Using Software-Defined Radio to Validate Wireless Models in Simulation

Ketan Mandke and Scott M. Nettles

Wireless Networking & Communications Group (WNCG), Dept. of Electrical & Computer Engineering
The University of Texas at Austin, 1 University Station C0803, Austin, TX 78712-0240
Email: kmandke@utexas.edu, nettles@ece.utexas.edu

***Abstract*— Simulation is an efficient and powerful tool for studying wireless networking. The lack of physical layer (PHY) accuracy in many popular network simulators such as ns-2 and OPNET has been observed to produce misleading results. In this paper, we demonstrate a technique for validating the packet-level PHY models in network simulators against the physical layer of a software-defined radio testbed. Specifically, we evaluate the accuracy of model-based simulations against hybrid (i.e., joint packet and waveform) simulations of the real PHY implementation directly executing in a simulation environment. We demonstrate that this direct-execution approach is an effective means of evaluating the accuracy of packet-level PHY models from the perspectives of different protocol layers using a broad set of operating conditions. Moreover, this is an effective means of establishing when a packet-level PHY model can be used to generate trustworthy simulation results.**

***Index Terms*— wireless network simulation; model validation; direct-execution simulation**

I. INTRODUCTION

Simulation is an important tool in studying wireless networks. In particular, it is widely used in the study of IEEE 802.11-style networks, such as WLAN, mesh, and ad hoc networks. The performance of wireless networks is profoundly influenced by the behavior of the physical layer (PHY) and the characteristics of the wireless environment. A lack of physical layer accuracy, however, in many popular network simulators such as ns-2 [2] and OPNET [3] has been shown to lead to incorrect conlusions [4], [5]. This motivates our work, which focuses on validating the PHY models used in network simulators as this is critical for establishing trust in the accuracy of results dervied from simulations [6], [7].

Conventional approaches for physical layer model validation involve comparing model-based simulations against measurements taken from a real-world testbed. This methodology can be limited by the difficulty in controlling and reproducing experiments in real wireless environments [8], as well as the difficulty in verifying the consistency between a simulated system and a real-world deployment.

In this paper, we present an alternative approach for physical layer model validation using a software-defined radio (SDR) testbed. Specifically, we directly execute the PHY implementation from an *operational* SDR testbed in a simulation environment to enable highly accurate hybrid (i.e., joint packet-level and waveform-level) simulations, thus creating a benchmark for evaluating the accuracy of packet-level PHY models. This direct-execution approach allows for fair side-by-side comparisons between a PHY model and a real implementation using the *exact same* communication protocols, traffic patterns, and wireless channels.

We use this direct-execution approach to evaluate the accuracy of the leading OFDM PHY model used to simulate 802.11-style networks in state-of-the-art network simulators, such as ns-3 [9], [10]. First, we use direct-execution to characterize link-level behavior of the PHY under a variety of wireless conditions. We show that, from the perspective of the physical layer, the model is accurate under a limited set of operating conditions and can produce incorrect results in more realistic wireless conditions.

Next, we study the operation of a rate adaptive MAC protocol under various wireless conditions and use direct-execution to evaluate the accuracy of the PHY model from the perspective of the MAC layer. We use our link-level characterization of the PHY to modify the policies of the protocol so that it operates "correctly" in a realistic wireless channel. In doing so, we also cause the PHY to operate in a regime where the model is more accurate and can therefore be used to generate trustworthy simulation results.

Our main goal in this paper is to demonstrate how direct-execution is an effective means of evaluating the accuracy of packet-level PHY models from the perspectives of different protocol layers. Further, this approach allows us to consider a broad set of wireless conditions to establish when model-based simulations will generate trustworthy results.

The remainder of the paper is organized as follows. Section II provides additional motivation and details of our direct-execution methodology. In Sections III and IV, we introduce the tools that form the foundation for the specific implementation of our approach. Section V presents results that characterize the link-level behavior of the physical layer. Then in Section VI we investigate model accuracy in terms of its impact on a rate adaptive MAC protocol. Finally, we conclude with a discussion of related works and a summary of our results in Sections VII and VIII.

This material is based in part upon work supported by the National Science Foundation under grant CNS-0626797.

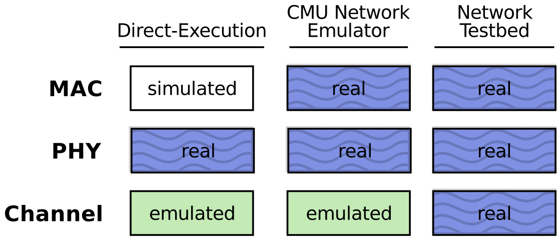

Fig. 1: Different approaches to validation.

II. Methodology

The physical layer models used in network simulators are responsible for simulating the packet reception behavior of a real implementation. This behavior can vary from one implementation to the next. As such, we must consider what it means when we attempt to answer the question: *is my model valid?* Here, we discuss the goals of physical layer model validation, conventional approaches to validation, and more about our direct-execution approach.

A. Role of Validation

The goal of validation is to establish that a model is a sufficiently accurate representation of a real system for its intended application [11]. In the context of packet-level PHY models, validation should establish that a model is an accurate representation of a *specific* implementation in a *specific* wireless scenario. Because of the intricate and complex interactions between the PHY, wireless channel, and higher-layer protocols, a specific validation effort cannot always be used to establish accuracy in a broader context. Simulation users, however, use PHY models to study many diverse wireless scenarios, often relying on external validation efforts to establish the accuracy of their results.

In practice, model validation should be used to address questions about accuracy as they relate to the concerns of an individual simulation user. That is, for a particular network scenario, a model user will be concerned with whether or not their simulations have produced trustworthy results about performance measurements at a particular protocol layer.

We believe that model validation should be considered in the investigative process for all simulation users. Effective model validation should provide the flexibility to consider the accuracy of a PHY model from the perspectives of different protocol layers and under a variety of wireless conditions. Our direct-execution approach provides an efficient means of evaluating the accuracy of PHY models in this manner.

B. Conventional Approaches to PHY Validation

The most common approach to validating the physical layer model of a network simulator is to compare model-based simulations against measurements taken from a testbed or emulator. The perspectives used in these kinds of validation studies can vary. For example, Bredel and Bergner compared MAC performance and fairness metrics from OMNet++ simulations against measurements taken from a wireless LAN testbed in order to validate the wireless models in the simulator [12]. One of the drawbacks of this approach is the difficulty in deploying and reliably generating repeatable results in real-world testbeds [8].

Borries, et al. compared the packet reception model in ns-2 against the behavior of a commercial WiFi card using a real-time emulation platform (CMU Network Emulator); studying model accuracy from the perspective of the physical layer [13]. This approach provides a reliable way of generating repeatable experimental results and a mechanism for looking at different wireless channel conditions. One of the drawbacks of this emulation approach (and testbeds) is that the higher-layer protocols used in real hardware may not be consistent with those in the simulated system. This makes it more difficult to reasonably assert the validity of a PHY model from the perspective of a higher-layer protocol.

Many different examples of these approaches exist in the literature [5], [7], [14], [15]. Figure 1 illustrates the main operational differences between these approaches and our direct-execution approach.

C. Direct-Execution

This paper presents an alternative approach for validation that utilizes *direct-execution* of the physical layer implementation from an operational SDR testbed. By replacing this one part of the simulated wireless system, we enable highly accurate simulations that establish a benchmark for evaluating the accuracy of packet-level PHY models. This approach allows us to compare the behavior of model-based simulations against that of a real implementation, using the *exact same* implementations of all other parts of the wireless system (e.g., wireless channels, topology, traffic sources, and communication protocols). In this way, direct-execution provides a *fair apples-to-apples comparison* between the model and a real physical layer implementation.

We use this direct-execution approach to answer two validation questions about the PHY model in our simulator: (i) *how does my PHY model differ from a real system?* and (ii) *how does this impact the accuracy of my network simulations?* Our goal is to demonstrate that direct-execution is an effective means of studying model accuracy from the perspectives of different protocol layers using a broad set of wireless conditions, thereby answering these questions.

III. A Software-Defined Radio Testbed

The physical layer implementation for our direct-execution approach comes from an operational SDR testbed called Hydra. This testbed features a completely software-defined protocol stack; PHY, MAC, and routing protocols are implemented entirely in software running on a general purpose processor. Each Hydra node consists of a laptop connected to a flexible RF front end. This unique testbed was designed as a flexible platform for rapidly prototyping new wireless algorithms and protocols [16]. In particular, we used Hydra as a tool for studying cross-layer issues in wireless networks. The testbed has been used extensively for prototyping and experimenting in practical real-world settings [17]–[20].

A. Details of the Physical Layer

The physical layer of Hydra implements the PHY specification of IEEE 802.11n [1]. Since the PHY model in our network simulator is intended to represent the physical layer of a *single-antenna* wireless device, we limit our study in this paper to the single-antenna, single spatial-stream modes of IEEE 802.11n. We refer the reader to [1] for details on the modulation and coding schemes defined by the standard.

We now specify some of the design choices and specific details of our physical layer implementation. In particular, these details may help in identifying if the Hydra physical layer should be used (or how it might be modified) to represent another implementation. Our PHY performs frame detection using the Schmidl-Cox algorithm (SCA) [21]. The PHY also uses the preamble of each frame to do least-squares (LS) channel estimation. This channel estimate is used for zero-forcing frequency domain equalization (ZF-FEQ) [22]. The physical layer can also be configured to use hard or soft-decision Viterbi decoding [23].

B. Real-World Wireless Impariments

There are a variety of wireless impairments that can adversely impact the performance of the physical layer of a real system. From our own observations doing real experiments, we identified that frequency selectivity in the wireless channel and carrier frequency offset between the transmitter and receiver significantly impacted Hydra. While radio impairments such as IQ imbalance and quantization noise did not significantly impact Hydra, the direct-execution approach in this paper could also be applied to develop a better understanding of these and other impairments.

1) Frequency-Selectivity: Transmitted radio signals reflect and scatter off different objects in the wireless environment. As a result, multiple delayed and attenuated copies of the original transmission can arrive at the receiver in such a way as to cause variations (or frequency selectivity) in the frequency response of the received signal. Frequency selective (or multipath) channels are common in modern wireless systems that operate over wide bandwidths [22]. Figure 2 illustrates the impact that a frequency selective channel might have on the spectrum of a transmitted signal.

2) Carrier Frequency Offset (CFO): The RF carrier that a device modulates its signal to and from can vary from one device to the next. As depicted in Fig. 3, after filtering the RF front end at the transmitter modulates the signal to the frequency f_t (close to the intended carrier frequency f_c). At the receiver, the signal is demodulated from f_r (also close to f_c), low-pass filtered (LPF), and processed by the physical layer. While the PHY attempts to compensate for the resulting carrier frequency offset ($\Delta f = f_t - f_r$) as best it can, residual CFO left after compensation can result in phase rotations and adversely impact decoding performance.

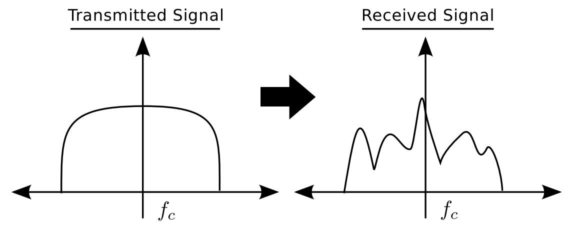

Fig. 2: Frequency response over a multipath channel.

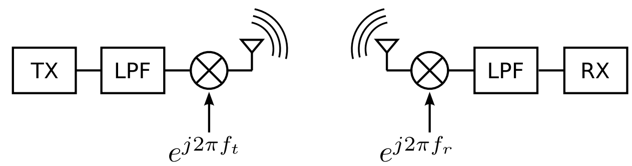

Fig. 3: Carrier frequency offset $\Delta f = f_t - f_r$.

IV. A Wireless Network Simulator

We developed a detailed wireless network simulator, called WiNS, as a tool for enabling high-fidelity simulations [24]. This tool was originally conceived of as a response to the skepticism surrounding many popular network simulators, particularly ns-2. The main design goal was to more accurately model the PHY and its interactions with the wireless channel and higher-layer protocols. Our development of WiNS coincided with that of ns-3, and shares the same physical layer decoding model. In this paper, we demonstrate our direct-execution approach using WiNS, but this approach could also be applied to ns-3 and other simulators.

WiNS is well-suited as a complementary tool to experimentation on the Hydra testbed. In particular, the design of the simulator draws on many aspects of Hydra. The majority of WiNS is implemented in Python, which requires no compilation and provides automatic memory management and garbage collection. Computationally intensive pieces of the simulator are implemented in C/C++, wrapped with a Python wrapper, and integrated into the simulator. This approach was used to integrate the Hydra PHY into WiNS and enable direct execution.

A. Physical Layer Reception Model

The physical layer reception model of a simulator is responsible for modeling two main functions: (i) frame detection and (ii) packet decoding. The PHY model in WiNS is intended to be used to represent the single-antenna operation of an OFDM physical layer from an IEEE 802.11 a/g/n device. We note that the packet-decoding model in WiNS is for all intents and purposes the same one employed by other state-of-the-art network simulators, namely ns-3 [9]. As such, our validation study also has implications about the fidelity of other simulators using this model.

1) Frame Detection: The detection model in WiNS seeks to mimic the stochastic behavior of the detection algorithm employed by Hydra (i.e., SCA). WiNS uses a parameterized model for frame detection that is fit to measurements collected from waveform-level simulations of the Hydra physical layer. The probability of successful frame detection, or frame-detection rate (FDR), is described as a function of signal-to–noise ratio (SNR) in dB:

$$\mathrm{FDR} = \begin{cases} 0, & \mathrm{SNR} < \Gamma_l \\ \frac{\mathrm{SNR}-\Gamma_l}{\Gamma_h-\Gamma_l}, & \Gamma_l < \mathrm{SNR} < \Gamma_h \\ 1, & \mathrm{SNR} > \Gamma_h \end{cases} \tag{1}$$

where Γ_l (1.5 dB) and Γ_h (5 dB) are low and high thresholds for detection. If the SNR of a packet is less than Γ_l, it will always be dropped; if SNR is greater than Γ_h, then the packet will always be detected. Intermediate values result in a random detection outcome based on FDR.

In contrast to this approach, most network simulators (including ns-3) use a basic threshold policy for determining if a packet is detected [9]. This is contrary to the stochastic nature of how detection occurs in most real physical layers.

2) Packet Decoding: After a frame has been successfully detected, the decoding model in WiNS determines if the PHY successfully decodes the data in the packet. This model is based on closed-form expressions for digital modulation (M-QAM, M-ary quadrature amplitude modulation) and the hard-decision decoding performance of convolutional codes presented by Qiao, Choi, and Shin [25]. Their results are based on analysis over an additive white Gaussian noise (AWGN) channel. As we mentioned, *this is the same model used in other state-of-the-art simulators, namely ns-3*.

This stochastic decoding model does not attempt to model the location or number of errors in a packet, only the outcome of the decoding process. At a high-level, it does this by calculating the packet-error rate (PER) and using this to randomly determine the success (or failure) of the decoding process. The basic procedure for calculating PER starts with using modulation order (M), coding rate (r), and link quality (i.e., SNR) to compute bit-error rate (P_b). This is then used to compute the PER for an L-byte packet as follows:

$$PER \approx 1 - (1 - P_b)^{8L} \tag{2}$$

For brevity, we omit details of calculating bit-error rate (BER) and refer the reader to [26] (or [10] or [9]) for a detailed discussion of this process.

B. Integrating a SDR Physical Layer

Incorporating the Hydra physical layer into WiNS required minimal modifications to the PHY model in WiNS. The inputs to the packet-level model already include parameters for the configuration of the transmitter, timing information, and pathloss information for the wireless channel. To enable direct-execution, we added the transmitted and received waveforms to this list of meta-data used by the PHY.

Meta-data in WiNS is implemented using flexible packet annotations that can be attached to each packet. In this way, each packet contains all of the information needed for the PHY to process it and simulate the outcome of detection and decoding. After modifying the transmitter and receiver so that they could create and process waveforms to and from packets using the Hydra PHY, the integration of the real PHY required the addition of two final components: (i) a realistic receive buffer and (ii) realistic wireless channel models.

1) Realistic Receive Buffer: When using direct-execution, the receiver applies the waveform-level representation of the wireless channel to the transmitted waveform annotated by each packet. Then using the timing information in the packet, this waveform is added to the physical layer's receive buffer. After that, signals in the receive buffer can be processed by the physical layer, as they would be in the real SDR system. The processed packet is then marked with a new packet annotation to indicate whether or not the packet was successfully detected and decoded.

2) Realistic Wireless Channel: To simulate a realistic wireless channel, WiNS uses the channel models specified by the IEEE 802.11n Task Group (TGn). These channel models represent a diverse set of indoor wireless scenarios, including time-varying frequency-selective (multipath) wireless channels. In this paper, we will look at results using the frequency-flat TGn Channel Model A (CM-A) and frequency-selective TGn Channel Model D (CM-D). We refer the reader to [27] for more details on the parameters and construction of the TGn channel models.

V. Link-Level Characterization

We now use direct-execution of the Hydra physical layer to evaluate the accuracy of the PHY model in WINS. This first set of experiments will consider the link-level behavior of the physical layer. First, we examine the behavior of the PHY in an AWGN channel with long and short packets. After using direct execution to validate our physical layer model under this scenario, we will compare the performance of the model to that of the Hydra physical layer under the two wireless impairments we discussed earlier, namely frequency-selectivity and carrier frequency offset.

A. Simulation Setup

Here we consider the communication between a single sender and receiver. First, we use direct-execution to evaluate the accuracy of *frame detection* in the PHY model under various wireless conditions. Then, we use this approach to characterize the *packet decoding* performance using low, medium, and high data rates: 6.5 Mbps, 26.0 Mbps, and 58.5 Mbps. In particular, we measure packet-reception rate as a function of the signal-to-noise ratio (SNR) of the link. We restrict our investigation to a representative subset of the available single-antenna modes in IEEE 802.11n so that our graphs will be more readable. The following four scenarios are considered in our results.

1) AWGN: We consider decoding performance in a frequency-flat channel sending maximum Ethernet size frames (1500 bytes). This is effectively an AWGN channel and helps to establish a baseline for model accuracy in the scenario for which the model was designed.

2) Short Packets: Then we consider the impact of using a shorter packet length (40 bytes) in this AWGN channel. We consider these packet sizes as they are the two most common packet lengths found in Internet traffic [28].

3) CFO: Next, we characterize PHY performance in the presence of carrier frequency offset using large (1500 byte) packets. CFO is modeled as a uniformly distributed random variable with a maximum offset of 13.675 ppm [29], which at a carrier frequency of 5 GHz corresponds to a maximum offset of approximately 70 kHz.

4) Multipath: Finally, we characterize link-level behavior in the frequency-selective channel defined by TGn Channel Model D (CM-D). The average RMS delay spread for the channels in our experiments was 56 ns.

B. Simulation Results

For the results in this section, we measured PHY performance as a function of the instantaneous SNR of the link. If we were to consider a time-varying (i.e, fading) channel, our results would be averaged over the specific distribution of SNRs the channel sees. We argue that this would obfuscate the actual accuracy of our model, and would not allow us to generalize our results to different fading channels.

In each scenario, we vary the sender-receiver separation distance to achieve a specific SNR. At each SNR, the sender transmits 100 packets, which we use to measure frame-detection rate and packet-reception rate. We use error bars to indicate 90% confidence intervals. All the results represent measurements averaged over 10 trials.

As a side note, we understand that one might argue that as the OFDM PHY model was not designed to represent a system with the wireless impairments in our study, it is not fair to assess the accuracy of the model with respect to these impairments. We posit, however, that characterizing the difference between the model and Hydra in this way provides insight into how to improve the PHY model to account for these impairments, and gives us a mechanism for identifying when the current model can be used to generate trustworthy results in simulations that consider these impairments.

1) Frame Detection: Figure 4 compares the detection performance of the parameterized model for frame detection in WiNS against direct-execution of the Hydra PHY in an AWGN channel, a frequency-selective (multipath) channel, and in the presence of CFO. As the graph shows, detection rate gradually rises with link SNR, and above 5 dB nearly all packets are successfully detected. In ns-3, a frame is successfully detected if the SNR of the frame exceeds the SNR capture threshold (5 dB) [9]. We have also included this threshold-based policy in the graph as a point of reference.

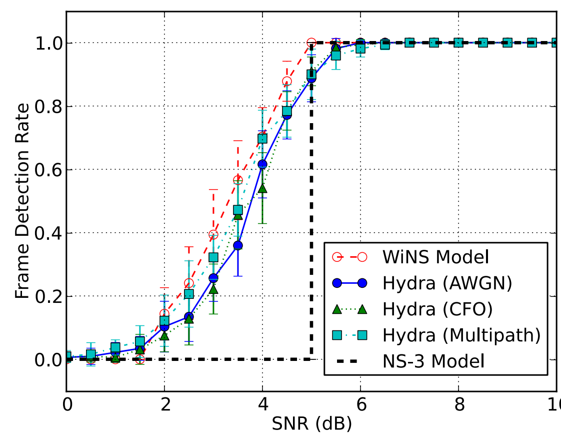

Fig. 4: Frame detection performance of model and Hydra.

From the results in Fig. 4, we see that the frame detection model in WiNS accurately captures the behavior of the real physical layer under all of the wireless scenarios in our study. This result is expected for the AWGN and CFO scenarios, since the empirical model is based on measurements in an AWGN channel and the Schmidl-Cox algorithm (used for detection in Hydra) is robust against carrier frequency offset. We were not sure, however, that this model would accurately capture behavior in multipath channels. Such channels can cause interference between the correlation terms in the Schmidl-Cox algorithm, which could adversely impact the detection performance of the PHY. To our surprise, the model continues to closely match the behavior of the real Hydra physical layer even in this frequency-selective channel.

2) Packet Decoding: We now use direct execution to evaluate the accuracy of packet decoding in our PHY model and study the impact of various wireless impairments. Each graph shows packet-reception rate (conditioned on successful frame detection) as it varies with link SNR for three data rates. We use solid lines to connect data corresponding to direct-execution of the Hydra physical layer, and dotted lines to connect data collected from model-based simulations.

AWGN: Figure 5 compares the decoding behavior of the model against that of the directly executing Hydra PHY in an AWGN channel. For each data rate, we see that the recepetion rate rises as the SNR of the link increases. These results indicate that the behavior of the model closely matches that of the Hydra PHY. Moreover, they validate the decoding behavior of the model for the AWGN scenario.

Short Packets: Figure 6 presents decoding performance of the physical layer using short 40 byte packets. These results show the model is slightly more pessimistic than the performance indicated by direct-execution of the Hydra PHY. Specifically, the Hydra physical layer begins to decode small packets at a SNR a few dB below that needed for the model to start decoding the packet. This difference in behavior is most notable at higher data rates. We speculate that this

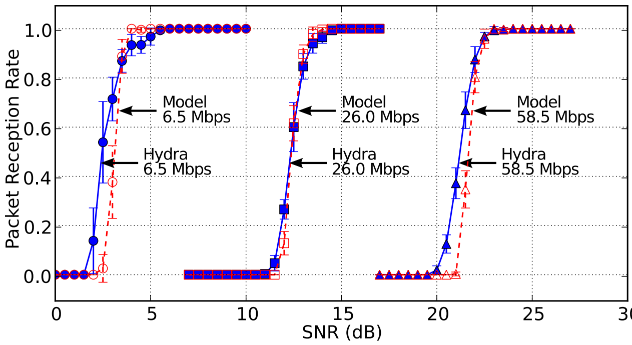

Fig. 5: Packet decoding performance in AWGN scenario.

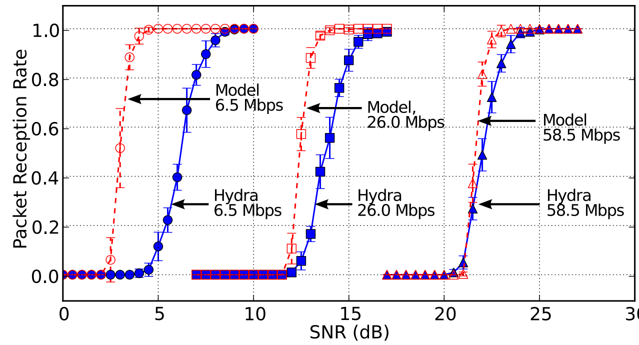

Fig. 7: Packet decoding performance with CFO.

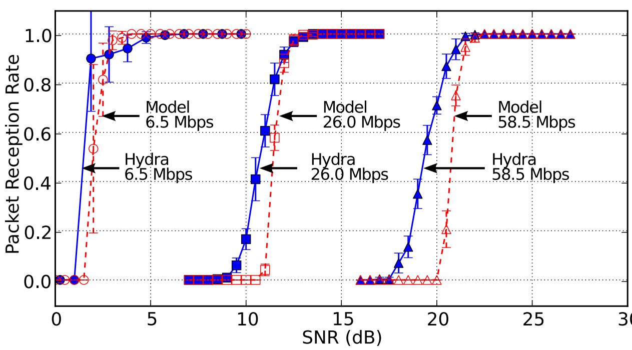

Fig. 6: Packet decoding performance with Short Packets.

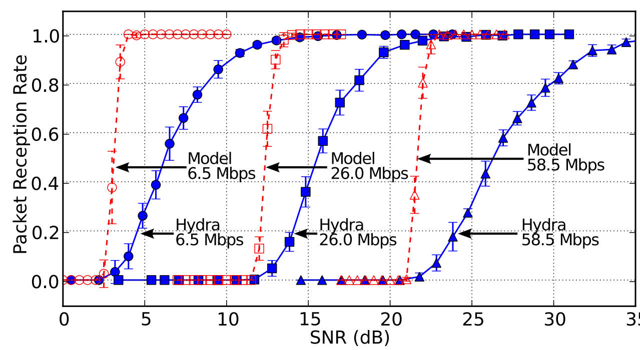

Fig. 8: Packet decoding performance in Multipath Scenario.

difference stems from the approximation the model uses for converting BER to PER in Eq. (2), which assumes that bit-errors are uniformly and identically distributed in the packet. This may be less valid with smaller packet lengths. Note that the error bars at the lowest data rate are larger in Fig. 6. This results from the loss of some transmitted data packets during detection, which reduces the effective sample size.

CFO: Figure 7 compares the decoding performance of the model against that of Hydra in the presence of carrier frequency offset. These results show that the behavior of the two systems significantly diverge, and that the impact of CFO is worse at lower SNR. This decoding behavior is a result of how noise impacts CFO compensation in Hydra. Estimates of CFO and other phase tracking parameters used in compensation become worse at lower SNR; thus, the residual CFO that remains after compensation degrades the decoding performance of the PHY, especially at low SNR. This impairment impacts the performance of lower data rates, which are normally robust enough to operate in this low SNR regime. For example, Fig. 7 shows that transmissions using the lowest data rate (6.5 Mbps) require approximately 5 dB of additional SNR to achieve reliable communication.

Multipath: The final set of link-level measurements in this section correspond to the performance of the physical layer in a frequency-selective channel. As Figure 8 shows, the packet decoding behavior of the model and Hydra diverge significantly. The most notable feature of PHY performance in this multipath channel is that, for each data rate, the transition from *receiving no packets* to *receiving most packets* ($> 95\%$) occurs over a much larger SNR range. To explain this behavior, we should consider the impact of frequency selectivity on the subcarriers of an OFDM system. In a multipath channel, the SNR for some of the subcarriers will fall below the average SNR and others will fall above it. While the average SNR of the channel might be sufficient for reliable communication in an AWGN channel, these "bad" subcarriers can introduce additional bit-errors that the decoder will have to correct. This diminishes the error-correcting ability of the physical layer.

C. Summary

Our results show that the PHY model used in WiNS (and other state-of-the-art network simulators) accurately models packet decoding under a limited set of wireless conditions. We used direct-execution to characterize link-level behavior and show how the model diverges under various realistic wireless conditions. We also showed that the detection model in WiNS is accurate under *all* of the wireless scenarios in our study. Our link-level characterization provides a better understanding of the operating regimes where the PHY model is still accurate in the presence of these wireless impairments. The impact that model-error has on the accuracy of network simulations depends on the specific topologies, protocols, and channels in a network scenario. Direct-execution provides an effective means of evaluating model accuracy in terms of its impact on network simulations.

VI. Investigating RBAR with Direct Execution

We now use direct execution to evaluate the accuracy of the PHY model from the perspective of a higher-layer protocol. Our goals in this section are twofold. First, we will show that direct execution is an effective means of evaluating the accuracy of a PHY model in terms of its impact on the performance of higher-layer protocols. Secondly, we show how this approach can identify when a PHY model is suitable for generating trustworthy simulation results under various wireless conditions. We do so by using direct execution to study the behavior of a simple rate adaptive MAC protocol, the Receiver-Based Auto Rate (RBAR) protocol [30].

A. *RBAR Protocol Description*

RBAR is a simple protocol for opportunistically adapting the data rate of transmissions in an IEEE 802.11 network. In the protocol, a receiver is responsible for recommending the "best" data rate for the sender's transmission. RBAR enables this receiver-centric adapatation by piggybacking rate information on the four-way handshake of IEEE 802.11 (i.e., the RTS-CTS-DATA-ACK exchange between a sender and receiver). We refer the reader to [30] for details on the exact operation of this modified four-way handshake.

RBAR uses a threshold-based policy to adapt the data rate used by the sender. The goal is to use the highest data rate that will maintain a reliable quality of service (QoS). Table I shows the *standard* set of RBAR thresholds implemented in WiNS. The thresholds were designed to achieve a target BER of 10^{-6} (PER $\approx 1\%$) based on the performance of the PHY in an AWGN channel (CM-A). Each threshold corresponds to the SNR needed to achieve this target QoS. Table I also shows an alternate set of thresholds that we will use later, as we try to improve the performance of RBAR and accuracy of simulations in a frequency-selective channel (CM-D).

B. *Simulation Setup*

For this investigation, we consider communication between a single sender and receiver using RBAR. We measure the MAC layer throughput of the protocol as a function of the average SNR of the link. Note that MAC layer throughput includes protocol overhead (e.g., RTS, CTS, etc.), as well as the latency associated with the backoff and retransmission policies of IEEE 802.11 [1]. The MAC protocol uses a maximum retransmission count of 7.

TABLE I: RBAR Adaptation Thresholds

Data Rate	**Standard SNR Thresholds** *(PER $\approx 1\%$ in CM-A)*	**Alternate SNR Thresholds** *(PER $\approx 5\%$ in CM-D)*
6.5 Mbps	N/A	N/A
13.0 Mbps	7.2 dB	14.8 dB
19.5 Mbps	10.1 dB	19.8 dB
26.0 Mbps	13.8 dB	22.0 dB
39.0 Mbps	16.9 dB	26.7 dB
52.0 Mbps	21.7 dB	29.2 dB
58.5 Mbps	22.9 dB	32.6 dB
65.0 Mbps	24.7 dB	36.3 dB

C. *Simulation Results*

Here we present results using direct-execution to evaluate the accuracy of our physical layer model in terms of its impact on the performance of RBAR. First, we use this approach to validate the model in a frequency-flat channel (TGn Channel Model A, CM-A). Then, we examine how PHY performance in a frequency-selective wireless channel (TGn Channel Model D, CM-D) impacts the accuracy of our simulation results. We vary the sender-receiver separation distance to achieve a particular average SNR. At each average SNR, the sender sends 100 packets, which we use to measure average throughput ($\mathrm{T_{put}}$). We use error bars to indicate 90% confidence intervals computed over 10 trials.

1) Frequency-Flat Channel: Figure 9 shows the performance of RBAR as it varies with SNR in a frequency-flat channel without time-dependent fading (i.e., instantaneous SNR is equal to average SNR). We see that as SNR increases the protocol selects higher data rates to take advantage of the improving link quality. At each SNR threshold in the *standard adaptation policy* specified by Table I, throughput performance increases in a stepwise fashion.

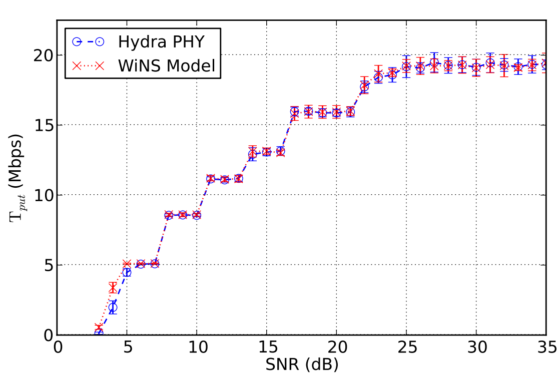

Fig. 9: RBAR throughput in CM-A *without* fading.

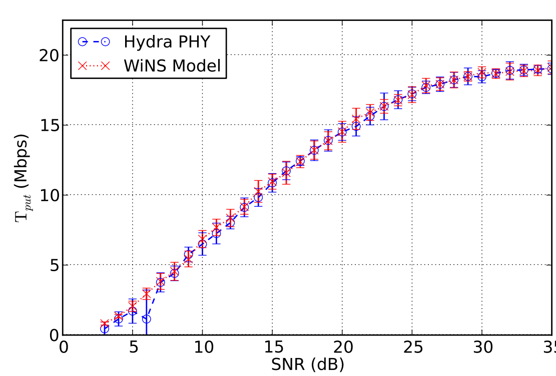

Fig. 10: RBAR throughput in CM-A *with* fading.

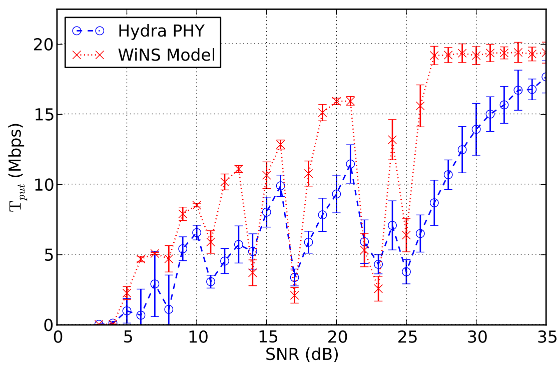

Fig. 11: RBAR throughput in CM-D *without* fading.

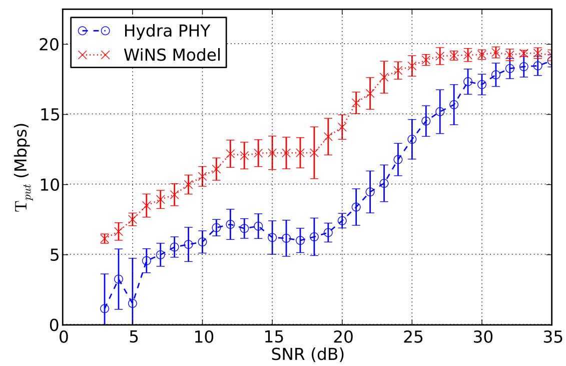

Fig. 12: RBAR throughput in CM-D *with* fading.

In Figure 10, we present the throughput performance of RBAR as it varies with the average SNR in a fading channel. For each average-SNR value, the corresponding throughput value represents throughput performance averaged over the distribution of SNRs seen by the fading channel. Note that RBAR uses instantaneous SNR to adapt the data rate for transmissions. The resulting throughput performance gradually increases as a function of average SNR.

These results confirm that the physical layer model is accurate in terms of its impact on simulations of RBAR in a frequency-flat channel with and without time-dependent fading. This is consistent with our link-level validation of the physical layer model in an AWGN channel.

2) Frequency-Selective Channel: Now we consider the impact of a frequency-selective channel (TGn Channel Model D, CM-D) on the performance of RBAR. Figure 11 illustrates the performance of RBAR in a multipath channel *without* time-dependent fading. There are two notable features of this graph. First, at each SNR threshold (or switching point), we see that RBAR prematurely selects the next higher data rate. That is, the physical layer is unable to deliver the intended target quality of service when this occurs, resulting in a dramatic loss of throughput. This accounts for the erratic way in which the throughput drops off and rises, instead of monotonically increasing. The second notable feature is the divergence between the throughput performance of RBAR using the model and RBAR using Hydra. In particular, the model is optimistic with regard to the Hydra physical layer. This is consistent with our expectations based on the link-level behavior of the two systems, shown in Figure 8.

Figure 12 presents RBAR performance in a *time-varying* frequency-selective channel (CM-D). We see that the erratic behavior of the system gets smoothed over the distribution of the SNRs for the fading channel. The graph shows that divergence between the model and Hydra still persists in the fading channel. From these results, we conclude that the PHY model is not suitable for simulating RBAR in a frequency-selective channel when using the *standard adaptation policy* in Table I. RBAR using these adaptation thresholds cannot achieve its target QoS in this frequency-selective channel.

D. Modifying the Adapation Policy

The failure of RBAR in the frequency-selective channel stems from the improperly tuned switching points in the *standard* adaptation policy in Table I. Using the results from our link-level characterization of the physical layer, we can design an adpatation policy that is appropriate for RBAR in a frequency selective channel. In this way, our direct-execution approach provides insight into the design of this rate adaptive protocol. Table I contains this alternative set of SNR thresholds, which are designed to achieve a target packet-error rate of 5% in a frequency-selective channel.

Using these more conservative SNR thresholds we once again measure the throughput performance of RBAR in a frequency-selective channel. Figure 13 shows the behavior of RBAR using this adaptation policy in a frequency-selective channel (CM-D) without fading. These results show that RBAR is able now to operate as intended, where throughput performance increases in a stepwise fashion. We note that at the lowest data rate, there is still some divergence between the model and Hydra, which is consistent with our link-level characterization of the PHY. Figure 14 shows the throughput performance of RBAR using the alternate adaptation policy when we use a fading multipath channel.

E. Summary

The results presented in this section demonstrate that direct-execution is an effective means of studying model accuracy in terms of its impact on higher-layer protocols. Our results validated the PHY model in terms of its accuracy in simulations of RBAR over a fading frequency-flat channel. We used direct-execution to study the impact of model accuracy on simulations of RBAR, and used insight from our link-level characterization to tune the adaptation policies of the protocol. Modifying these policies allowed RBAR to operate in a frequency-selective channel as it was intended. In “correcting” RBAR, we also caused the physical layer to operate in a regime where the PHY model was more accurate. As a result, we can now use the PHY model to generate trustworthy results when simulating RBAR in both frequency-flat and frequency-selective channels.

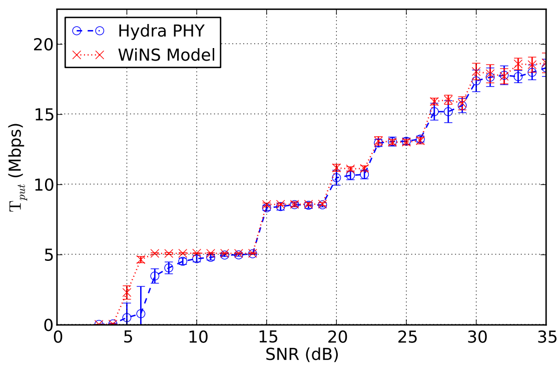

Fig. 13: $\mathrm{T_{put}}$ with Alternative RBAR Thresholds (*no* fading).

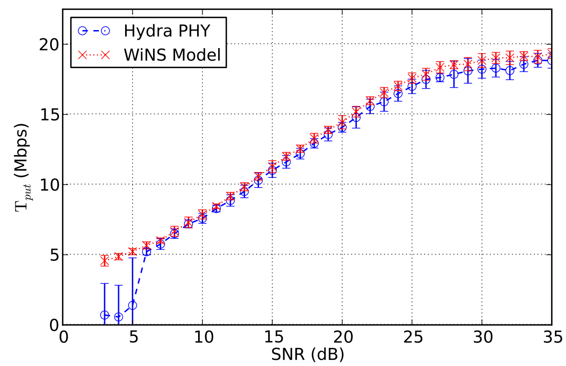

Fig. 14: $\mathrm{T_{put}}$ with Alternative RBAR Thresholds (*with* fading).

VII. Related Work

There have been various efforts to develop a better understanding of the link-level behavior of the PHY and improve the accuracy of network simulation. Our direct-execution approach is comparable to these related efforts. Here we focus on two of the most closely related techniques: physical layer emulation and hybrid-simulation.

Judd and Steenkiste presented a study of the link-level behavior of an IEEE 802.11 system using physical layer emulation [8], [31]. This technique connects real wireless devices over a high-performance wireless channel emulator (the CMU Netwok Emulator), implemented on a DSP-and-FPGA-based platform. This emulation approach provides a means of addressing one of the main drawbacks of testbeds, that is their inability to reliably generate repeatable experimental results. Physical layer emulation is an alternative to deploying testbeds in real-world environments that provides a high degree of control and repeatability. In the same way, our direct-execution approach is an alternative to model validation using measurements from testbeds deployed in real-world environments. It also provides a high degree of control and repeatability, and maintains the credibility associated with using a real physical layer implementation.

The most similar approach to direct-execution is hybrid-simulation. This is a technique for enabling highly accurate simulation by replacing the packet-level models of the physical layer and wireless channel in a simulator with detailed waveform-level implementations. The main drawback of this technique is its computational complexity. As Mittag, et al. showed, using a waveform-level PHY implementation can increase the runtime of simulations by a factor of 100 or more over their packet-level counterparts [32]. This is further exacerbated when using realistic detailed fading models of the wireless channel, increasing complexity by an additional factor of 100 or more. While some complexity reducing approaches have been proposed, this technique remains vastly more computationally expensive than simulations using packet-level models [32]–[34]. As such, we believe this motivates the need for continued development of efficient *and* accurate models of the physical layer.

A key distinction between direct-execution and hybrid-simulation is that our approach draws its physical layer implementation from an operational SDR testbed. In this way, we can credibly claim that the operation of the physical layer accurately reflects reality. In addition, unlike hybrid-simulation, the goal of direct-execution is not to supplant the packet-level models of the physical layer and wireless channel. Rather, direct-execution provides an effective means of identifying when such models are accurate and can be used to generate trustworthy simulation results.

VIII. Conclusion

Physical layer inaccuracy in network simulation has been observed to produce misleading results, which contributes to doubts about the overall credibility of simulation. It is important to address this issue by validating and improving the accuracy of models used in simulation. In this paper, we demonstrated a technique for validating the physical layer reception models in simulators using direct execution of a real PHY implementation from an operational SDR testbed. This approach leverages the inherent credibility of the testbed, while maintaing the control, scalability, and repeatability associated with simulation. In doing so, it allows us to study model accuracy from the perspective of different protocol layers using a broad set of wireless scenarios.

We used direct-execution to characterize the link-level behavior of a leading OFDM PHY model used in state-of-the-art network simulators. Our results showed that the model is accurate under a limited set of wireless conditions, and accounted for inaccuracies under various realistic wireless impairments. We also used direct-execution to study the PHY model in terms of its impact on simulations of a rate adaptive MAC protocol (RBAR). We showed how modifying the policies of a higher-layer protocol can change the impact that a model has on the accuracy of network simulations.

Direct-execution is an effective means of evaluating the accuracy of a PHY model in a flexible and controlled manner. Further, this approach is an effective means of identifying when a PHY model is suitable for producing trustworthy simulation results, even in scenarios where the model is not accurate from the perspective of the physical layer.

A potential direction for extending this study in future work might include the use of direct-execution to validate multi-antenna PHY models used to study IEEE 802.11n networks. The framework developed for our specific implementation of direct-execution, using Hydra and WiNS, would be well-suited to this task, as the channel models in WiNS and physical layer of Hydra are already designed for multi-antenna operation. Finally, we note that code for WiNS is publicly available for download [24].

REFERENCES

[1] IEEE Std 802.11n, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," pp. c1 –502, Oct. 2009.

[2] ns-2, "The Network Simulator," http://www.monarch.cs.rice.edu/cmu-ns.html.

[3] OPNET Technologies, Inc., "OPNET Modeller," http://www.opnet.com/solutions/network_rd/modeler.html.

[4] M. Takai, J. Martin, and R. Bagrodia, "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: ACM, 2001, pp. 87–94.

[5] C. Newport, D. Kotz, Y. Yuan, R. S. Gray, J. Liu, and C. Elliott, "Experimental Evaluation of Wireless Simulation Assumptions," *Simulation*, vol. 83, no. 9, pp. 643–661, 2007.

[6] R. Bagrodia and M. Takai, "Position Paper on Validation of Network Simulation Models," in *DARPA/NIST Network Simulation Validation Workshop*, 1999.

[7] D. Johnson, "Validation of Wireless and Mobile Network Models and Simulation," in *DARPA/NIST Network Simulation Validation Workshop*, 1999.

[8] G. Judd and P. Steenkiste, "Repeatable and Realistic Wireless Experimentation Through Physical Emulation," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 63–68, 2004.

[9] T. Bingmann, "Accuracy Enhancements of the 802.11 Model and EDCA QoS Extensions in ns-3," Ph.D. dissertation, University of Karlsruhe (TH), 2009.

[10] A. Wang and Z. Liu, "Analysis and Utilizing of the Error Models in Network Education with NS3," in *Second International Workshop on Education Technology and Computer Science (ETCS)*, vol. 1, March 2010, pp. 15 –18.

[11] R. G. Sargent, "Validation and Verification of Simulation Models," in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, vol. 1, Dec. 2004, pp. 2 vol. (xliv+2164).

[12] M. Bredel and M. Bergner, "On the Accuracy of IEEE 802.11g Wireless LAN Simulations Using OMNet++," in *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 1–5.

[13] K. Borries, X. Wang, G. Judd, P. Steenkiste, and D. Stancil, "Experience with a Wireless Network Testbed Based on Signal Propagation Emulation," in *2010 European Wireless Conference (EW)*, April 2010, pp. 833 –840.

[14] U. M. Colesanti, C. Crociani, and A. Vitaletti, "On the Accuracy of OMNeT++ in the Wireless Sensor Networks Domain: Simulation vs. Testbed," in *PE-WASUN '07: Proceedings of the 4th ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*. New York, NY, USA: ACM, 2007, pp. 25–31.

[15] S. Ivanov, A. Herms, and G. Lukas, "Experimental Validation of the ns-2 Wireless Model using Simulation, Emulation, and Real Network," in *Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN 2007)*, Bern, Switzerland, 2 2007, pp. 433–444.

[16] K. Mandke, S.-H. Choi, G. Kim, R. Grant, R. Daniels, W. Kim, R. Heath, and S. Nettles, "Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, Apr. 2007, pp. 1896 –1900.

[17] R. C. Daniels, K. Mandke, S. W. Peters, S. M. Nettles, and R. W. Heath, Jr., "Machine Learning for Physical Layer Link Adaptation in Multiple-Antenna Wireless Networks," in *Proceedings of the Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, ser. WiNTECH '08. San Francisco, California, USA: ACM, 2008, pp. 113–114.

[18] R. C. Daniels, K. Mandke, K. T. Truong, S. M. Nettles, and R. W. Heath, Jr., "Throughput/Delay Measurements of Limited Feedback Beamforming in Indoor Wireless Networks," in *Proceedings of the IEEE Global Communications Conference*. IEEE, 2008.

[19] W. Kim, O. Khan, K. Truong, S.-H. Choi, R. Grant, H. Wright, K. Mandke, R. Daniels, R. Heath, and S. Nettles, "An Experimental Evaluation of Rate Adaptation for Multi-Antenna Systems," in *IEEE INFOCOM*, Apr. 2009, pp. 2313 –2321.

[20] W. S. Kim, "Improving the Performance of Wireless Networks using Frame Aggregation and Rate Adaptation," Ph.D. dissertation, University of Texas at Austin, 2010.

[21] T. Schmidl and D. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613 –1621, Dec. 1997.

[22] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[23] B. Sklar, *Digital Communications: Fundamentals and Applications (2nd Edition) (Prentice Hall Communications Engineering and Emerging Technologies Series)*, 2nd ed. Prentice Hall PTR, January 2001.

[24] K. Mandke, "WiNS: Wireless Network Simulator," Website [Last accessed: May 11, 2011], http://netlab.ece.utexas.edu/mandke/wins/.

[25] D. Qiao, S. Choi, and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 1, no. 4, pp. 278–292, 2002.

[26] K. Mandke, "Validating Wireless Network Simulations Using Direct Execution," Ph.D. dissertation, The University of Texas at Austin (UT), 2012.

[27] T. Paul and T. Ogunfunmi, "Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment," *IEEE Circuits and Systems Magazine*, vol. 8, no. 1, pp. 28 –54, First Quarter 2008.

[28] R. Sinha, C. Papadopoulos, and J. Heidemann, "Internet Packet Size Distributions: Some Observations," University of Southern California, Tech. Rep. ISI-TR-2007-643, May 2007.

[29] E. Perahia, et al., "Joint Proposal Team PHY Simulation Results," *IEEE802.11-06/0067r02*, January 2006.

[30] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. Rome, Italy: ACM, 2001, pp. 236–251.

[31] G. Judd and P. Steenkiste, "Understanding Link-Level 802.11 Behavior: Replacing Convention with Measurement," in *WICON '07: Proceedings of the 3rd International Conference on Wireless Internet*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 1–10.

[32] J. Mittag, S. Papanastasiou, H. Hartenstein, and E. Strom, "Enabling Accurate Cross-Layer PHY/MAC/NET Simulation Studies of Vehicular Communication Networks," in *Proceedings of the IEEE*, vol. 99, no. 7, July 2011, pp. 1311 –1326.

[33] G. Yeung, M. Takai, and R. Bagrodia, "Effects of Detailed OFDM Modeling in Network Simulation," *SIMULATION*, vol. 81, no. 4, pp. 241–253, 2005.

[34] J. Farooq and T. Turletti, "An IEEE 802.16 WiMAX module for the NS-3 Simulator," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09, Rome, Italy, 2009, pp. 8:1–8:11.