB Simulink OFDM radio and channel model into QualNet, a scalable packet-level simulator. This technique effectively captures the effect of the radio propagation and device while still maintaining reasonable simulation execution time. Also, the results depict that detailed modeling is necessary in studying the performance of higher-level protocols when compared to an abstract physical layer model. For example, the performance of the Auto Rate Fallback algorithm sharply decreases as the network load increases and this effect is more gradual when the transmission data rate is fixed. The detailed model incorporates the effects of different combinations of physical layer variables, e.g., path loss, shadowing, multipath, Doppler fading, and delay spread for each individual transmission in the simulation. Since traditional abstract modeling methods oversimplify the aforementioned variables, this lead to erroneous higher layer results when compared against the detailed model where these effects are modeled and observable. Model detail fidelity and simulation time tradeoff analysis are studied and compared.

Chapter 1 Introduction

Network simulation is commonly used for the evaluation of wireless network protocols. Using a discrete event simulation model, the network simulator models network activities on a packet-by-packet basis, on time step of 10s of microseconds, and includes a model for each layer of the entire protocol stack. Abstract models can be acceptable if they do not significantly compromise the accuracy of the simulation results. However, these abstract models are often in place because detailed models are too difficult to implement or run efficiently.

Studies on physical layer techniques and their performance under varying channel conditions often utilize highly specialized mathematical tools such as MATLAB, Maple, and Mathematica [Matlab][Maple][Mathematica]. These software packages provide a rich set of built-in libraries and standard building blocks for use in rapid development. Channel characteristics, modulation and demodulation techniques are modeled, simulated, and studied under various parameterizations. It should be noted that this highly detailed technique of modulating and demodulating every bit and simulating the transfer of the bit across the wireless channel comes at a high processing cost and execution time.

At first, the tradeoff between abstract and detailed simulation methods is not so obvious. An abstract model may replace a detailed model if such a model does not produce inaccurate results. Such an example would be the recently proposed fluid-based analytical model to determine queue size for large flow networks [Misra00][Yung01]. In other cases, detailed simulation models are necessary to accurately predict network performance. This is especially true for the physical layer in wireless networks where slight inaccuracy may become critically magnified in higher layer protocols. In [Takai00][Takai01], the authors present credible reasons for considering the physical layer as a necessity to fully determine ad hoc network routing performance. Even with very strong evidence at hand to do against so, current network simulators apply abstract modeling methods to simulate the propagation layer and radio device characteristics. All in all, current network simulation implementation simply neglects to accurately model the physical layer. It instead favors the abstracted, simple model, for the sake of execution speed and efficiency.

There is significant information to be gained in detailed simulation of the physical layer. In a wireless medium where channel condition changes frequently, nanosecond step simulation of devices provides valuable insights that otherwise would be lost in abstract modeling. This thesis presents a strategy to develop appropriate interfaces between a packet-level simulator, QualNet [QualNet], and a MATLAB model for OFDM (Orthogonal Frequency Division Multiplexing) radio and channels. It is demonstrated that detailed simulation of the physical layer significantly affects the performance prediction of higher layer protocols. Specifically, it is shown that the number of MAC (Medium-Access-Control) retransmissions and MAC packet drops may significantly differ when the abstract and detailed models are compared at various data rates. This, in turn, causes a varying degree of effects on the packet delivery ratio.

The interfaces defined between QualNet and the OFDM model provides a clean, modular, and scalable multi-granular simulation paradigm. The technique defined in future chapters is novel in that the integrated system can simulate on different levels of granularity in response to user requirements. That is, the proposed detailed model can simulate every bit of the network, a subset of the bits of the network using a robust cache model, or bypass the detailed model altogether and use a basic, yet speedy abstract model. Also, the integrated simulator, like QualNet, is also able to run on parallel architectures. The implication of this is that a user can change the underlying physical layer model and depending on the time versus accuracy requirement, analyze the effects of the change on simulations of varying granularity and accuracy.

The remainder of the thesis is organized as follows. Chapter 2 gives a general overview of related works. Chapter 3 discusses QualNet, the discrete event network simulator and the orthogonal frequency division multiplexing (OFDM) modulation. Chapter 4 discusses the IEEE 802.11a MAC and PHY protocol, and the modeling and verification of it in QualNet. Chapter 5 describes the details associated with the integration of the OFDM simulator in QualNet. Chapter 6 explains the simulation studies, results of the integrated model versus the abstract model, and simulation execution time performance. Finally, conclusions are discussed in Chapter 7.

Chapter 2 Related Work

This chapter describes simulation models that have been proposed in the past. In particular, multi-granular, multi-paradigm, and multi-simulator simulation systems are discussed.

2.1 Ptolemy and Ptolemy II

The Ptolemy [Buck94][Chang95] project studies heterogeneous modeling, simulation, and design of concurrent embedded systems, particularly those that mix technologies (i.e. analog and digital electronics, hardware and software, and electronics and mechanical devices). The project concentrates on systems that are complex in the sense that they mix widely different operations, such as signal processing, feedback control, sequential decision making, and user interfaces. The idea is to use a heterogeneous software environment to develop heterogeneous designs. The interaction between different modules and layers in the software environment is managed through object-oriented principles. Ptolemy has been used for a broad range of applications including telecommunications, parallel processing, network design, radio astronomy, real time systems, and hardware/software co-design [Lao94][Pino95]. The co-simulation architecture allows hardware/software co-simulation in ways that give designers full system feedback on their design choices. These design choices include hardware/software partitioning, CPU selection, and scheduler selection.

Ptolemy II [Lee02][Liu03] proposes an actor-oriented design methodology that tackles complex control system issues by separating the data-centric computational components (actors) and the control-flow-centric scheduling and activation mechanisms (frameworks). Semantically different frameworks are composed hierarchically to manage heterogeneous models and achieve actor and framework reuse.

When using Ptolemy II as the design environment, some of the widely used models of computation for control system design — continuous time, discrete event, synchronous dataflow, timed multitasking, and finite state machine — are implemented as responsible frameworks [Lee01][Eker03]. Closed-loop control performance can be simulated and quickly fed back to designers at each step and gradual model enrichment can bring the simulation closer to reality.

The Ptolemy approach is similar to the approach used in this thesis. Our method of integrating heterogeneous models to develop heterogeneous designs is applied to the network simulation domain to realize the innovations that are being made on different layers of the network stack whereas the Ptolemy project concentrates on hardware/software co-design.

2.2 MIC and MILAN

Model-Integrated Computing (MIC) [Davis02][Karsai93] was also developed to model embedded software systems. MIC provides rich, domain specific modeling environments combined with model analysis and model-based program synthesis technology. The key element is the extension of the scope and usage of models such that they form the "backbone" of a model-integrated system development process. By integrating multiple-view models to capture the information relevant to the system to be developed, models can explicitly represent the designer's understanding of the entire system, including the information processing architecture, the physical architecture, and the environment it operates in. These models act as a repository of information that is needed for analyzing and generating the system. MIC allows designers to create domain specific models of systems, validate these models, and perform various computational transformations on the models.

Model based Integrated simuLAtioN (MILAN) [Agrawal01][Ledeczi03] is built using MIC technology to facilitate multi-granular simulation and different abstractions into a unified framework. Using the modeled applications, resources, and constraints, MILAN is able to perform several activities included design space exploration, system generation, and simulation configuration. Functional simulations verify the functionality of the application. The integrated high-level simulator provides a rapid, reasonably accurate estimate of the different performance criteria of the system. Lower-level power and performance simulation are also supported to simulate some components at higher fidelity. While these can be very accurate, their slow speed may prevent the simulation of the while system. The goal of the framework is to get to a handful of candidate solutions that satisfy all of the input constraints. These candidate solutions are then subjected to further and more detailed analysis.

MIC and MILAN are similar to the MAYA approach discussed later in the next section. Similar to Ptolemy, MIC and MILAN focus on embedded software systems, MAYA and the contributions of this thesis concentrate on scalable and accurate network simulation.

2.3 MAYA

Maya [Zhou03] is a multi-paradigm, multi-resolution, scalable and extensible network modeling framework for emulating distributed applications. The goal is to study the tradeoffs between speed and accuracy of multiple modeling approaches, as a function of different types and scales of networks, protocols, traffic and application types, and metrics. A combination of analytical models, packet level parallel simulation, detailed bit level simulation, and emulation is used to model different granularities and model paradigms.

An example of the Maya architecture is the integration of a network analytical model into a packet level simulator [Yung01]. The fluid flow analytical model [Misra00] has been shown to be able to capture the dynamics of TCP flows with RED as the network AQM policy, and can scale well to a large number of flows. Specifically, the TCP traffic is described by a set of Stochastic Differential Equations (SDEs). The fluid flow model is incorporated into QualNet and the resulting mixed mode simulator shows good validation with the results obtained from pure packet-level simulation.

The Maya architecture is shown in Figure 2.1. In [Zhou03], this extensible network modeling framework is used to emulate a distributed multimedia application. A combination of discrete event simulation (QualNet), analytical model (fluid-flow), and physical network emulation (computer interfaces) are tied together to form a real time heterogeneous modeling paradigm. The results show that this modeling paradigm is able to keep up with large MANET simulations with real time video application requirements.

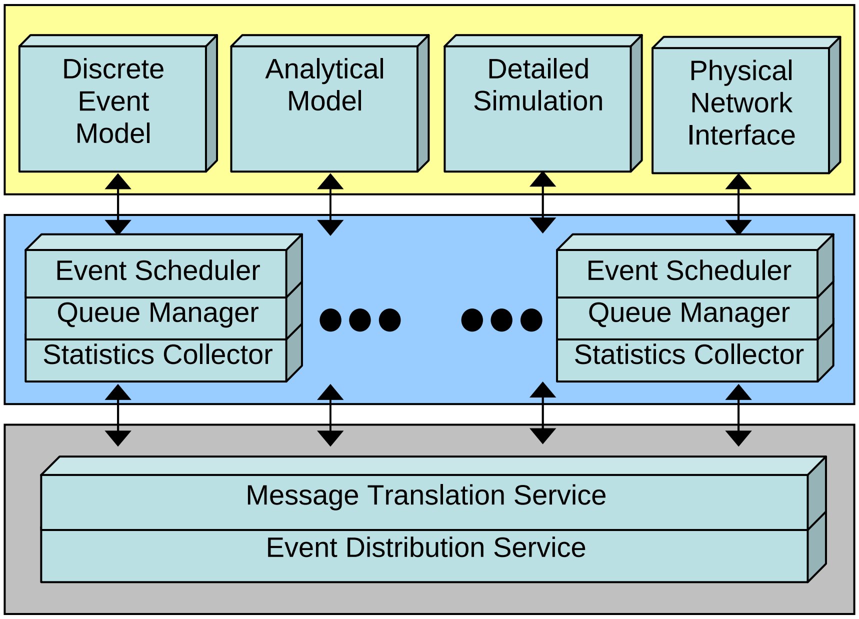


Figure 2.1: Maya Architecture

One can see the contributions of this thesis as an improvement to the MAYA architecture. The integration of an OFDM model into a network simulator is the first nanosecond time step framework used for network systems study. Similar to the MAYA philosophy, we both aim to accurately characterize the network and derive error bounds where tradeoffs for speed instead of complete accuracy are needed.

2.4 Multi-Simulator Simulation with Georgia Tech Backplane

Georgia Tech developed a backplane that enabled the user to bring multiple network simulators together and harness their models in a single experiment [Riley01]. By bridging multiple heterogeneous network simulators, the backplane provides users with the ability to take advantage of the strength and capabilities of different simulators. The simulation engine exchanges meaningful event messages with other simulators, even when they do not share a common event message format. [Xu01] presents a split protocol stack methodology for network simulation that allows network researchers to run different layers of the network at different simulators. The integration detailed an architecture where multiple simulators are operating at different levels of fidelity in a single experiment.

The Georgia Tech Backplane gels network simulators of different strengths together to form one integrated network simulation platform. Likewise, the contributions of this thesis marry simulators of dramatically different time granularity and fidelity to investigate cross layer effects of different network layer technologies.

Chapter 3 The QualNet Simulator and OFDM Modulation

In this chapter, the QualNet [QualNet] network simulator and OFDM modulation are described. Just enough details are provided to the reader to give a general overview of each technology and to let the reader appreciate the challenges of integrating an OFDM simulator for network simulation.

3.1 The QualNet Packet-Level Simulator

QualNet is the next generation of the scalable GloMoSim (Global Mobile Information Systems Simulator) [GloMoSim] simulator. GloMoSim was designed to simulate large-scale wireless networks with thousands of mobile nodes, each of which may have different communication capabilities via multi-hop ground, aircraft, and satellite media [Xiang98]. QualNet has extended GloMoSim's capabilities to wired networks as well as mixed wired and wireless networks. Like its predecessor, QualNet uses the parallel simulation kernel provided by the PARSEC discrete-event simulation language [Lokesh99]. As a result, QualNet is among the few simulators for wireless and wired networks that have been implemented on sequential and parallel architectures. Example of parallel platforms include an 8-processor DELL™ PowerEdge 6100 running Windows NT®, a 24-processor Sun® Enterprise 5000 running Solaris™, a 28-processor SGI™ 2000 running IRIX®, and a dual-processor Intel Xeon® machine running Redhat™ 8.

This work focuses primarily on QualNet's capability for simulating wireless networks on sequential architectures. Other commonly used discrete-event network simulators include ns-2 [ns2] and OPNET [Opnet].

QualNet includes detailed models of commonly used protocols at each of the primary layers of the protocol stack. These ranges from commonly used applications like file transfer (ftp) and web browsing (http) to transport, routing, and MAC layer protocols. In each case, commonly used protocols in both wired and wireless networks have been modeled. For instance, routing protocols like OSPF and RIP that are common in wired networks have been modeled, as well as AODV and DSR for wireless networks. Protocols for GSM cellular and WiFi networks have also been developed. The current list of protocol models that are available in QualNet version 3.6 are listed in Table 3.1.

• Application	ftp, telnet, cbr, Tcplib, http, MODSAF, synthetic traffic generators, self-similar traffic with long range dependency
• Transport	TCP (FreeBSD, Reno, Tahoe, New Reno, Westwood), UDP, RSVP
• Routing	Bellman-Ford, OSPFv2, RIPv2, Flooding, Fisheye, DSR, LAR1, AODV, ODMRP, STAR, DVMRP, MOSPF, PIM-DM, QOSPF, BGPv4
• MAC	CSMA, MACA, IEEE 802.11 DCF, GSM
• Physical	Point-point link, wired bus, satellite, IEEE 802.11 radio
• Propagation	Path loss (free space, 2-ray ground reflection, trace, ITM , TIREM), fading (Rayleigh, Ricean), shadowing
• Mobility	Random waypoint, group mobility, trace files

Table 3.1: QualNet model library protocols

QualNet defines simple APIs between neighboring layers to enhance modular composition of protocol models developed at different layers by different designers. A sample listing and interaction view of the protocol layers are shown in Figure 3.1. The APIs are kept as close as possible to the operational protocol stack, such that even operational code is easily integrated into QualNet with this layered design. The integration capability has already been demonstrated at the transport layer in QualNet by extracting the TCP Lite model from the protocol code distributed with the FreeBSD operating system. The only restrictions made in the QualNet APIs are that network nodes can communicate with other nodes only through the lowest layer, and models at other layers cannot directly access data from other network nodes. A number of statistical metrics at each layer of the protocol stack are collected automatically by the simulator and can subsequently be used by the analyst to understand the application level performance metrics.

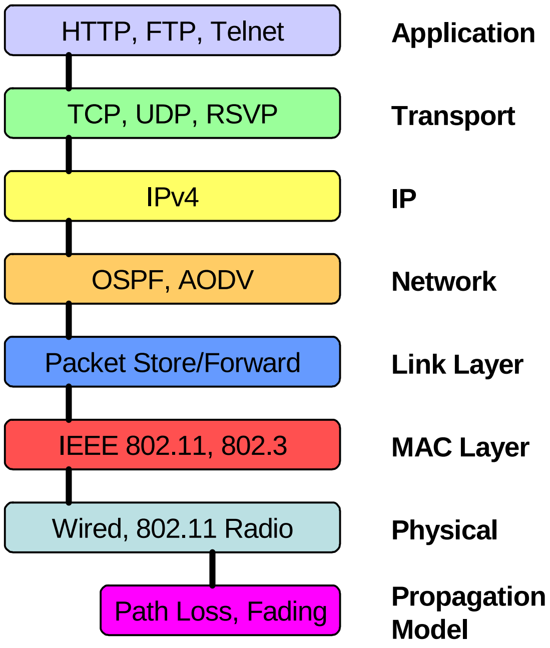


Figure 3.1: Protocol stack layers

3.2 Orthogonal Frequency Division Multiplexing (OFDM)

OFDM is a modulation scheme that converts a wideband signal into a series of independent narrowband signal placed side-by-side in the frequency domain. The main benefit of OFDM is that the subcarriers in the frequency band can actually overlap one-another. The data to be transmitted is split into *n* parallel data streams, each of which modulates a subcarrier as shown in Figure 3.2. Due to implementation complexity, OFDM applications have been scarce until recently with the advances in DSP technology. The IEEE 802.11 working group adopted OFDM technology in IEEE 802.11a and IEEE 802.11g wireless networks. OFDM modulation is also used in DVB (Digital Video Broadcasting) and DSL (Digital Subscriber Line).

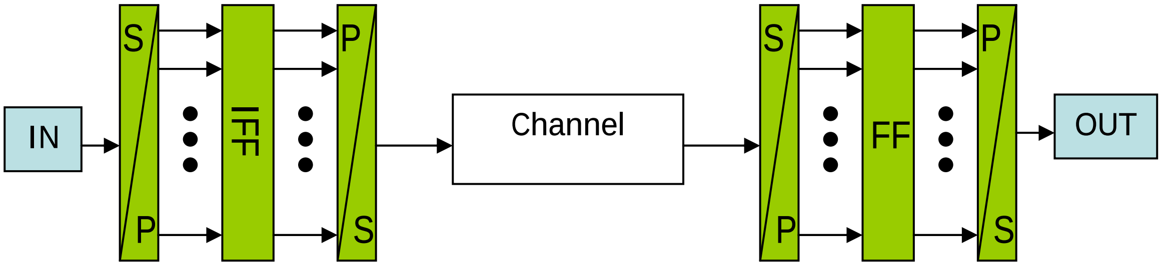


Figure 3.2: Abstract view of OFDM sender and receiver modulation

OFDM can be thought of as a combination of multi-carrier modulation (MMC) and frequency shift keying modulation (FSK). Orthogonality amongst the carriers is achieved by separating the carriers by an integer multiple of the inverse of symbol duration of the parallel bit stream. The entire allocated channel is occupied through the aggregated sum of the narrow orthogonal sub-bands. In order for the carriers to not interfere with each other, the spectral peak of each carrier must coincide with the zero crossing of all the other carriers as depicted in Figure 3.3.

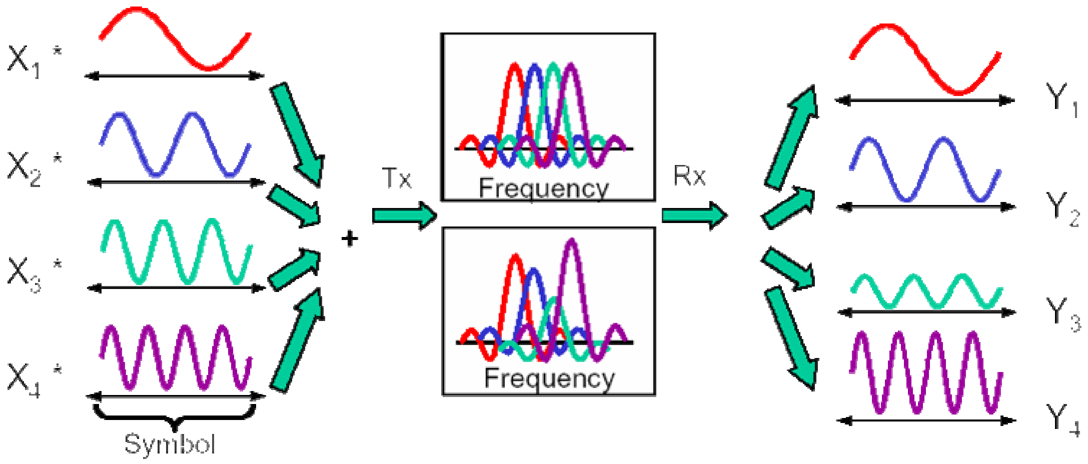


Figure 3.3: Overlapping orthogonal subcarriers when viewed from the frequency domain

OFDM communication systems naturally alleviate the problem of multipath propagation with its low data rate per subcarrier. The data rate per subcarrier is only a fraction of conventional single carrier systems having the same throughput. This is one of the biggest advantages of OFDM modulation. Pilot tones are often used in OFDM systems for channel estimation refinement. In IEEE 802.11a, four of the 52 subcarriers are used as pilot tones for correcting residual frequency offset errors that tend to accumulate over symbols. Interested reader should refer to [Keller00][Nee00] for more details.

The integration of a detailed OFDM simulator into QualNet poses numerous technical difficulties. The interfaces between the simulators, the time scale and fidelity differences, and execution speed consideration must be carefully evaluated. A brief overview of both technologies was described to let the reader appreciate these divergent disciplines. One can see that this integration is crucial for accurate wireless network performance prediction. An accurate physical layer model can be modeled with dynamic online simulation that includes the effect of path loss, shadowing, multipath, Doppler fading, and delay spread. Together with higher network layer simulation, the integration allows physical, MAC, routing, transport, and application protocol designers to see the effects of their designs as a whole on a full scale system level.

Chapter 4 Implementation and Verification of IEEE 802.11a

This section describes the IEEE 802.11a MAC and PHY protocol and how the MAC and an abstract OFDM PHY is modeled and verified in QualNet. The IEEE 802.11a MAC and PHY were not yet implemented as of QualNet version 3.5. For the purpose of this study, the IEEE 802.11a model was implemented to accurately compare detailed and abstract simulation models of the OFDM radio device at varying channel conditions and data rates.

4.1 IEEE 802.11 MAC Distributed Coordination Function (DCF)

The IEEE 802.11 DCF MAC is primarily responsible for two things: maintaining the NAV (network allocation vector), and to request for medium access. Upon detecting channel transmission, a node sets its NAV timer to the maximum of the current NAV or the duration of the transmission specified in the transmitting packet header. Channel medium reservation is implemented through the RTS/CTS (Request-To-Send/Clear-To-Send) [Bharghavan94] message exchange with CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). A detailed two node transmission sequence is shown in Figure 4.1.

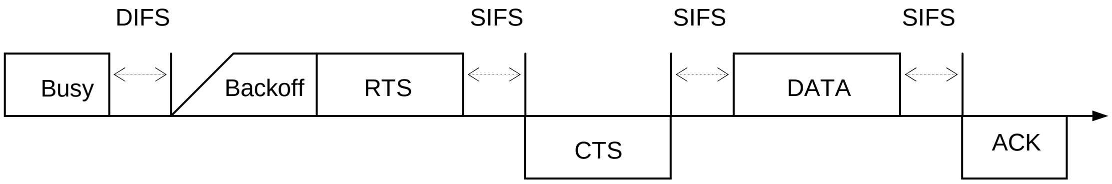


Figure 4.1: IEEE 802.11 MAC RTS/CTS transmission sequence

The key idea of DCF is to allow each station a fair chance of accessing the medium without a central coordinator. The medium reservation protocol proceeds as follows: a source transmits an RTS frame requesting for reservation of the medium for the transmission duration of the sequence of frames (CTS, Data, and ACK) plus three SIFS (Short InterFrame Space) time. The destination, upon hearing the RTS frame from the source, responses by transmitting a CTS frame including the previously announced medium reservation duration minus its own CTS frame transmission time and a SIFS time. Now, both the sender and the receiver have notified each other that the transmission is about to take place. Nodes that overhear these frame exchanges do not transmit frames until the transmission reservation time expires. Used to solve the "hidden" terminal problem, the RTS/CTS exchange is an optional part of IEEE 802.11 DCF (Distributed Coordinate Function) but is typically used in MANETs [MANET]. The basic protocol sequence is shown in Figure 4.2 and the relevant timing parameters for IEEE 802.11a are shown in Table 4.1. Interested readers should examine [ieee5ghz99][ieee80211_99][Fullmer97].

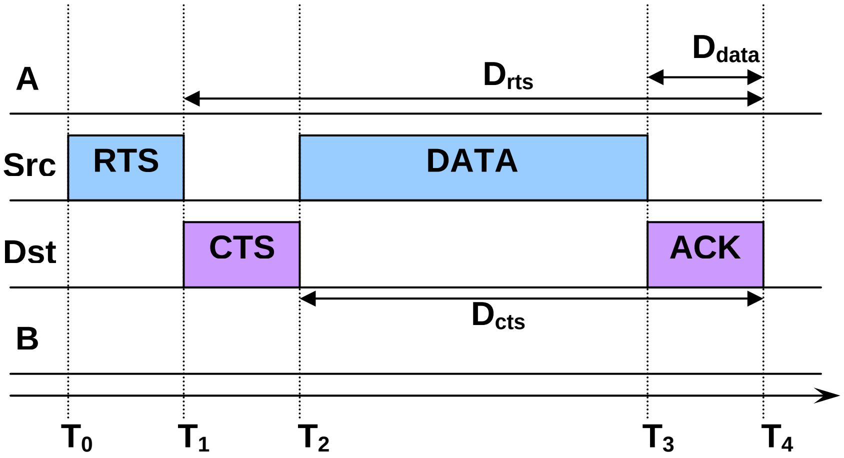


Figure 4.2: IEEE 802.11 MAC NAV timing

Characteristics	**Duration (μs)**	**Comments**
aSIFSTime	16	SIFS Time
aDIFSTime	34	DIFS Time
aEIFSTime	56	EIFS Time
tPLCPPreamble	16	Preamble Duration
tPLCPHeader	4	Signal Duration
tSymbol	4	Symbol Duration

Table 4.1: IEEE 802.11a timing characteristics

4.2 IEEE 802.11a OFDM PHY

The PHY layer is responsible for pre-pending physical preamble to the MAC frame, guard time and pilot tone insertion, and modulating and coding the data packet to the desired data rate. This physical preamble is used to allow the receiver to detect start of packet transmission and to access the channel. The use of pilot subcarriers to correct frequency offset implicitly assumes that channel variations during packet transmission are negligible. When the delay spread is shorter than the guard time and the coherence time of the channel is longer than the transmission duration, the OFDM receiver has a much higher chance of correctly demodulating the perceived signal.

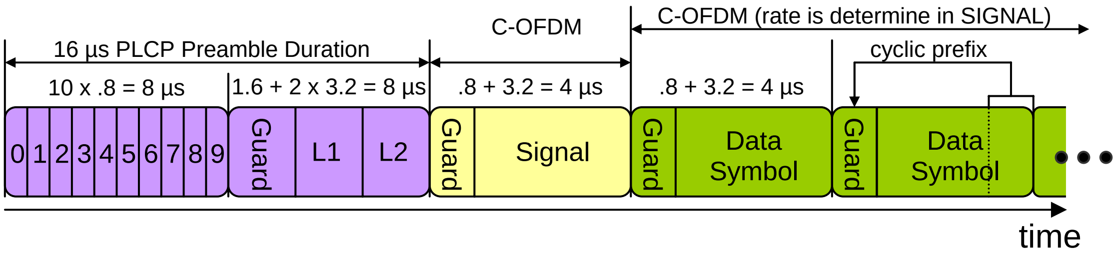


Figure 4.3: IEEE 802.11a preamble and start of data

As shown in Figure 4.3, the PLCP (Physical Layer Convergence Protocol) preamble begins with 10 short training symbols of 0.8 µs each followed by two long pulses of 4.0 µs each. This preamble sequence allows the receiver to first detect the incoming packet followed by a coarse and fine channel estimation algorithm. The first seven short training symbols are use for AGC (Antenna Gain Control), packet detection, and diversity selection. The remaining three short training symbols are used for coarse frequency offset estimation and symbol timing. The next two OFDM symbols contain long training pulses used for channel estimation and fine frequency offset estimation. The signal field, always encoded at the lowest data rate, tells the receiver the encoded data rate and length of the payload. Finally, the payload data is transmitted in an integral number of data symbols modulated and encoded at the scheme specified by the MAC. Each data symbol is 4.0 μs long.

The first ten short training symbols correspond to the top sparse preamble in Figure 4.4. Individual subcarrier fading is combated with convolutional coding, bit scrambling, and interleaving techniques. Each of the subcarriers is spaced 312.5 kHz apart and a guard time (cyclic prefix) of 800 *ns* is added to each symbol. A combination of different modulation and coding schemes are used to give IEEE 802.11a the wealth of data rates. Interested reader should refer to [Keller00][Nee00][Rappaport95][Terry01] for more details.

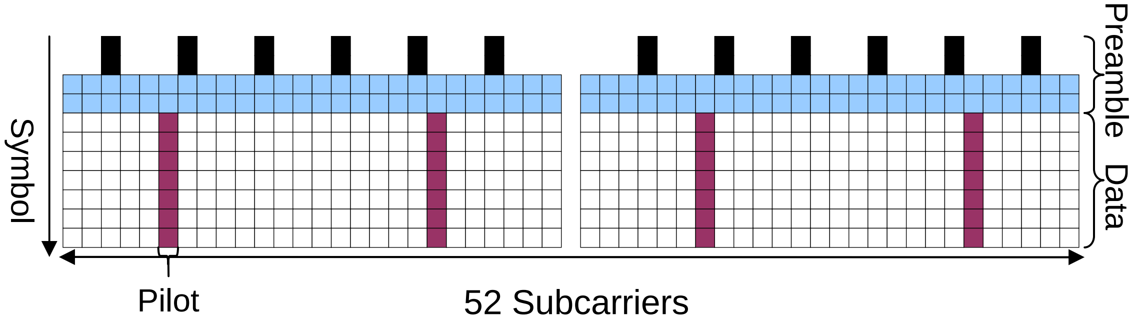


Figure 4.4: IEEE 802.11a PHY frequency view

4.3 IEEE 802.11a PHY Implementation

A great deal of attention has been paid to accurately capture the effects of the physical radio layer characteristics described earlier. The basic idea of our effort is to accurately model physical preamble timings and correctly calculate the transmission duration at different data rates. Furthermore, BER (Bit Error Rate) versus SINR (Signal to Interference and Noise Ratio) performance tables for each of IEEE 802.11a data rates must be derived. Meticulous attention was paid to model SIFS, DIFS (Distributed InterFrame Space), and EIFS (Extended InterFrame Space) time spacing as well (see Table 4.1 for values).

IEEE 802.11a, operating in the 5 GHz band, specifies data rates ranging from 6 to 54 Mbps. Table 4.2 contains a listing of the eight specified PHY data rates. Four different modulation schemes are used: BPSK, 4-QAM, 16-QAM, and 64-QAM. Each higher performing modulation scheme requires better channel condition for accurate transmission. These modulation schemes are coupled with the various forward error correction convolutional encoding schemes to give a multitude of *N*umber of *d*ata *b*its *p*er *s*ymbol (*Ndbps*) performance.

Data Rate (*Mbps*)	**Modulation**	**Coding Rate**	***Ndbps***	**1472 byte Transfer Duration (*μs*)**
6	BPSK	½	24	2012
9	BPSK	¾	36	1344
12	4-QAM	½	48	1008
18	4-QAM	¾	72	672
24	16-QAM	½	96	504
36	16-QAM	¾	144	336
48	64-QAM	⅔	192	252
54	64-QAM	¾	216	224

Table 4.2: PHY modes of IEEE 802.11a

The BER versus SINR performance curve was generated using the OFDM simulator from [Terry01]. The performance curve was made by running the OFDM model and statistically generating the results over a number of trial runs at the specified modulation, channel, and coding rate. This was provided by [SNT], the maker of QualNet. Figure 4.5 and Figure 4.6 shows the BER versus SINR curve at all supported IEEE 802.11a data rates. The BER signal reception model looks up the BER for a given SINR and probabilistically determines whether the node receives a frame with or without errors using Equation 1, where *numBits* is the number of bits simulated for that particular BER. It evaluates each frame segment, in which the interference from other transmissions are constant. If the error probability is greater then the generated random number in QualNet for the evaluation, that packet is presumed to have errored and the node's radio unlocks on signal reception; the signal becomes noise. The SINR value is derived using Equation

2 where *P* is the reception power, *T* – the temperature of the environment, *F* – the noise factor of the radio, *B* – the data bandwidth, and *k* – Boltzmann's constant. All signals are assumed to conform to Gaussian noise characteristics.

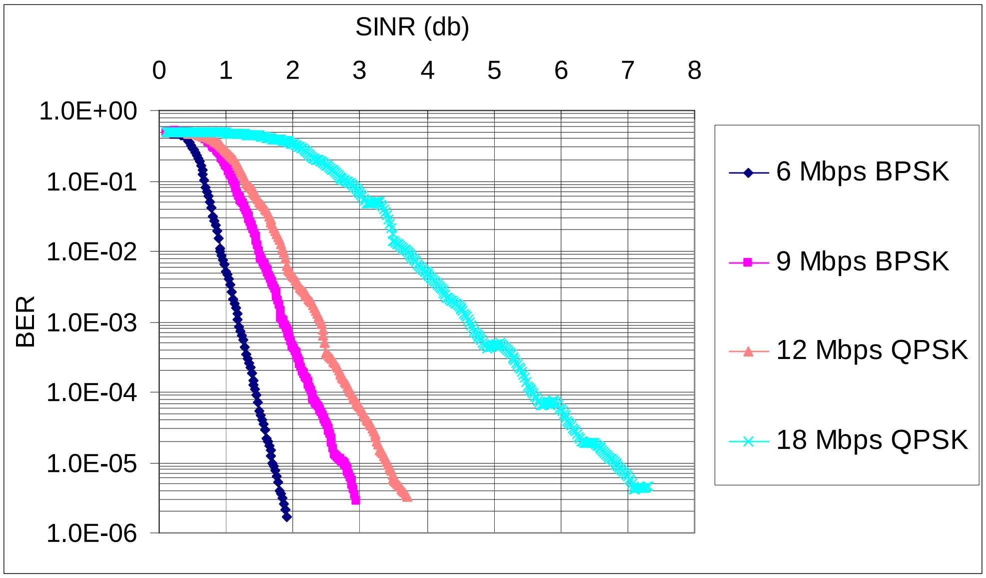


Figure 4.5: BER vs. SINR at low IEEE 802.11a data rates

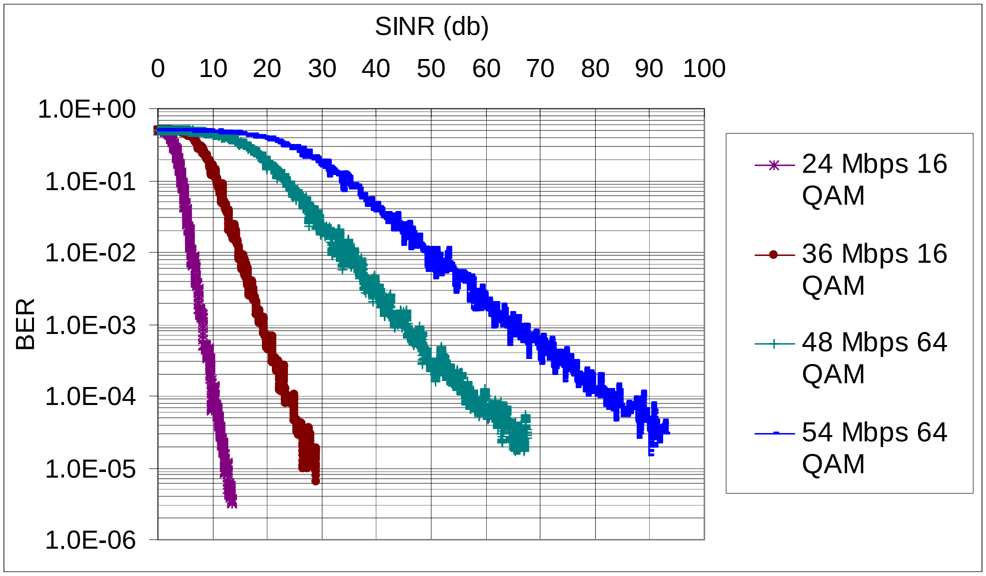


Figure 4.6: BER vs. SINR at high IEEE 802.11a data rates

$$errorProbability = 1 - (1 - BER)^{numBits}$$

Equation 1: Error probability calculation in QualNet

$$SINR_i = \frac{P_i}{\sum_{\forall\, j \neq i} P_j + FkTB}$$

Equation 2: SINR calculation in QualNet

4.4 A Simple IEEE 802.11a Analytical Model

An analytical model of IEEE 802.11a was developed to determine the accuracy of the implementation. Based on [Qiao01], for an *L*-byte long information packet to be transmitted with the 802.11a PHY and modulation mode *m*, the transmission duration to transmit and immediately acknowledge that data frame is given in Equation 3.

$$\begin{aligned} T_{data}^{m}(L) = {} & 2*(tPLCPPreamble + tPLCPHeader) + \\ & \left\lceil \frac{30.75 + L}{BpS(m)} \right\rceil * tSymbol + \\ & aSIFSTime + \left\lceil \frac{16.75}{BpS(m)} \right\rceil * tSymbol \end{aligned}$$

Equation 3: Time to transmit one MAC frame (no RTS/CTS)

The MAC header for a data frame consists of a total of 28 octets. Six "zero" tail bits and a 16-bit SERVICE field are added, resulting a total MAC overhead of 30.75 octets for that data frame. The *B*ytes-*p*er-*S*ymbol information for PHY mode *m, BpS(m)*, is given in Table 4.2. Similarly, an ACK frame consists of 16.75 octets.

For an *L*-byte long information packet to be transmitted with the RTS/CTS mechanism enabled, the transmission duration for an RTS-CTS-DATA-ACK sequence is defined in

Equation 4. An RTS frame consists of 20 octets and a CTS frame consists of 16.75 octets.

$$
\begin{aligned}
T_{data}^{m}(L) = {} & 4 * (tPLCPPreamble + tPLCPHeader) + \\
& \left\lceil \frac{30.75 + L}{BpS(m)} \right\rceil * tSymbol + \\
& 3 * aSIFSTime + 2 * \left\lceil \frac{16.75}{BpS(m)} \right\rceil * tSymbol + \\
& \left\lceil \frac{20}{Bps(m)} \right\rceil * tSymbol
\end{aligned}
$$

Equation 4: Time to transmit one MAC frame using RTS/CTS mechanism

4.5 IEEE 802.11a Model Validation

After implementation of our IEEE 802.11a model in QualNet, we compared the results of the model to the analytical model derived in Section 4.4. The experiment is set up as follows: in a 2-node topology, a constant bit rate (CBR) session is setup to transfer packet size of 512 bytes or 1472 bytes without using the RTS/CTS mechanism. The channel is assumed to be perfect. The result is shown in Figure 4.7. Our model matches closely with the theoretical analytical model.

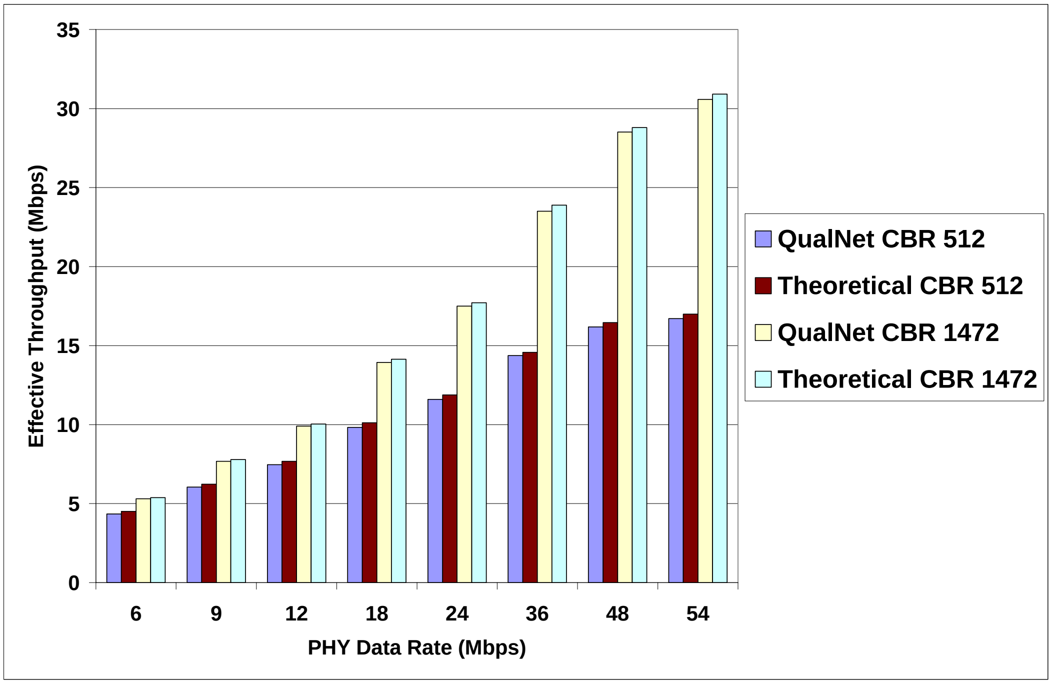


Figure 4.7: Comparison of implemented IEEE 802.11a with theoretical model without RTS/CTS mechanism

The result of the same experiment run with RTS/CTS enabled is shown in Figure 4.8. The implemented model matches the analytical model equally well.

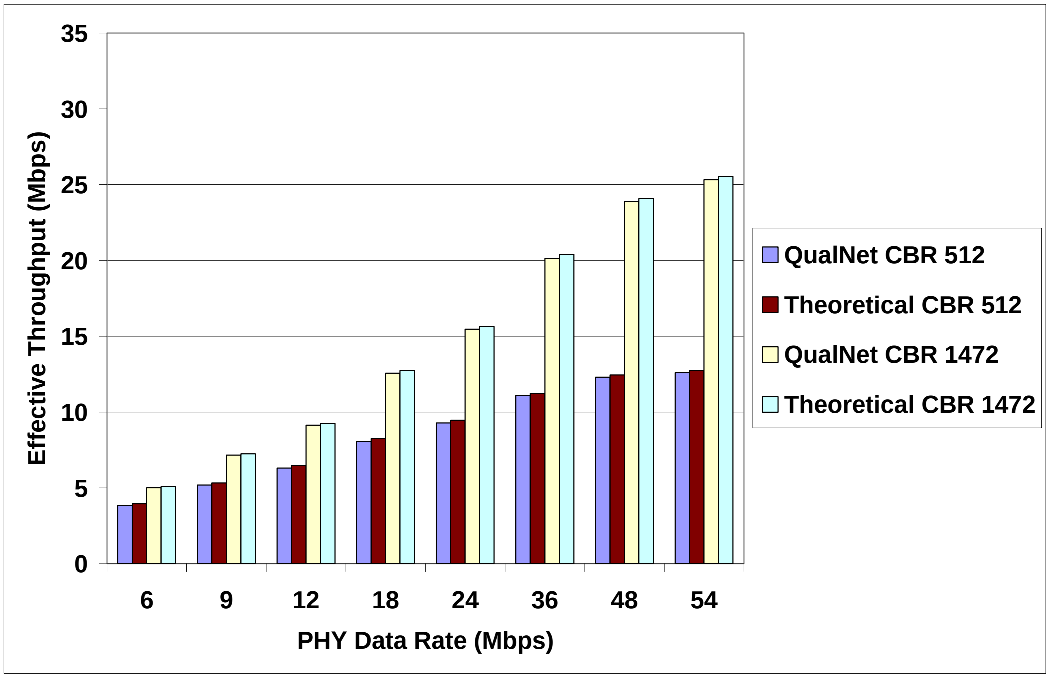


Figure 4.8: Comparison of implemented IEEE 802.11a with theoretical model with RTS/CTS mechanism

4.6 Auto Rate Fallback

A wireless device typically chooses the modulation and coding scheme, hence data rate, through a process called rate adaptation. By dynamically switching the data rate to best match the varying channel conditions, the sender hopes to select the highest data rate that the receiver can decode successfully and correctly receive the frame. Several auto rate-adjusting algorithms have been proposed in the past to take advantage of this multi-rate capability and the inherent channel condition differences among devices. The first of these proposed algorithms documented in literature is Auto Rate Fallback (ARF) [Kamerman97].

The basic idea of the ARF protocol is to keep track of the number of successful transmissions and only after a number of successful attempts do the sender attempt to send data at the next higher data rate. The sender also keeps a timer; and when that timer expires, the sender also tries to send the next packet at the next higher data rate. The protocol decreases the sender's transmission rate when it either misses two consecutive ACKs or when it fails to receive an ACK immediately after raising the transmission data rate.

The timer value, 60 *ms*, is experimentally found to be optimal in [Holland01]. One thing we noticed that was not discussed in the previous works is that in order for ARF to achieve good performance, when a node fails to receive the CTS packet after a RTS transmission, the sending node should count the missed CTS packet as an "ACK failure". Therefore, two missed CTS packets would lead to a subsequent data rate decrease. Another issue we noticed in this algorithm is the implementation complexity involved. A node must keep a history of all transmissions to other nodes separately. This is because the nodes must differentiate between each other as their channel conditions are likely not the same relative to each other. Later in Section 4.7, we will validate the performance of ARF with state-of-the-art IEEE 802.11a hardware.

4.7 Validation with Existing Hardware

IEEE 802.11a compliant products are beginning to appear on the commercial marketplace today [Intel][Linksys][SMC]. We obtained two Intel PRO/Wireless 5000 CardBus Adapters and performed several benchmarking experiments using NetPerf [Jones92]. NetPerf is a benchmarking tool that can be used to measure the performance of many different types of networks. It provides test for both unidirectional throughput and end-to-end latency.

We installed the cards on two different laptops and placed the cards in ad hoc mode. Using NetPerf's default UDP test suite, we measured the single hop performance of the Intel cards at UDP frame size of 1024 and 1472 bytes with and without medium access reservation messages. The laptops were placed 5 feet apart in an office environment. Similarly, we simulated the same scenario in QualNet with Ricean fading enabled. A K-Factor of 5 dB is used for the Ricean parameter. The K-Factor is the ratio of LOS (line-of-sight) to NLOS (non-line-of-sight) that determines what percentage of the energy is coming from a direct LOS source as opposed to reflective sources. A K-Factor of 5 dB is typical of office environments with harsh multipath conditions. A list of simulation parameters is shown in Table 4.3. The TX power, RX sensitivity, and RX threshold for each modulation constellation are taken from [SMC]'s product documentation.

Channel frequency		5.2 [GHz]
Signal reception		BER based
Data rate		ARF
Antenna gain		2.15 [dBi]
Fading model		Ricean
K-Factor		5.0 [dB]
BPSK	**TX Power**	20.0 [dBm]
	RX Sensitivity	-85.0 [dBm]
	RX Threshold	-76.0 [dBm]
QPSK	**TX Power**	19.0 [dBm]
	RX Sensitivity	-83.0 [dBm]
	RX Threshold	-73.0 [dBm]
16-QAM	**TX Power**	18.0 [dBm]
	RX Sensitivity	-78.0 [dBm]
	RX Threshold	-68.0 [dBm]
64-QAM	**TX Power**	16.0 [dBm]
	RX Sensitivity	-69.0 [dBm]
	RX Threshold	-59.0 [dBm]

Table 4.3: Set of parameters used for simulator implementation verification

The results of this verification are presented in Figure 4.9. We can conclude that our implementation of ARF and the derived BER performance tables for different data rates are satisfactory. However, we note that the proprietary auto rate adjustment algorithm implemented in the Intel LAN cards is not publicly documented and therefore might account for a significant amount of the differences between our simulated and the benchmarked results. Furthermore, we do not know the exact channel condition of the office environment.

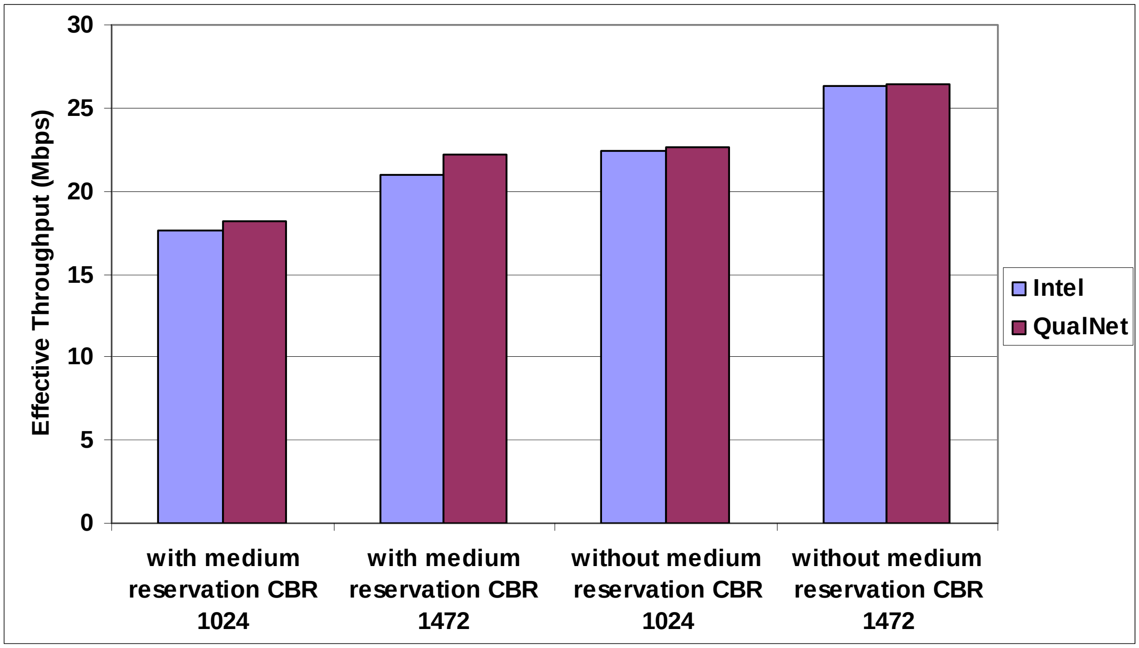


Figure 4.9: Simulator versus actual hardware comparison

Chapter 5 Integration of OFDM Simulator into QualNet

5.1 The OFDM Simulator Overview

An OFDM simulator is built using MATLAB Simulink by Mr. Alireza Mehrnia and Dr. Babak Daneshrad. Simulink is a simulation and prototyping environment for modeling dynamic systems [Simulink]. The OFDM simulator contains a large number of variable parameters that leads to a myriad of channel conditions and BER rates. The relevant variable parameters for the purpose of this study include:

- Modulation type – BPSK, 4-QAM, 16-QAM, 64-QAM
- Multipath – up to six channel tap delays and loss
- Number of effective subcarriers – 33-1024 subcarriers
- Number of symbols in cyclic prefix and cyclic postfix
- Transmitter antenna gain, receiver antenna gain
- Mean transmit power, receiver noise figure
- SINR
- Frequency offset

These variable parameters are first fed into MATLAB. A dynamic channel is then generated base on the input parameters. The OFDM model in Simulink, upon the start of simulation, generates a stream of bits and modulates them to the specified modulation scheme. Modulation is the process of translating an outgoing data stream into symbols for transmission by the sender. The symbols are then brought to the transmitter RF front end and simulated across the generated channel. On the receiver side, the OFDM receiver locks on the incoming signal and the receiver baseband demodulates the signal back to the stream of bits. The transmit and receive bits are compared and BER is calculated base on the number of error bits and the total number of bits send. Furthermore, the receiver calculates the effective SINR per OFDM subcarrier seen at the receiver baseband. The average received effective SINR is calculated at the end of simulation. Simulation of 100 OFDM symbols takes about 50 seconds on 2.4 GHz Intel Xeon machine equipped with 512 MB of memory. A picture of the OFDM simulator is shown in Figure 5.1.

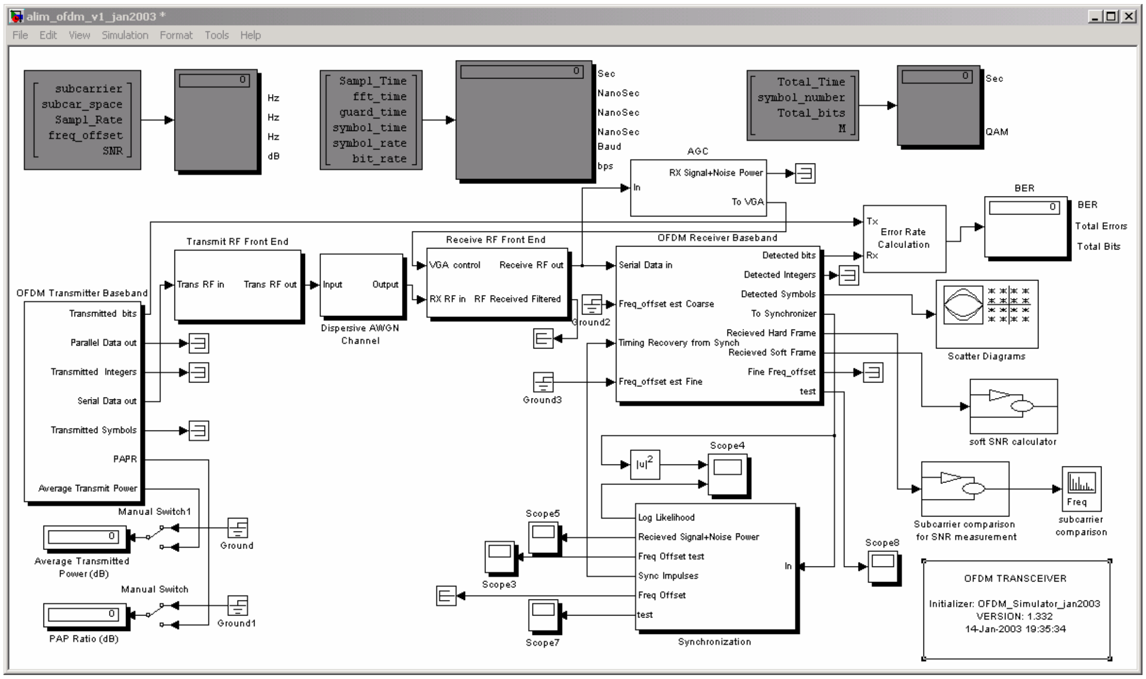


Figure 5.1: The MATLAB Simulink OFDM Simulator

5.2 In-depth Explanation of the OFDM Simulator

The simulated OFDM system works as follow. The transmitter baseband, the leftmost center block and in expanded view in Figure 5.2, generates data symbols based on the specified modulation and number of subcarriers. A random bit generator is used to provide the input stimulus for the system instead of having data feed in from a MAC layer. Pilot symbols are added and the last OFDM symbol is zero-padded prior to the IFFT. Guard blocks are added by cyclically pre-pending and post-pending the specified number of data samples to the beginning and ending of each individual OFDM symbol. Finally, the preamble block generates the preamble, which consists of training symbols for packet detection, frequency offset, and channel estimation at the receiver. It is important to note that FEC (Forward Error Correction) coding is not yet implemented in this simulator; the transmitted bits are uncoded unlike IEEE 802.11a.

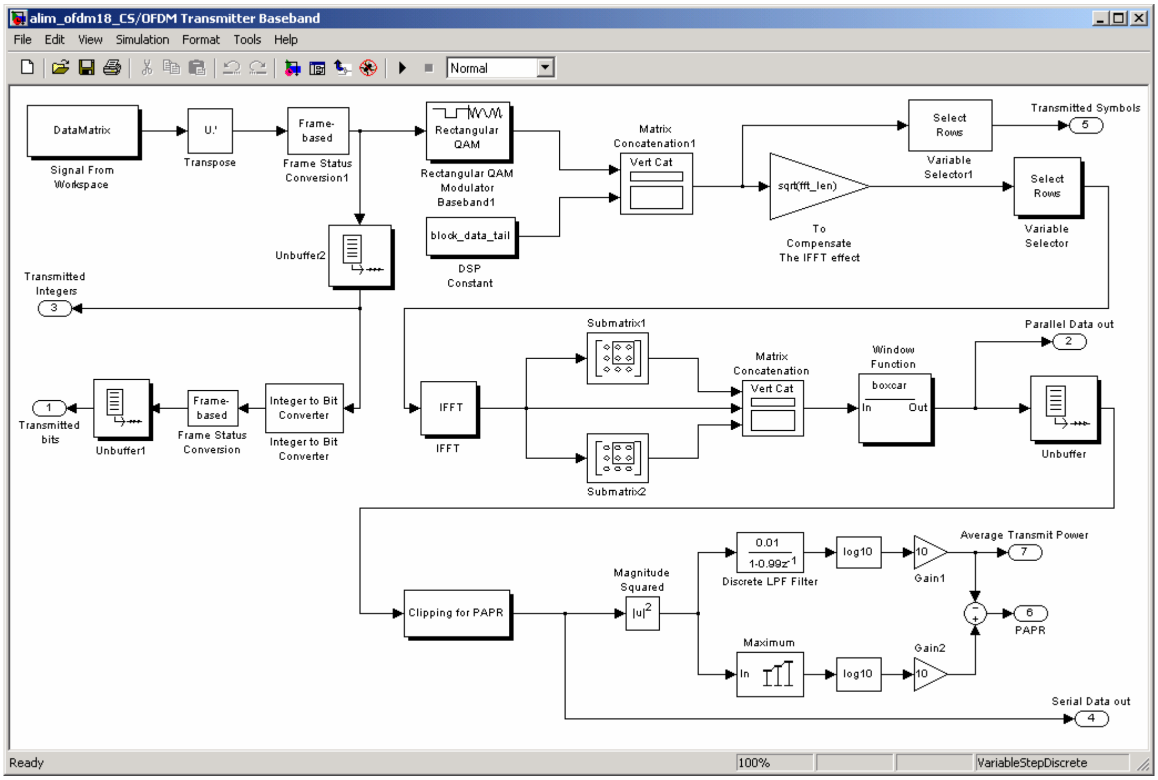


Figure 5.2: OFDM Simulator transmitter baseband expanded view

The RF front-end, the block that the transmitter baseband feeds into, transforms the information signals into radio frequency (RF) carriers. Since RF carriers are sinusoids, the three salient features are its amplitude, phase, and frequency. After the RF front-end is simulated, the information signals are then simulated in the dynamic channel model, Figure 5.3. In wireless communication, signals are subjected to distortions caused by reflections and diffractions generated by the signals' interactions with obstacles and terrain conditions. The distortions experienced by the signals include delay spread, attenuation in signal strength, and frequency shifting. In addition, multipath, the reception of multiple transmission paths to the receiver, also affects the receiver performance. Under the assumptions of Gaussian scatters (AWGN) and multiple propagation paths to the receiver, the channel is characterized by time-varying propagation delays, attenuation factors, and Doppler shifts. The Doppler effect occurs when the source and receiver are moving relative to one another and can cause significant problems in OFDM systems because the transmission technique is sensitive to carrier frequency offsets.

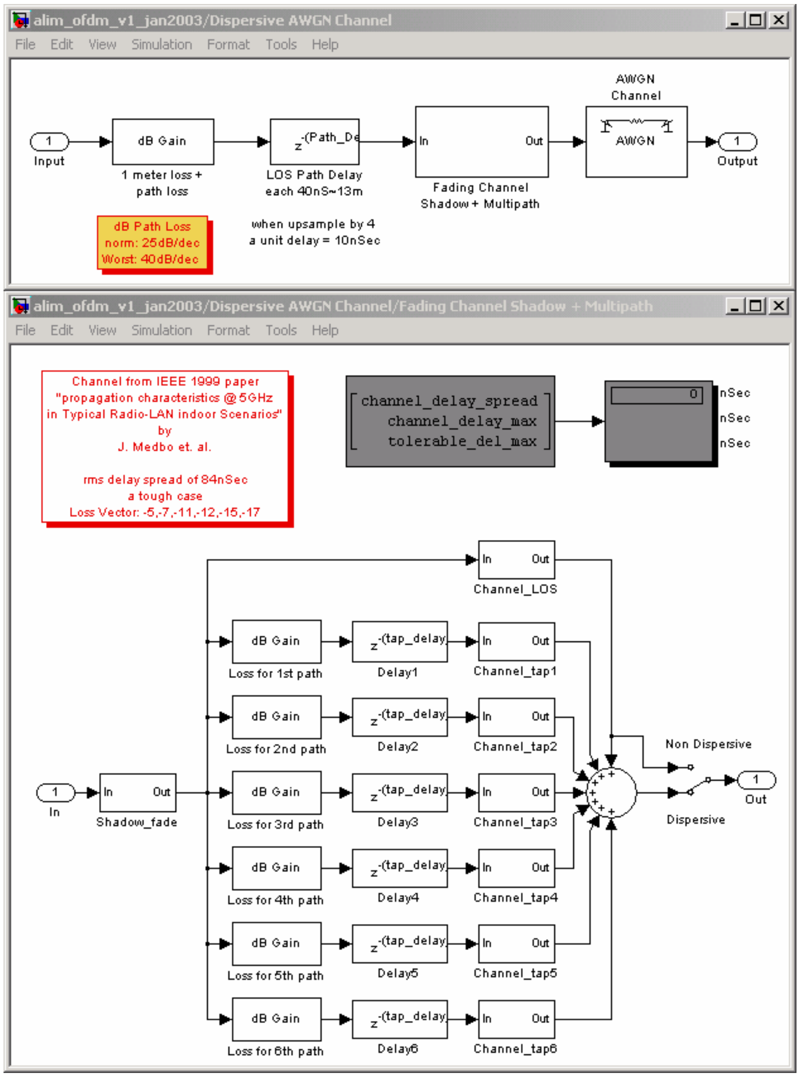


Figure 5.3: OFDM Simulator channel model expanded view

At the receiver, the transmitted information embedded in the RF carrier must be recovered. The receiver must decide which of the possible digital waveforms most closely resembles the received signal, taking into account the effects of the channel. The receiver front-end consists of an ADC (Analog-to-Digital Converter). The receiver baseband, Figure 5.4, first performs the functions of packet detection, time synchronization, and removal of the symbol cyclic prefix and postfix. After the fine-time-synchronizer block, the synchronized signal goes to the FFT block. The remove pilots block removes the pilot carriers and reorders the data carriers from the FFT block.

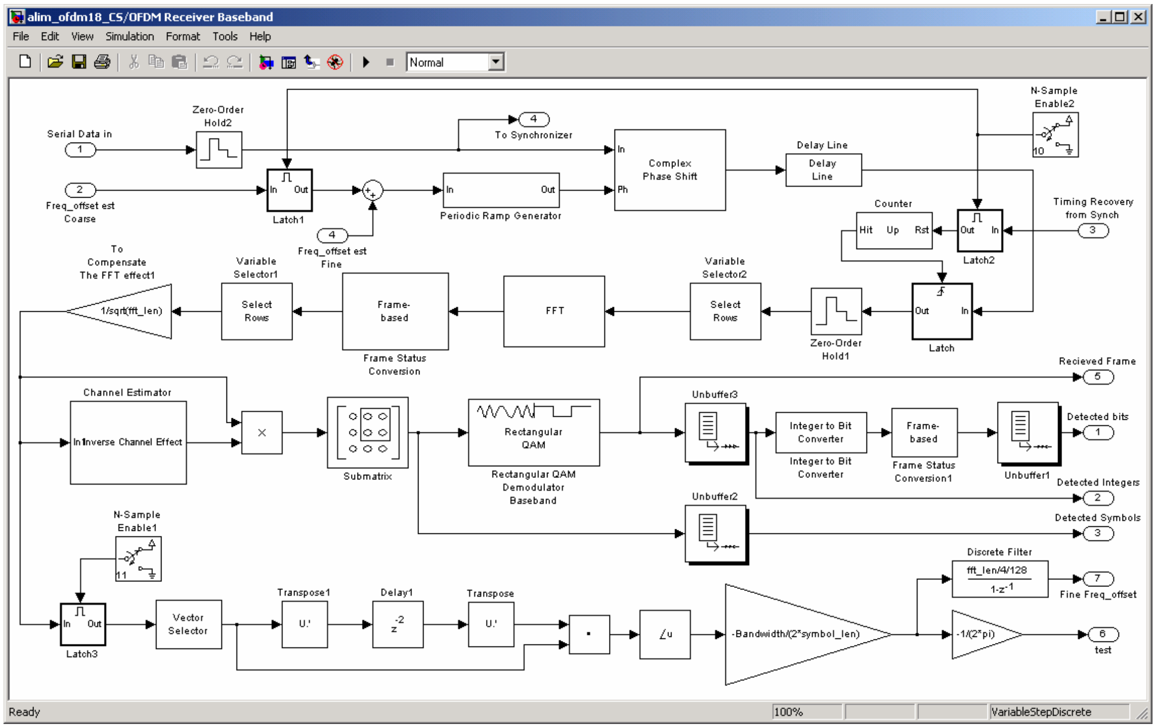


Figure 5.4: OFDM Simulator receiver baseband expanded view

Detailed simulation of the radio device and channel allows the ability to rapidly prototype and test new physical layer algorithms and ideas. The integration of the detailed model into a network simulator allows designers to see the effects of physical layer technology on higher level performance in a multi-granular simulation environment. The performance data captured from this integrated simulator is much more accurate because each bit in the packet is modulated, simulated across a dynamic channel, and demodulated at the receiver, whereas typical BER/SINR curves would not accurately capture the effects of the device and ever changing channel. Using the integrated detailed simulation model, one can make recommendations on the viability of the radio design by analyzing the performance of the whole mobile system, not just the performance a particular network layer.

5.3 Integration of OFDM Model into QualNet

This section discusses some implementation issues associated with the integration of the OFDM model into QualNet. As QualNet is developed using a layered approach, we can modify the implementation details at a particular layer without affecting other layers. To integrate the OFDM model, the physical layer in QualNet was modified to invoke the OFDM model when necessary. The OFDM simulator simulates on a time granularity on the order of nanoseconds or per OFDM symbol basis while QualNet simulates on a time granularity of 10s of microseconds or per packet basis. Obviously, there is much to be gained from a more refined simulation model.

When a QualNet node detects an incoming signal, it first determines if that signal is above the receiving threshold (RXT). If the signal is above the specified RXT value, the radio tries to receive the signal. SINR is calculated from the strength of the signal and the noise of the channel. QualNet does not currently model device hardware, Doppler, or frequency offset effects. Hence, the integration of the OFDM model is carried out as follows: the QualNet node's original SINR, $SINR_{in}$, is fed into the detailed OFDM model. In combination with the user specified multipath, Doppler, and frequency offset, a dynamic channel is generated. The OFDM model is then simulated and the resulting SINR, $SINR_{out}$, seen at the receiver baseband, is used to calculate the loss defined in Equation 5. This loss value, as we will explain later, is then stored in a table inside QualNet. The new SINR result is then used to calculate weather or not the packet errors by mapping it to a BER value. Notice that, the simulated receiver SINR from the model is used instead of the model BER because of the short length of the data packet. The BER value would not be accurate for such short packet simulation. Also, while we could have just return weather the packet errored or not at the end of the detailed simulation, it would not have allowed us to use our novel simulation speedup techniques nor allowed us to develop a probabilistic model based on the detailed simulation. In our approach, the interaction of the device with the channel is modeled.

$$SINR_{out} = \frac{Signal_{in} - Loss}{Noise_{in} + Loss}$$

Equation 5: Loss value calculation

In `Phy802_11CheckRxPacketError` of `phy/phy_802_11.c`, the detailed simulation model is inserted with a system call to MATLAB `(system("matlab -nosplash -nodesktop -r QualNet-Online-Driver");)`. The file format of `QualNet-Online-Driver` is as follows:

```
[Bandwidth, DataMatrix, LOS, M, Path_Delay, Path_Loss,
RX_NF, RXgain, Rec_Si_Power, SNR, Sampl_Rate,
Sampl_Time, Si_Power, Simulate_for_Predefined_SNR,
TXgain, Total_Time, Total_bits, VGA_table, alpha,
bit_rate, block_data_tail, chan_pilots1,
channel_delay_max, channel_delay_spread, constant1,
constant2, ctrl_tap, cyclic_post, cyclic_prefix,
distance, fc, fft_len, fft_time, fm, freq_offset,
guard, guard_time, loss_1m, noise_pow,
normalize_tap_loss_vec, not_used_subcar,
num_synch_pilot, rayleigh, setup_time, std_shadow,
subcar_space, subcarrier, symbol_len, symbol_number,
symbol_rate, symbol_time, synch_pilots, synch_select,
tap_delay_vec, tap_loss_vec, tolerable_del_max,
var_noise] =
OFDM_Simulator_jan2003_no_gui(Modulation_Type,
Number_of_OFDM_Symbols_to_Simulate, Transmitter_SNR);
sim('alim_ofdm_v1_jan2003');
```

In the function above, the modulation type, number of OFDM symbols to simulate, and transmitter SINR are fed into the OFDM system with the channel parameterization. A host of other parameters, all given values to mimic the IEEE 802.11a standard, is exported to the MATLAB workspace. `Cyclic_post`, `guard_time`, `subcar_space`, `subcarrier`, `symbol_len`, `symbol_rate`, and `num_sync_pilot` were all given IEEE 802.11a specified values as described earlier in Chapter 4.2. Other parameters are described later in the simulation setup, Section 6.1. At the end of the OFDM simulation, the mean receiver SINR is written out to a file as follows:

```
    fid=fopen('MatlabResults.txt', 'wt');
    fprintf(fid, '%f\n', mean(mean_soft_SNR));
    fclose(fid);
```

This SINR is used as the basis of packet error calculation in QualNet and is read into QualNet by the same `Phy802_11CheckRxPacketError` function.

As mentioned earlier, simulation of the OFDM model is time consuming. While bit level simulation in wireless environments is desirable, large-scale network simulations must trade off between simulation execution time and accuracy. Simulation time of this integrated system is considerably reduced via two methods: simulation of only a portion of the data frame and a caching mechanism to cache similar scenarios.

While evaluating the OFDM simulator, it is noticed that the simulated resulting SINR value does not change significantly after the simulation of a certain number of OFDM symbols. This is because the transmission duration is less than the coherence time. The coherence time of the channel is a measure of the speed at which the channel characteristics change. Using this fact, the OFDM simulation was stopped after the SINR measurement stabilized, which was after 40 OFDM symbols. This reduced simulation time as typical packet transmission length might last for 100s of OFDM symbols. For example, a 1472 byte packet modulated at 6 Mbps would transmit 503 OFDM symbols.

More significantly, a caching mechanism was developed to take advantage of scenarios with similar SINR and channel conditions. That is, after running the OFDM simulator at a given SINR and channel condition, the loss resulted from that run would be saved. The loss value is the signal strength loss; it becomes part of the noise. When a similar SINR and channel condition transmission occurs, the resulting SINR is calculated using Equation 5 with the loss value previously cached. The loss value is cached instead of the resulting SINR value because the granularity of the input SINR is rounded to the nearest integer; an input SINR of 11.5 dB and 12.4 dB would map to the same loss value, not the same SINR. Caching the original resulting SINR value would be inaccurate because of the large granularity; but using the loss value calculated, a realistic effective SINR value that includes the effects of the device and channel is obtained. Using the newly calculated SINR value, the corresponding BER is looked up. The error probability for the packet is then calculated and the packet is tested for error. Simulation runtime is sped up considerably with this caching mechanism as we will discuss later in 6.4.

Chapter 6 Simulation Studies

6.1 Scenario Descriptions and System Parameters Setup

This chapter quantifies the effects of the OFDM radio and channel modeling on typical scenarios used in the performance evaluation of MANETs. Scenarios for this comparison are created as follows: each scenario is configured with a stationary 50-node network placed over a 1000m x 1000m terrain. We assumed that the scenario simulates a flat terrain that is grided into a standard pattern and each radio is placed randomly within a unique cell. Twenty-five nodes are randomly chosen to be CBR (Constant Bit Rate) sources, each of which generates 512-byte data packets to a randomly chosen destination at a rate of 5, 10, 20, 40, and 60 packets per second in a total simulation time of 90 seconds. The network uses AODV (Ad Hoc On-Demand Distance Vector) [Perkins99] routing for each CBR source to discover a route to the destination. Each data point represents the average value from seven runs with different random number seeds. With the different seeds, the node placement and CBR sessions in the network are set differently. Some common parameters are listed in Table 6.1. The transmission power and receiver sensitivity are taken from [SMC], a commercial implementation of the IEEE 802.11a.

Channel frequency		5.2 [GHz]
Effective Subcarriers		48
Data rate		24 Mbps, ARF
Antenna gain		0 [dBi]
BPSK	**TX Power**	20.0 [dBm]
	RX Sensitivity/Threshold	-85.0 [dBm]
QPSK	**TX Power**	19.0 [dBm]
	RX Sensitivity/Threshold	-83.0 [dBm]
16-QAM	**TX Power**	18.0 [dBm]
	RX Sensitivity/Threshold	-78.0 [dBm]
64-QAM	**TX Power**	16.0 [dBm]
	RX Sensitivity/Threshold	-69.0 [dBm]

Table 6.1: Set of parameters used by QualNet in the simulation studies

In this evaluation, two data rate types were chosen. First, every node is set to transmit only at 24 Mbps. This corresponds to the 16-QAM modulation in the OFDM model. Second, each node uses the Auto Rate Fallback algorithm for automatic rate adjustment. The OFDM constellation will vary between BPSK, 4-QAM (QPSK), 16-QAM, and 64-QAM depending on the data rate. Table 6.2 contains a list of parameters fed into the OFDM model by QualNet, considered as typical outdoor conditions. All of the variable parameters are chosen to mimic the IEEE 802.11a characteristics described earlier in Chapter 4.2.

Fading model	Rayleigh
Doppler Spread	250.0 [Hz]
Number of Cyclic Prefix	20
Number of Cyclic Postfix	1
Path loss exponent	3

Table 6.2: Set of parameters used by OFDM model in the simulation studies

6.2 Packet Delivery Ratio and MAC Total Retransmission at Fixed Rate

The PDR (Packet Delivery Ratio) performance of the integrated OFDM model simulation is significantly lower than that of the original abstract model when the transmitting data rate is fixed. As the network load increases, the PDR decreases considerably due to packet transmission error and channel congestion as shown in Figure 6.1. At the highest packet rate scenarios, the detailed OFDM model simulation result PDR is only one-third of that of the original abstract model. Figure 6.2 shows the number of MAC retransmission attempts. In the figure, the difference between the integrated model and abstract model is clear. The number of retransmission attempts is significantly higher for the integrated OFDM model simulation. This correlates well with the lower PDRs depicted in Figure 6.1. At 40 and 60 packets per second per flow, the number of MAC retransmission attempts is closer to that of the abstract model. One can draw the conclusion that network becomes saturated at this point. Similarly, as expected, the average number of MAC packet drops per node is much higher in the integrated OFDM model than the abstract model as shown in Figure 6.3. Again, network saturation is seen in the 40 to 60 packets per second per flow range.

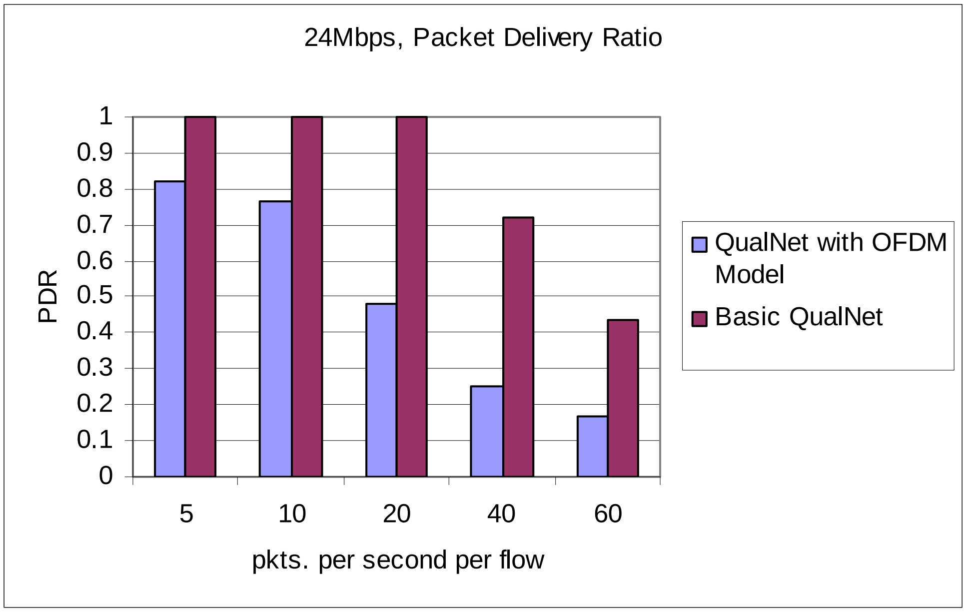


Figure 6.1: PDR of integrated detailed OFDM model vs. abstract model at 24 Mbps

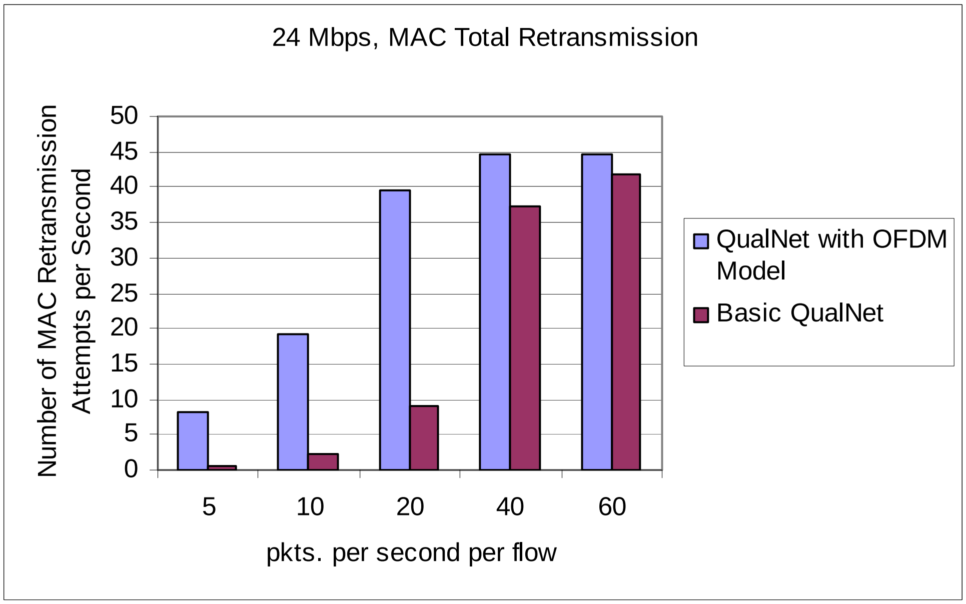


Figure 6.2: Number of retransmission attempts of integrated detailed OFDM model vs. abstract model at 24 Mbps

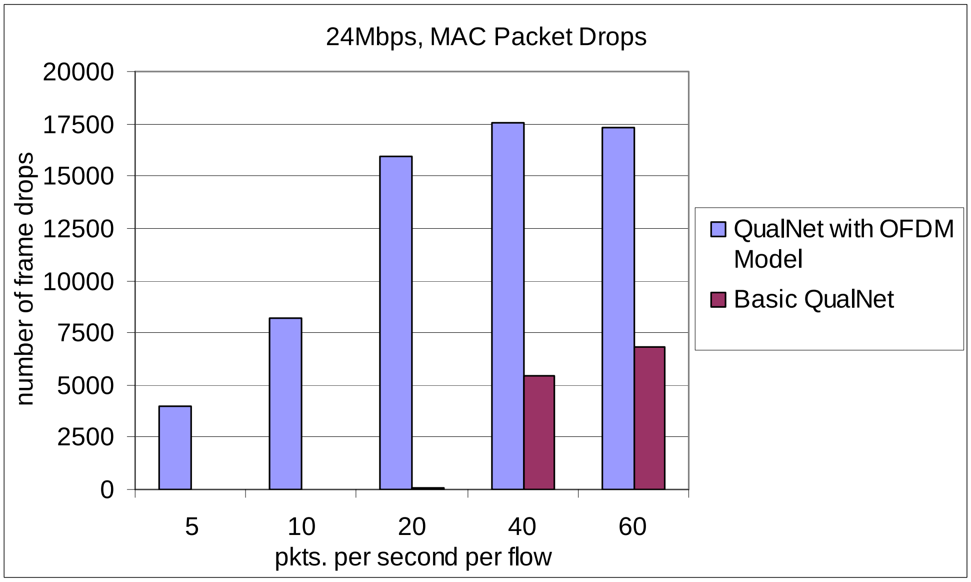


Figure 6.3: Number of MAC packet drops of integrated detailed OFDM model vs. abstract model at 24 Mbps

6.3 Packet Delivery Ratio and MAC Total Retransmission Using ARF

The results are quite different when each node uses ARF as its data rate control algorithm. The two different simulation models' PDR, number of retransmission attempts, and average MAC packet drops per node closely match each other as shown in Figure 6.4, Figure 6.5, and Figure 6.6. Because ARF automatically adjust data rates based on channel conditions, in sparse network scenarios, ARF can lower the node's transmitting data rate to ensure packet delivery without overloading the transmission medium. By comparing the PDR of Figure 6.1 and Figure 6.4 at 5, 10, and 20 packets per second per flow, it is easily seen that ARF takes advantage of the sparse traffic to ensure packet delivery. It is also clear that the gradual PDR decrease from the OFDM model in Figure 6.1 is caused by other wireless network traffic interference. ARF adapts to light load noisy environments well. However, as the packet rate increases, ARF is actually detrimental to PDR performance. Notice that the PDR performance in Figure 6.1 at 40 and 60 packets per second per flow is higher than that of Figure 6.4's. By lowering the data rate, ARF, in highly congested environments, causes longer packet transmission duration and in effect longer delays and more queue overflows. This leads to a lower PDR ratio in congested scenarios when compared with fixed data rate scenarios.

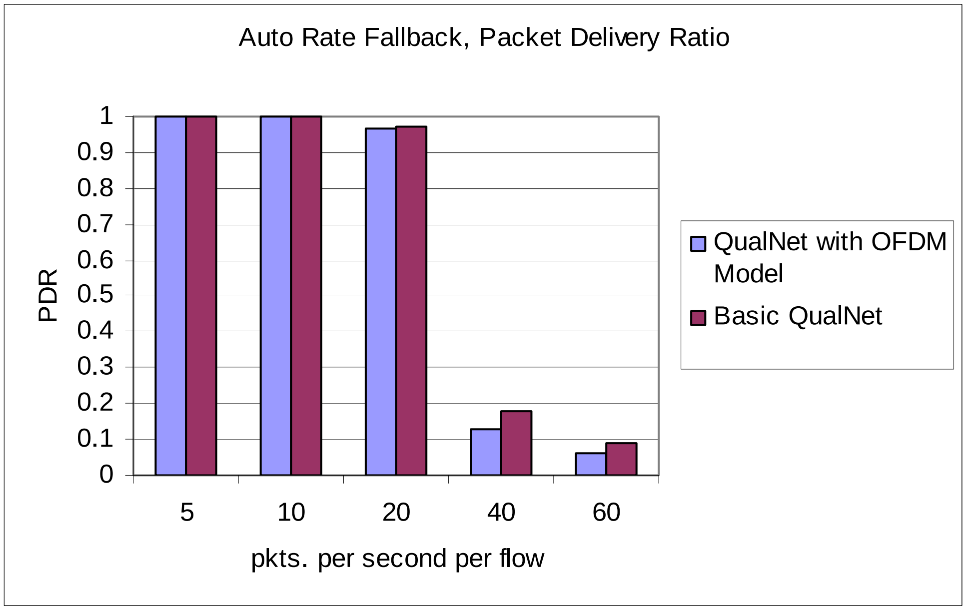


Figure 6.4: PDR of integrated detailed OFDM model vs. abstract model using Auto Rate Fallback

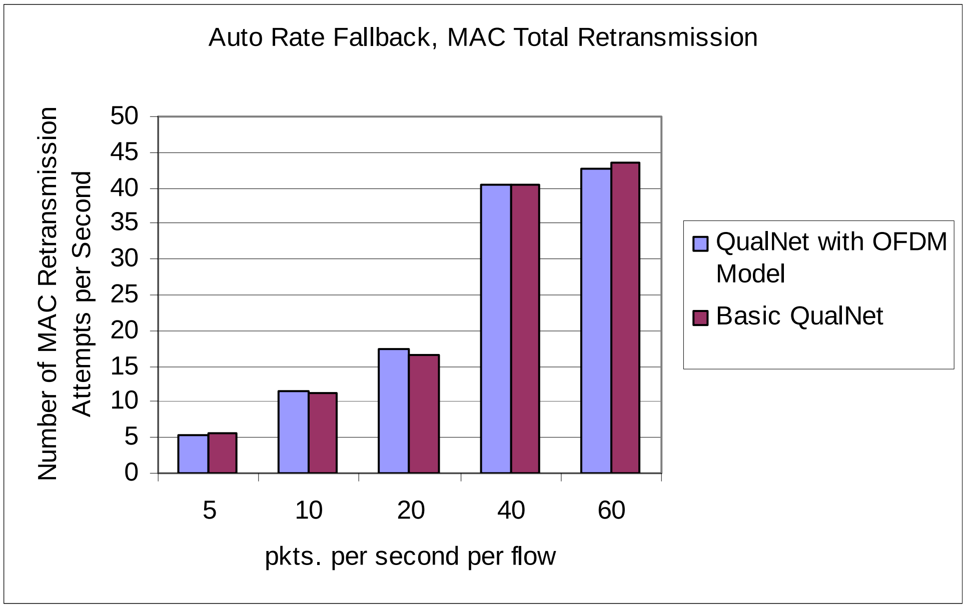


Figure 6.5: Number of retransmission attempts of integrated detailed OFDM model vs. abstract model using Auto Rate Fallback

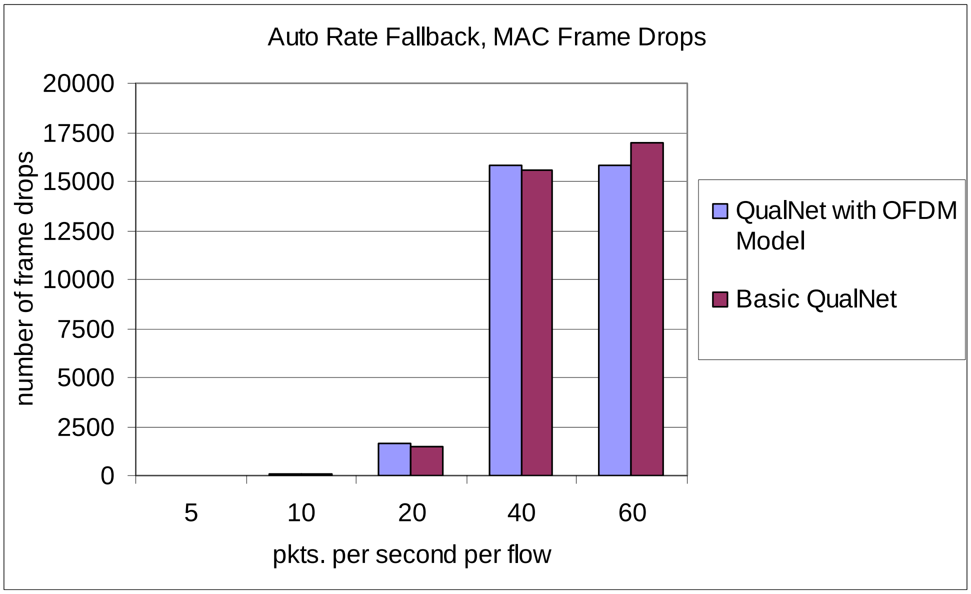


Figure 6.6: Number of MAC packet drops of integrated detailed OFDM model vs. abstract model using Auto Rate Fallback

6.4 Simulator System Performance

As previously stated, OFDM simulation is very computationally expensive. On average, only 100 OFDM symbols are simulated in 50 seconds on a modern Intel 2.4 GHz Xeon processor machine. MAC data frames are on the order of hundreds of OFDM symbols. While detailed simulation of every bit of the network is desirable, one cannot expect to use this OFDM simulator to simulate every packet for large MANET scenarios. Our integration technique (described in Chapter 5.3) of caching the signal loss and partial transmission simulation is novel in that it captures the interaction the wireless channel with the radio device and yet still maintaining a reasonable execution time to allow for large MANET simulations. Figure 6.7 depicts the execution speedup benefit of using the signal loss cache detailed model method as oppose to simulating every single bit in the network. The cache detailed model method is able to scale with the abstract model whereas it would have taken years to simulate an otherwise simple scenario using the every bit detailed OFDM simulation model. In the simulation, a stationary 50-node network is placed over a 1000m x 1000m terrain. Twenty-five nodes are randomly chosen to be CBR sources, each of which generates 512-byte data packets to a randomly chosen destination at a rate of 20 packets per second. The network uses AODV for each CBR source to discover a route to the destination. Each data point represents the average value from three runs with different random number seeds. The X-axis in Figure 6.7 shows the average number of signals locked on by each receiver. It is easily seen that detailed simulation of every bit does not scale while the cache detailed simulation model is able to scale with the abstract model.

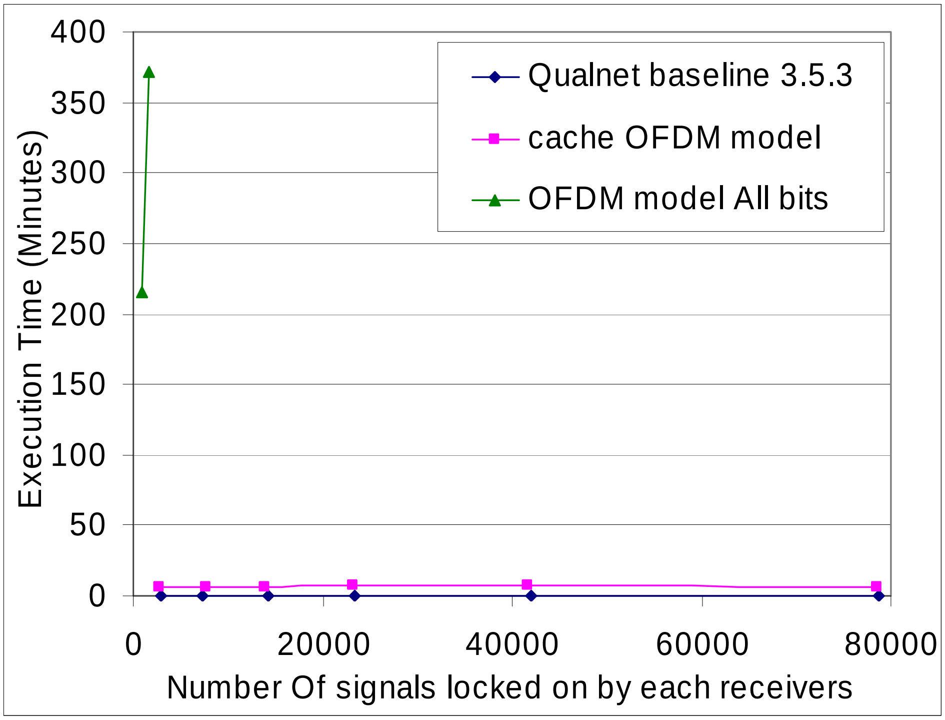


Figure 6.7: Execution time comparison

In another test case, the nodes are randomly chosen to send 512 byte data packets at a rate of 10 and 20 packets per second per flow. The execution time versus the simulation time of a 1, 2, 4, 8, and 16 minutes network is plotted in Figure 6.8. Each data point represents the average of five trail runs. We were not able to run the every data bit detailed simulation model on even a 1-minute network simulation. On the other hand, the cache detailed model is able to scale up for large, lengthy network scenario simulations.

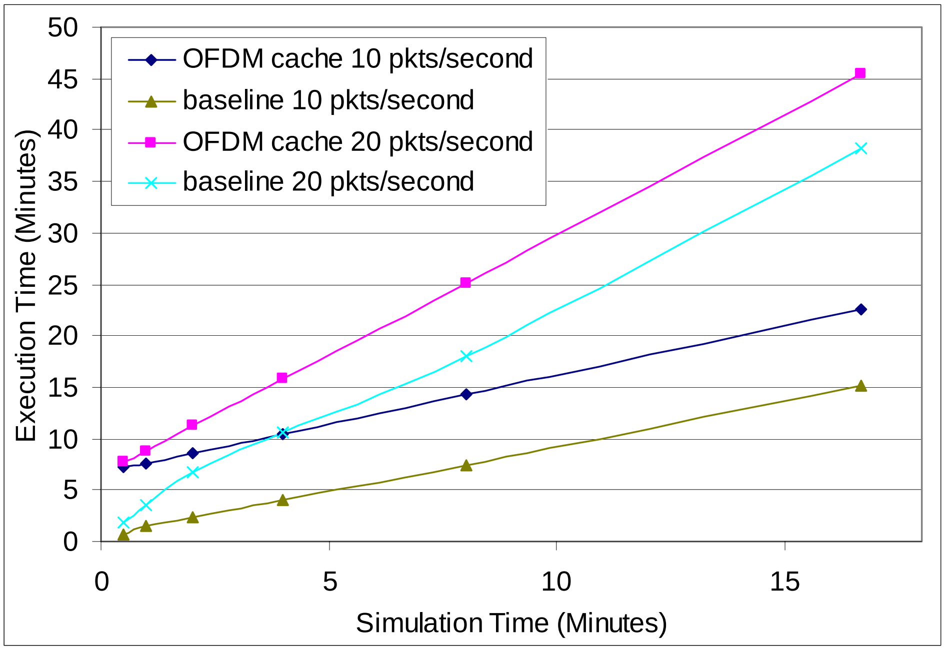


Figure 6.8: Execution versus simulation time comparison

Chapter 7 Conclusions

This thesis has focused on the effects of detailed OFDM and channel modeling on the performance evaluation of higher layer protocols. An approach to integrate a detailed OFDM simulator to a packet level simulator is presented along with the implementation and verification of an abstract IEEE 802.11a model. This integration provides a realistic and efficient model of the propagation and device layer for network performance analysis. The results show that device and wireless channel can impact packet delivery ratios and even point out a deficiency of the ARF protocol. The need for accurate and efficient physical layer models in network simulation is clear. Traditionally, radio engineers have analyzed the performance of their designs against others only on point-to-point performance checks under varying channel conditions. Now engineers can evaluate the performance of their designs on the radio's cross layer interactions and higher layer behavior. Also, the integration delineates a method in which simulators of dramatically different time granularities are combined using network layer boundary APIs.

In terms of performance, our novel integration technique and cleanly defined interfaces are clearly beneficial in any traffic scenario. Using pure OFDM simulation to simulate every bit in the network is too computationally expensive and it leads to unscalable and unreasonable execution time. Our technique scales along with the basic abstract model and still captures the essence of the radio device and its performance characteristics in varying wireless channels. The results show that significant benefits can be obtained from our caching and partial simulation technique, with a caveat that careful evaluation of what to cache and the length of the partial simulation duration must be properly understood.

With advances in antenna, modulation, and coding technology, it becomes increasingly important for higher network layers to understand their interactions with the physical device and medium. Future work on the detailed OFDM model includes enhancements to the cache scheme, dynamic channel and fading characteristics utilizing detailed 3-D terrain models and movement of the nodes. The issues of OFDM synchronization protocols and algorithms, FEC coding/decoding and interleaving, smart antenna and/or diversity processing, chip size and package type, and power consumption, among others, should also be considered for study as well. Moreover, it might be possible to implement this OFDM model in a FPGA and integrate it into QualNet. One could simulate in real time, with the actual radio to determine the effects of implementation performance loss on higher layer application protocols.

Other important physical layer technologies, MIMO (Multiple-In-Multiple-Out) antenna, SDR (Software Define Radio), and UWB (Ultra Wide Band) models could be integrated into QualNet for system evaluation of these alternative technologies. Also, the detailed models will be compiled into binary format using the MATLAB Compiler or natively written in C to further speed up execution time. The ideas of our multi-granular, split stack framework could be used to incorporate protocols and device models originally developed in various simulation and measurement tools into a single framework. The goal is to avoid unrealistic assumptions that may produce misleading simulation results whose impact is known only after when the system has been deployed.

REFERENCES

[Agrawal01] Agrawal A., Bakshi A., Davis J., Eames B., Ledeczi A., Mohanty S., Mathur V., Neema S., Nordstrom G., Prasanna V., Raghavendra, C., Singh M.: MILAN: A Model Based Integrated Simulation Framework for Design of Embedded Systems, Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES), , Snowbird, UT, June, 2001.

[Bharghavan94] Vaduvur Bharghavan, Alan Demers, Scott Shenker, Lixia Zhang, “MACAW: A media access protocol for wireless LAN’s,” in proceedings of ACM SIGCOMM, 1994.

[Buck94] J. T. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, “Ptolemy: A framework for simulating and prototyping heterogeneous systems,” International Journal of Computer Simulation, special issue on “Simulation Software Development,”, vol. 4, pp. 155–182, April 1994.

[Chang95] W.-T. Chang, A. Kalavade and E. A. Lee, ``Effective Heterogeneous Design and Cosimulation,'', chapter in Hardware/Software Co-design, G. DeMicheli and M. Sami, eds., NATO ASI Series Vol. 310, Kluwer Academic Publishers,1996. Also presented at NATO Advanced Study Institute Workshop on Hardware/Software Codesign, Lake Como, Italy, June 18 -- 30, 1995.

[Davis02] Davis J.: Model Integrated Computing: A Framework for Creating Domain Specific Design Environments, The 6th World Multiconference on Systems, Cybernetics, and Informatics (SCI), , Orlando, FL, July, 2002.

[Eker03] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong, “Taming heterogeneity—the Ptolemy approach,” Proceedings of the IEEE, vol. 91, no. 1, pp. 127–144, Jan. 2003.

[Fullmer97] C. L. Fullmer and J. J. Garcia-Luna-Aceves, “Solutions to hidden terminal problems in wireless networks,” in proceedings of ACM SIGCOMM 1997, pages 14-18, September 1997.

[GloMoSim] GloMoSim: Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/projects/glomosim/.

[Holland01] G. Holland, N. H. Vaidya and P. Bahl, “A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in proceedings of MobiCom 2001, July 2001.

[ieee5ghz99] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band, Supplement to IEEE 802.11 Standard, September 1999.

[ieee80211_99] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Standard, IEEE, August 1999.

[Intel] Intel PRO/Wireless 5000 CardBus Adapter Product Spec., http://intel.com/network/connectivity/products/5000_cardbus_adapter.htm.

[Jones92] Rick Jones. The Public NetPerf Homepage. available at http://www.netperf.org/netperf/NetperfPage.html. Anderson, R.E. Social impacts of computing: Codes of professional ethics. Social Science Computing Review, 2 (Winter 1992), 453-469.

[Kamerman97] A. Kamerman, L. Monteban, “WaveLAN-II: A high performance wireless LAN for the unlicensed band,” Bell Labs Technical Journal, Vol. 2, No. 3, pages 118-113, Summer 1997.

[Karsai93] Karsai G., Padalkar S., Franke H., Sztipanovits J.: Model-based Programming Tools for Integrated Monitoring, Simulation, Diagnosis, and Control, Proc. of Computing in Aerospace 9, AIAA paper 93-4578, 1993.

[Keller00] Thomas Keller and Lajos Hanzo. "Adaptive Multicarrier Modulation: A Convenient Framework for Time-Frequency Processing in Wireless Communications,” in proceedings of the IEEE, Vol 88, No. 5, May 2000: 609-640.

[Lao94] A. Lao, ``Heterogeneous Cell-Relay Network Simulation and Performance Analysis with Ptolemy," Memorandum No. UCB/ERL M94/8, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA 94720, February 17, 1994.

[Ledeczi03] Ledeczi A., Davis J., Neema S., Agrawal A.: Modeling Methodology for Integrated Simulation of Embedded Systems, ACM Transactions on Modeling and Computer Simulation, (accepted), 2003.

[Lee01] Edward A. Lee, "Overview of the Ptolemy Project," Technical Memorandum UCB/ERL M01/11, University of California, Berkeley, March 6, 2001.

[Lee02] Edward A. Lee, Stephen Neuendorffer and Michael J. Wirthlin,"Actor-Oriented Design of Embedded Hardware and Software Systems," Invited paper, Journal of Circuits, Systems, and Computers , Version 2, November 20, 2002. Technical Memorandum UCB/ERL M02/13, University of California, Berkeley, CA 94720, May 1, 2002.

[Linksys] Linksys Dual-Band Wireless A+B Broadband Router User Guide, ftp://ftp.linksys.com/pdf/wrt51ab_ug.pdf.

[Liu03] Xiaojun Liu, Jie Liu, Johan Eker, and Edward A. Lee, "Heterogeneous Modeling and Design of Control Systems," Software-Enabled Control: Information Technology for Dynamical Systems, Tariq Samad and Gary Balas (eds.), Wiley-IEEE Press, April 2003.

[Lokesh99] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Rajive Bagrodia, "Simulation of Large-Scale Heterogeneous Communication Systems", Proceedings of MILCOM'99, November 1999.

[MANET] IETF manet (Mobile Ad-hoc Networks) working group http://www.ietf.org/html.charters/manet-charter.html.

[Maple] Waterloo Maple User's Guide, http://www.maplesoft.com/.

[Mathematica] Mathematica User's Guide, http://www.wolfram.com/.

[Matlab] MATLAB User's Guide, http://www.mathworks.com/.

[Misra00] Vishal Misra, Weibo Gong, Don Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in proceedings of ACM SIGCOMM, 2000.

[Nee00] Richard Van Nee, Ramjee Prasad, OFDM for Wireless Multimedia Communications. Boston, MA: Artech House, 2000.

[ns2] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/.

[Opnet] Opnet User's Manual, http://www.opnet.com/.

[Perkins99] Charles. E. Perkins, Elizabeth. M. Royer, "Ad-hoc on-demand distance vector routing," in proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Application, pages 90-100, 1999.

[Pino95] J. L. Pino, S. Ha, E. A. Lee and J. T. Buck, ``Software Synthesis for DSP Using Ptolemy," Journal on VLSI Signal Processing, vol. 9, no. 1, pp. 7-21, Jan., 1995.

[Riley01] Riley, G. F.; Ammar, M. H.; Fujimoto, R. M.; Perumalla, K.; Xu, Donghua ``Distributed Network Simulations using the Dynamic Simulation Backplane" Proceedings of International Conference on Distributed Computing Systems 2001 (ICDCS'01)

[Qiao01] Daji Qiao and Sunghyun Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," in proceedings of IEEE ICC, June 2001.

[QualNet] QualNet User's Manual, http://www.qualnet.com/.

[Rappaport95] T. S. Rappaport, "Wireless Communications: Principles & Practice," Prentice Hall, 1995.

[SNT] Scalable Network Technologies, http://www.scalable-networks.com/.

[Simulink] Simulink User's Guide, http://www.mathworks.com/products/simulink/.

[SMC] SMC2755W – EZ Connect 802.11a Wireless Access Point User Guide, http://www.smc.com/drivers_downloads/library/SMC2755W_MN.pdf/.

[Takai00] M. Takai, R. Bagrodia, K.Tang, M. Gerla, "Efficient Wireless Network Simulation with Detailed Propagation Model," ACM Wireless Network Journal, December 2000.

[Takai01] Mineo Takai, Jay Martin, Rajive Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc network," in proceedings of MobiHoc 2001, pages 87-94, October 2001.

[Terry01] John Terry, Juha Heiskala, OFDM Wireless LANs: A Theoretical and Practical Guide. Indianapolis, IN: Sams Publishing, December 2001.

[Xiang98] Xiang Zeng, Rajive Bagrodia, Mario Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks," in proceedings of the 12th Workshop on Parallel and Distributed Simulations -- PADS '98, May 26-29, 1998 in Banff, Alberta, Canada.

[Xu01] Xu, Donghua; Riley, G. F.; Ammar, M. H.; Fujimoto, R. M. ``Split Protocol Stack Network Simulations using the Dynamic Simulation Backplane" In

Proceedings of the Ninth International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems, 2001 (MASCOTS'01)

[Yung01] Tak-Kin Yung, Jay Martin, Mineo Takai, Rajive Bagrodia, “Mixed fluid flow and packet level simulation models for large scale networks,” in proceedings of SPIE, Vol 4523, August 2001.

[Zhou03] Junlan Zhou, Zhengrong Ji, Mineo Takai, Rajive Bagrodia, “Maya: a Multi-Paradigm Network Modeling Framework,” in proceedings of PADS 2003.