Experimental Validation of Analytical Performance Models for IEEE 802.11 Networks

Shrirang Mare
Computer Science
Dartmouth College
Hanover, NH, USA

David Kotz
Computer Science; ISTS
Dartmouth College
Hanover, NH, USA

Anurag Kumar
ECE Department
Indian Institute of Science
Bangalore, India

***Abstract*—We consider the simplest IEEE 802.11 WLAN networks for which analytical models are available and seek to provide an experimental validation of these models. Our experiments include the following cases: (i) two nodes with saturated queues, sending fixed-length UDP packets to each other, and (ii) a TCP-controlled transfer between two nodes. Our experiments are based entirely on Aruba AP-70 access points operating under Linux. We report our observations on certain non-standard behavior of the devices. In cases where the devices adhere to the standards, we find that the results from the analytical models estimate the experimental data with a mean error of 3-5%.**

I. Introduction

Wireless access networks based on the IEEE 802.11 suite of standards have become extremely popular as they offer inexpensive, high-speed and ubiquitous access to the Internet and to networked enterprise services. In spite of the large installed base of these networks, systematic techniques for engineering these networks still remain elusive. Since the publication of Bianchi's seminal paper [1] in 2000, there has been considerable progress in analytical models of IEEE 802.11 networks [2], [3], [4], [5], [6], [7]. Such models could help to provide a rough comparison of alternate designs, leading to a narrowing down of possibilities, which could then be studied by detailed simulation, or even deployment. Further, after WLANs are deployed, they are often instrumented to yield packet traces, from which several statistics can be derived [8], [9], [10], [11]. It is a challenge, however, to make inferences from these statistics. Analytical models can also be useful in making such inferences. For modeling-based approaches to be effective, it is important to understand how well the analytical models are able to capture the performance as seen in practice. With this in mind, in this paper we provide some preliminary results on a comparison of analytical model predictions with experimental performance measurements on simple IEEE 802.11 networks.

Important to any validation effort is a good approach for comparing the model estimates against the experiment results. Johnson [12] first suggested the use of log information for comparison, in addition to the end-to-end information, to get better validation results. There are earlier studies on the validation of analytical models and simulations against experimental testbeds [13], [14], [15], [16]. Ivanov et al. [15] provide an experimental validation of a wireless model in ns-2 against a real network to show how well a model in ns-2 represents a real wireless network. Angelakis et al. [14] validate channel interference models in 802.11a, but they use an emulated wireless medium to avoid any non-deterministic interference. The validation effort by Aziz et al. [13] is similar, in some aspects, to what we present in this paper. They validate their model, which suggests a change in the standard backoff behavior and retry limits to improve the throughput of wireless mesh networks. They, however, use end-to-end throughput measurements to validate their model. They also do not consider channel errors, which we observed to be significant enough to affect the results.

In this paper, we focus on contention models in single-hop wireless networks, especially for 802.11b. In addition to the driver statistics, we use detailed log information from two sniffers for measurements and also to analyze the network behavior. We also provide and validate an analytical model for a single-flow TCP connection and validate it on a two-node network. In a network with more than two nodes, the capture effect plays a significant role and affects the fairness and hence the expected failure rates in the network. So we could not validate the model for more than two nodes.

Carrying out controlled experiments on WLANs using off-the-shelf products in a cluttered RF environment is a major challenge. Furthermore, Bianchi et al. [17] show in an experimental assessment of commercial wireless cards that the commercially available wireless cards often do not comply with the IEEE 802.11 standards. Non-standard behavior makes it even more difficult to set up the validation experiments.

This paper makes following contributions. We describe the challenges we faced in deploying an experimental infrastructure for model validation; we hope that these experiences will be useful to other researchers. The published models of TCP-controlled file transfers do not model packet loss probability; we provide an extension to the analysis by including packet loss probability. Finally, we validate two models and show that the estimates from the analytical models are within 3-5% of the experimentally measured performance; in doing so, we

Part of this work this work was done while David Kotz and Shrirang Mare were at IISc, India.

978-1-4244-5489-1/10/ $26.00

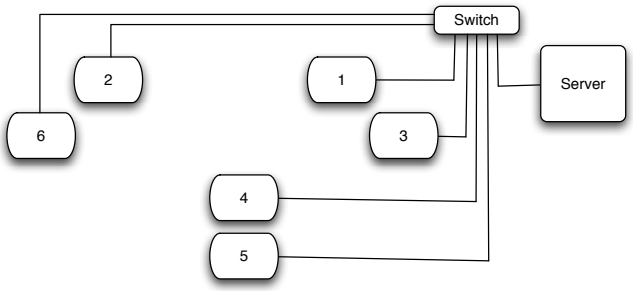

Fig. 1. Indoor Test-bed setup with six Aruba AP-70 nodes and a server connected with a private Ethernet network.

describe the non-standard behavior of wireless devices and how we accommodated that behavior in verifying the model.

The remainder of the paper is organized as follows. Section II describes our experimental methodology. The two-node UDP case is described in Section III. We describe the non-standard behavior of commercial cards in Section IV. We present the two-node TCP case in Section V. We discuss some possible applications of these models in real networks in Section VI, followed by our conclusions.

II. Experimental Methodology

Our indoor testbed consists of six Aruba AP-70 [18] access points flashed with OpenWrt [19] Linux, which then become our general-purpose Wi-Fi experimental platform. Each node uses the Atheros 5212 chipset and MadWifi driver version 0.9.4.5 (svn r2568). AP-70s can be powered over Ethernet (PoE), which makes it convenient to build testbeds. All the six nodes were connected to a switch by PoE cables. These six nodes (numbered $1, 2, 3, 4, 5, 6$) were placed in pairs as shown in Figure 1. All the nodes were close (within 3 meters) and hence they form a single cell. To avoid any possible interference due to foreign traffic, we performed our experiments in the closed environment of our laboratory[1] on channel 11. The campus Wi-Fi production access points near our laboratory only used channels 1 to 6; with a 5 channel isolation, interference was minimal. Nonetheless, to ensure a clean experiment, we checked the medium using a sniffer before each experiment to verify that there was no significant foreign traffic in the medium.

Each Aruba AP-70 had two wireless interfaces, but the AP-70 CPU was not fast enough to sniff on both interfaces at once; we used only one wireless interface and disabled the other. A server machine, also connected to the switch, executed the experiments using scripts that set the nodes in ad-hoc or monitor mode (as required) captured the traffic from sniffer nodes, started the traffic on other nodes, and stored all experimental data on the server. We controlled and monitored the experiment through an isolated Ethernet network.

[1]Experiments were performed in ECE Department, IISc, India.

TABLE I
Ratio of analytical results to experimental results - before and after considering PER in analysis.

Experiment	Before considering PER	After considering PER
UDP case	9.28 ($\pm$1.29)	4.12 ($\pm$0.91)
TCP case	9.95 ($\pm$2.29)	4.39 ($\pm$1.17)

For the experiments, any two nodes were picked from two different pairs, to form an 802.11b ad-hoc network. We fixed the transmission rate at 11Mbps, used only the basic access mechanism (i.e., no RTS-CTS), and we used Iperf software[2] to generate traffic. The nodes adjacent to the two selected nodes were set in monitor mode so that they could act as sniffers and capture the traffic of the adjacent node. For instance, if node pair 1-2 was picked for the experiment, then nodes 1 and 2 would form an ad-hoc network and nodes 3 and 6 would act as sniffers for nodes 1 and 2 respectively. We confirmed that the sniffer next to a node would capture most[3] of the transmissions from (and to) that node.

Early in our experiments we realized that even when nodes are close to each other, the packet error probabilities are sufficiently high to affect the analytical results. Table I shows the effect of packet errors on the analytical results. It displays the ratio of analytical results to experimental results – before and after considering packet errors in analysis – for the two cases that we discuss in this paper. Hence, it was important to measure these error probabilities and use the version of the analysis that incorporates error probabilities.

The **probability of channel error** (p_e) depends on the path between two nodes and is asymmetric. In wireless networks, path reception symmetry is a commonly mistaken axiom [20]. Channel errors for a link can be different from that of the reverse link, i.e., p_e for path 1 to 2 (node 1 to node 2) will be different than that for path 2 to 1. To measure p_e for a path, say 1 to 2, we transmitted about 16,000 frames from node 1 to 2 and measured the total number of attempts (including retries) made by node 1, using the sniffer[4] placed close to node 1. From these measured values we could calculate the probability of channel error as

$$p_e = \frac{\text{number of retries}}{\text{total number of attempts}} \tag{1}$$

The error probability for path 2 to 1 could be different, so we calculated it separately using the above described method. Since the channel error probability changes with time and environment, we also calculated new p_e values for each run, just before the run begins.

[2]http://sourceforge.net/projects/iperf

[3]A comparison between sniffer traces and driver statistics revealed that the sniffer captured more than 99% of the traffic through (to and from) a node when placed close that node.

[4]p_e can also be measured using driver statistics at sender node or by comparing the number of packets sent by sender and received at receiver. For the latter approach, however, we have to gather data from two nodes as compared to other approaches where data from one node (either sniffer or sender) is sufficient.

The **failure probability** (γ) includes both collision probability and channel error probability. It can be measured either with statistics from the driver[5] or by examining traces from sniffers. For our experiments, we measured values using driver statistics and sniffer traces; we observed that the difference in the values was negligible, mainly because we placed sniffers close to the node we wanted to monitor. The results presented in this paper, however, are from driver statistics. Using either method, we can obtain the number of retries and the total number of attempts made by a node. Then the failure probability can be calculated as

$$\gamma = \frac{\text{number of retries}}{\text{total number of attempts}} \tag{2}$$

Although this formula appears identical to Equation (1), the experimental conditions differ. During channel error probability measurement (Equation 1) only one node transmits data, so there is no contention and no collisions. All the failed attempts by the transmitting node can be attributed to channel errors. During the actual experiment, on the other hand, both nodes contend to transmit data to each other, so the measured failure probability (Equation 2) includes failures due to collision as well as channel errors.

Although the AP-70s have the same Atheros chipset, it cannot be said with certainty that all nodes will display the same behavior. Some minor differences do exist even among nodes with the same hardware. We tried to average out these differences, if any, by using different pairs of nodes for the experiments. Using different pairs, also to some extent, averages across location. In addition, these experiments were performed during different times to get an average across time.

III. Two-node UDP case

In the most basic case analysis of 802.11b, there are two nodes sending UDP packets to each other such that their transmission queues are full.

A. Analysis

For this case we use a previously developed model by Kumar et al. [21]. Consider a network of IEEE 802.11 compliant nodes, close together, such that only one transmission can be sustained at any point of time. Such networks are called *single-cell networks.* We assume that all nodes use the same backoff parameters and call this the homogeneous case. We use the mean backoff of 16 for all nodes.[6]

The network alternates between periods during which the nodes are "backing off" (i.e., their backoff timers are running), and periods during which there is some activity on the channel (a transmission or a collision). The approximation proposed by Bianchi [1] is to assume that during the backoff periods, the probability that a node attempts in a slot is a constant β, and that the nodes attempt with this probability independently in each slot. γ_c denotes the probability that a node's attempt collides. The backoff process of a node is characterized (see Kumar [21]) by the mean backoff duration before each attempt, and the maximum number of attempts that can be made for a packet. We define the following notation:

$K :=$ the maximum number of retries; at the $(K+1)^{th}$ attempt, either the packet succeeds or is discarded.
$b_k :=$ the mean backoff duration (in slots) at the k^{th} attempt for a packet, $0 \leq k \leq K$.
$\gamma_c :=$ Pr(an attempt made by a node fails due to collision).
$\beta :=$ Pr(a node attempts in a given slot).

The nodes in the network always have data to send, that is, their transmission queues are saturated. In an n node single-cell network, the probability that an attempt by a node succeeds in a given slot is the same as the probability that $n-1$ nodes do not attempt in that slot, i.e.,

$$Pr(\text{success}) = (1-\beta)^{n-1}$$

From this, the equation for γ_c can be written as

$$\gamma_c = 1 - (1-\beta)^{(n-1)} \tag{3}$$

A node takes backoffs with a larger mean value as it encounters more collisions. It has been shown in Kumar [21] that for a given γ_c the attempt rate of a node per slot is given by

$$\beta = G(\gamma_c) = \frac{1 + \gamma_c + \gamma_c^2 + \cdots + \gamma_c^K}{b_0 + \gamma_c b_1 + \gamma_c^2 b_2 + \cdots + \gamma_c^K b_K} \tag{4}$$

Substituting β in Equation (3), we get the fixed point equation

$$\gamma_c = 1 - (1 - G(\gamma_c))^{n-1}$$

Solving this fixed point equation gives us the collision probability.

Three main assumptions were made in this model: (1) the channel is error free, so all failures are due to collisions; (2) the system is homogeneous, i.e., all nodes have equal backoff parameters; (3) the transmission queues of all nodes are saturated.

In practice, however, the channel is never error free. Equation 3 can be modified to account for channel errors. We now define γ to be the probability that an attempt by a node fails, i.e., either due to a collision or channel error. Let p_e be the probability that an attempt by a node fails due to channel errors. Under the assumption that channel errors are statistically independent from collisions, the failure probability, γ, for the i^{th} node can be obtained by

$$\gamma_i = 1 - \frac{1-p_{ei}}{1-\beta_i} \prod_{j=1}^{n} (1-\beta_j), \quad 1 \leq i \leq n \tag{5}$$

For the two-node case, the above equation simplifies to

$$\begin{aligned} \gamma_1 &= 1 - (1-p_{e1})(1-\beta_2) \\ \gamma_2 &= 1 - (1-p_{e2})(1-\beta_1) \end{aligned} \tag{6}$$

Note that p_{e1} and p_{e2} refer to the channel error probabilities as observed by nodes 1 and 2, respectively, and they could be different. Using experimentally measured p_e values and

[5]Driver statistics are usually considered accurate since they do not involve any measurement from the medium.

[6]Mean backoff of 16 matches with the observed backoff that the nodes use, which is in the interval 1-32.

TABLE II
TERMS USED IN REST OF THE TABLES AND THEIR MEANINGS

Term	Meaning
PER	Packet Error Rate
PER-(1,S,R)	PER for node 1 or Sender or Receiver
FP	observed Failure Probability
FP-(1,S,R)	FP for node 1 or Sender or Receiver
Err-(1,S,R)	Normalized error (in %) for node 1 or Sender or Receiver
CI	Confidence interval (95%)

Equation (6) we calculate the collision probabilities of the two nodes in this two-node UDP case, and then compare these probabilities with experimentally measured values to validate the models. We present the results in the next section.

B. Experiment Results

To see how well the model matches with the experimental results, it is important to set up an experiment that meets the model's assumptions well. In this experiment, both the remaining assumptions – saturated transmission queues and homogeneous system – were satisfied. All the nodes in this experiment had the same hardware and even the same driver; hence we call the system "homogeneous". We confirmed that queues were saturated by examining driver statistics after each run.

For this two-node UDP case, we selected two nodes and formed an ad-hoc network, and set their adjacent nodes in the monitor mode. During each run (60 seconds) the two nodes continuously transmitted 1500 byte UDP frames to each other; our software ensured that the transmission queues of the two nodes were saturated throughout the experiment. For each pair, we gathered data over 5 runs and computed the average value.

Table II explains the meaning of the terms used in the tables throughout this paper. The experiment results for this case are listed in Table III. The first column shows the node pair used for that particular experiment run. The values for each pair are an average of 5 runs. The difference in pair 1-2 and 2-1 is who starts the data transfer first. Even though both nodes' transfers are initiated by a script, it is not possible to start both transfers at exactly the same time and one of the nodes might have a head start. So for the pair 1-2, the script starts first node 1 and then node 2; for pair 2-1, node 2 starts transmitting first. According to the standard this effect should even out with time. To average out the differences, if any, we choose to identify such pairs as different. Note that even though the observed packet error rates seem low ($0.1\% - 0.4\%$), they are high enough to affect the analytical results, as shown in Table I.

As we can see from the results, the average error (normalized difference between model prediction and observed experimental value) across different node pairs is about 4%, which is an acceptable variation from models.

TABLE III
COMPARISON BETWEEN MODEL ESTIMATES AND EXPERIMENTALLY MEASURED VALUES FOR THE TWO-NODE UDP CASE, WITH 95% CONFIDENCE INTERVAL

Pairs	PER-1	PER-2	FP-1	FP-2	Err-1	Err-2
1-2	0.0014	0.0045	0.0618	0.0651	3.46	3.46
1-4	0.0057	0.0042	0.0664	0.0663	4.09	6.22
1-6	0.0043	0.0049	0.0654	0.0658	4.51	4.30
1-5	0.0040	0.0047	0.0653	0.0662	4.77	5.01
2-1	0.0044	0.0022	0.0619	0.0590	1.27	2.39
2-4	0.0054	0.0046	0.0676	0.0660	6.19	5.08
2-3	0.0040	0.0019	0.0667	0.0645	6.49	6.48
2-5	0.0043	0.0035	0.0657	0.0637	4.81	3.21
4-1	0.0035	0.0050	0.0663	0.0670	7.10	5.77
4-2	0.0038	0.0061	0.0659	0.0675	6.16	4.88
4-6	0.0032	0.0020	0.0624	0.0610	1.40	1.17
4-3	0.0068	0.0052	0.0675	0.0668	4.06	5.61
6-1	0.0044	0.0044	0.0646	0.0652	3.22	4.11
6-4	0.0024	0.0032	0.0614	0.0624	1.22	1.38
6-3	0.0039	0.0049	0.0632	0.0639	1.70	1.39
6-5	0.0026	0.0035	0.0618	0.0627	1.52	1.42
3-2	0.0019	0.0042	0.0643	0.0664	6.27	6.00
3-4	0.0053	0.0076	0.0673	0.0686	6.03	4.43
3-6	0.0041	0.0040	0.0636	0.0627	2.12	0.76
3-5	0.0053	0.0035	0.0660	0.0655	3.88	5.88
5-1	0.0043	0.0040	0.0661	0.0655	5.58	5.20
5-2	0.0025	0.0044	0.0629	0.0641	3.38	2.34
5-6	0.0039	0.0033	0.0615	0.0604	1.02	1.70
5-3	0.0039	0.0061	0.0679	0.0693	8.67	7.41
Avg	0.0040	0.0043	0.0647	0.0648	4.12	3.98
CI	0.0005	0.0006	0.0009	0.0011	0.91	0.84

IV. PROBLEMS WITH COMMERCIAL CARDS

Since our models are based on the 802.11 standard [22] it is important that the commercials cards used for the experiments also strictly adhere to the standards, if we hope to compare models and experiments. Unfortunately, there is no authority to certify whether cards comply with the standards. Wi-Fi, which is a trademark of the Wi-Fi Alliance for certifying products based on the IEEE 802.11 standard, only warrants interoperability between different wireless devices. It does not warranty that the devices are following the access mechanism or fairness standards as laid down by IEEE.

A. Non-Standard Behavior

We observed three types of non-standard behavior in our testbed.

a) Intermediate transmission: In the two-node UDP case, we used equal-length packets for both nodes and observed that the number of retries made by both nodes were approximately equal, as predicted by model. But when we used packets of different lengths, we observed that the node sending smaller packets always retried more times than the node sending longer data packets, which should not be the case. When two frames of different length collide, which results in a failure, the standard recommends that both nodes should wait for the longer frame to finish its transmission and then again wait for an Extended InterFrame Space (EIFS) duration before resuming the back-off procedure. Our analysis of sniffer traces, however, suggest that in our experiments the node sending smaller packet did not wait for the longer frame to finish its transmission; it retransmitted the frame, only to get

another failure.[7] This behavior explains the higher number of retransmissions by the node sending smaller frames. Bianchi et al. [17] reported similar behavior in their experimental assessment of commercial cards.

b) Special (non-standard) features: During our initial experiments, we observed that two nodes set to 802.11b mode were transmitting 3,000 byte frames, double the maximum size in the standard. Apparently, Atheros chipsets have some special features, including *turbo mode* and *fast frame*. With these features, usually enabled by default, Atheros-based cards can aggregate frames to reduce overhead and improve performance; to use these features both cards should be Atheros cards. Since we used all Atheros-based cards, we had to explicitly turn off these features.

c) Preamble: In 802.11 mode, the receiver uses a preamble to decode the wireless signal and synchronize itself with the transmitter. The standard recommends that every wireless device should support the traditional *long preamble* $(192\mu s)$, whereas support for *short preamble* $(96\mu s)$ is optional. Most commercial cards support the short preamble because it improves throughput because of its smaller size. If any node in a network does not support short preamble, then all nodes use long preamble. But if all nodes in a network support short preamble, they collectively choose short preamble for their transmissions. Some drivers, such as the MadWifi driver we used, allow one to set the nodes to use long preamble. During our experiments we observed that when we used short preamble, there were cases when nodes sometimes initiated their transmission even when the medium was busy, thereby causing a collision. The number of such cases decreased substantially when we set the nodes to use long preamble. We suspect that the channel sensing function of the Atheros cards depends on length of preamble and it works better for long preamble compared to short preamble, for given environment.

B. Work Around

Due to the non-compliant behavior of Atheros cards we could not perform TCP experiments using Iperf, because in TCP the Data and ACK packets are of different lengths. As a work around, we developed a TCP-like tool, TCPsim, that simulates TCP's behavior – delayed ACK and dynamic window size – but which uses the same size packet for both Data and ACKs.

To evaluate TCPsim we ran a comparison test between Iperf and TCPsim on Ethernet. We set up two laptops to send data to each other, first using Iperf and then using TCPsim. In both cases we set the Data packet length at 1,480 bytes; in Iperf the TCP segment size was 1,448 bytes, 24 bytes less than the UDP payload size of 1,472 bytes to compensate for the smaller UDP header, so that the length of the Data frame in both cases is same. One of the two laptops was set to promiscuous mode; using tcpdump we measured the number of Data and ACK packets transmitted by the nodes during the 60 second runs.

[7]The node sending the smaller frame re-transmitted (about) 500 usec after the other node (sending a larger frame) began its transmission, which takes about 1300 usec.

TABLE IV
COMPARISON BETWEEN TCPSIM AND IPERF: NUMBER OF DATA AND ACK PACKETS SENT BY BOTH DURING EACH RUN.

Run	Iperf		TCPsim		Error (%)	
	Data	ACKs	Data	ACK	Data	ACK
1	488649	244886	489591	244799	0.19	0.04
2	488673	244904	492296	246151	0.74	0.51
3	488543	244832	487659	243836	0.18	0.41
4	488222	244671	488921	244487	0.14	0.08
5	488873	245031	488225	244129	0.13	0.37
6	488233	244657	492193	246282	0.81	0.66
7	488272	244752	488406	244217	0.03	0.22
8	488550	244822	493608	246814	1.04	0.81
9	488427	244791	492605	246310	0.86	0.62
10	488567	244798	493326	246781	0.97	0.81
Avg	488501	244814	490683	245381	0.5	0.45

Table IV shows the results of this experiment and we observed that the difference between TCPsim and Iperf, in terms of the number of packets, was less than 1%. The number of Data packets and ACK packets sent by TCPsim is equivalent to the number of packets sent by Iperf. So we used TCPsim, instead of Iperf, for the experiments to validate TCP models as described in next section.

V. TWO-NODE TCP CASE

In the two-node UDP case the nodes' transmissions are independent of each other. In a comparable TCP case, the nodes' transmissions are dependent on each other and bound by the TCP window size. In a single TCP download, the "sender" node transmits TCP Data packets and the "receiver" node acknowledges with TCP ACK packets.

A. Analysis

Analysis of TCP is different from the two-node UDP case due to the window driven sender-receiver interaction and the delayed ACK feature in TCP. When the delayed ACK feature is enabled, the receiver sends an ACK for every two Data frames or after a timeout, whichever is sooner. With this feature the overall throughput of TCP increases. In single-cell networks, the nodes are close enough and we can safely eliminate the timeout possibility. So for these cases, we can say the receiver sends an ACK for every two Data frames.

We develop the analytical model [23], [24] by embedding the random process of the number of ACKs queued at the TCP receiver at the ends of transmission successes on the medium. Each success can be that of a DATA packet from the sender or an ACK from the receiver. When the sender and the receiver both have packets, either could succeed with probability half. If sender succeeds then the receiver may or may not queue an additional ACK. We model this probabilistically by a probability of $1/2$ of a sender success creating an ACK at the receiver. These assumptions result in the process embedded at the ends of successes being a Markov chain with the transition probability diagram as shown in Figure 2. Thus, on a sender success the receiver could change its state with probability $1/4$. Exceptions to above rule are the initial and final states,

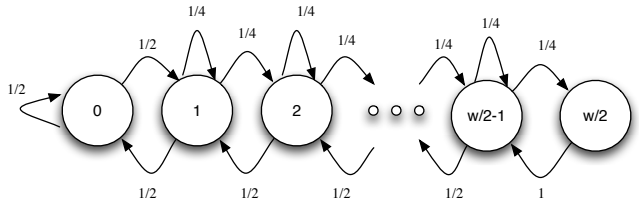

Fig. 2. The transition diagram of the Markov chain of the number of ACKs queued at the TCP receiver at the ends of successes on the wireless medium.

0 and $w/2$. When the receiver is in state 0, it has no queued ACK to transmit; when it receives Data frame the probability that it changes its state is $1/2$. When the receiver is in state $w/2$, the sender's window is full and it cannot transmit. Hence the receiver success probability is 1.

We have a finite-state irreducible Markov chain. Let $\pi_i, 0 \leq i \leq w/2$, denote the stationary probabilities of the states. Thus

$$\sum_{i=0}^{w/2} \pi_i = 1. \tag{7}$$

and, from the balance equations,

$$\begin{aligned} \pi_1 = \pi_0;\ \pi_i = \pi_0 \left(\frac{1}{2}\right)^{i-1} \quad \text{for } 1 \leq i \leq w/2\ -1; \\ \pi_{w/2} = (1/4)\pi_{w/2\ -1} \end{aligned} \tag{8}$$

Substituting π_i values in Equation 7 we get

$$\pi_0 \approx 1/3;\ \pi_{w/2} = \pi_0 \left(\frac{1}{2}\right)^{w/2} \approx 0. \tag{9}$$

Thus, we conclude that the sender finds the medium free of contention from the receiver for one third of the cycles. Hence, during such cycles the sender's transmission attempts fail only due to channel errors; we denote the error probability (p_e) observed by the sender as p_s, and that observed by the receiver as p_r. The receiver, however, will have contention from the sender for nearly every cycle. So from the receiver's point of view during these cycles the interaction between the sender and the receiver is identical to the 'two-node UDP case' discussed in Section III. During such cycles we assume that the two-node saturated model applies as developed earlier (see Kumar et al. [21]).

Let A_r and A_s denote the random number of attempts by the receiver and the sender, respectively, in a cycle, and let Φ_r and Φ_s denote the corresponding number of failures (collision or packet errors). Then by Markov regenerative theory [25] we can write

$$\gamma_s = \frac{\pi_0 E_0 \Phi_s + \pi_+ E_+ \Phi_s}{\pi_0 E_0 A_s + \pi_+ E_+ A_s} \tag{10}$$

where the subscript $+$ denotes the fact that the receiver queue is non-empty, and E_0 and E_+ denote expectation corresponding to the initial states 0 and $+$ in a cycle; replacing s with r in Equation 10 gives the failure probability for the receiver. We have used the approximation that the probability of the sender being empty is 0 (see Equation 9). In a cycle in which the receiver queue size is 0, the sender has a failed attempt only due to channel error, and with probability p_s. Hence, $E_0\Phi_s = p_s/(1-p_s)$ and $E_0 A_s = 1/(1-p_s)$, yielding

$$\gamma_s = \frac{(1-\pi_0)E_+\Phi_s + \pi_0 p_s/(1-p_s)}{(1-\pi_0)E_+A_s + \pi_0/(1-p_s)} \tag{11}$$

During a cycle in which both nodes contend, the failure probability of each node is given by the two-node model developed earlier. Let $\gamma_s^{(2)}$ and $\gamma_r^{(2)}$ denote the failure probability of the sender and the receiver, respectively, in these cycles, where the superscript denotes the number of contending nodes, two in this case.

$$\gamma_s^{(2)} = \frac{E_+\Phi_s}{E_+A_s};\ \gamma_r^{(2)} = \frac{E_+\Phi_r}{E_+A_r} \tag{12}$$

Simplifying Equation (11)

$$\gamma_s = \frac{\gamma_s^{(2)} + \frac{\pi_0 p_s}{(1-\pi_0)(1-p_s)E_+A_s}}{1 + \frac{\pi_0}{(1-\pi_0)(1-p_s)E_+A_s}} \tag{13}$$

Let β_s and β_r denote the probability of attempt by the sender and the receiver, respectively. Then E_+A_s, the expected number of attempts made by the sender in a cycle, can be expressed in the form of a recursive equation as

$$\begin{aligned} E_+A_s = \frac{\beta_s(1-\beta_r)(1-p_s)}{\beta_s(1-\beta_r) + \beta_r(1-\beta_s) + \beta_r\beta_s} \\ + \left(\frac{p_s\beta_s(1-\beta_r) + \beta_s\beta_r}{\beta_s(1-\beta_r) + \beta_r(1-\beta_s) + \beta_r\beta_s}\right)(1 + E_+A_s) \end{aligned}$$

Solving the above equation for E_+A_s we get

$$E_+A_s = \frac{\beta_s(1-\beta_r) + \beta_r\beta_s}{\beta_s(1-\beta_r)(1-p_s) + \beta_r(1-\beta_s)} \tag{14}$$

Applying the model described in Section III-A to the contention cycles, we get the following expression for $\gamma_s^{(2)}$ and $\gamma_r^{(2)}$.

$$\begin{aligned} \gamma_s^{(2)} = 1 - (1-p_s)(1-\beta_r) \\ \gamma_r^{(2)} = 1 - (1-p_r)(1-\beta_s) \end{aligned} \tag{15}$$

To validate this model, we measured the channel error probabilities, p_s and p_r, and then using Equations (13), (14), and (15) we computed failure probabilities for the sender and the receiver, which were then compared with experimentally measured failure probabilities. The comparison results are presented in next section.

B. Experiment Methodology and Results

Figure 1 shows the setup of the experiment for this case. The methodology is similar to the 'two-node UDP case' except that we used TCPsim for reasons described in Section IV-A. We again had two nodes, sender and receiver; the sender sent 1,500 byte frames to the receiver and the receiver acknowledged every two Data packets with an ACK packet. We fixed the transmission rate at 11Mbps. Each run lasted 60 seconds and we gathered data over 5 runs to compute an average.

TABLE V
TABLE SHOWING COMPARISON BETWEEN MODEL ESTIMATES AND EXPERIMENTALLY MEASURED VALUES FOR THE TWO-NODE TCP CASE, WITH 95% CONFIDENCE INTERVAL

Pairs	PER-S	PER-R	FP-S	FP-R	Err-S	Err-R
1-2	0.0032	0.0038	0.0308	0.0580	5.94	7.06
1-4	0.0044	0.0033	0.0336	0.0600	0.19	2.57
1-6	0.0037	0.0043	0.0347	0.0639	4.94	2.14
1-5	0.0043	0.0044	0.0339	0.0624	0.83	0.31
2-1	0.0039	0.0037	0.0313	0.0555	5.86	11.39
2-4	0.0058	0.0043	0.0363	0.0631	4.33	1.24
2-3	0.0040	0.0049	0.0371	0.0678	10.30	6.98
2-5	0.0032	0.0038	0.0341	0.0624	4.25	0.57
4-1	0.0035	0.0038	0.0339	0.0626	2.73	0.83
4-2	0.0038	0.0052	0.0348	0.0644	4.88	1.66
4-6	0.0032	0.0030	0.0298	0.0559	9.84	9.97
4-3	0.0035	0.0051	0.0348	0.0626	5.53	0.92
6-1	0.0055	0.0046	0.0364	0.0648	5.11	3.02
6-4	0.0028	0.0039	0.0304	0.0561	6.69	11.24
6-3	0.0040	0.0043	0.0335	0.0610	0.42	2.49
6-5	0.0040	0.0049	0.0311	0.0578	6.88	9.01
3-2	0.0055	0.0046	0.0366	0.0661	5.66	5.22
3-4	0.0048	0.0046	0.0365	0.0655	7.14	4.18
3-6	0.0046	0.0037	0.0330	0.0603	2.80	2.88
3-5	0.0039	0.0035	0.0325	0.0577	2.31	7.35
5-1	0.0042	0.0033	0.0349	0.0618	4.06	0.49
5-2	0.0034	0.0035	0.0324	0.0597	1.30	3.45
5-6	0.0038	0.0035	0.0321	0.0569	3.27	8.63
5-3	0.0039	0.0054	0.0331	0.0606	0.05	4.77
Avg	0.0040	0.0041	0.0337	0.0611	4.39	4.51
CI	0.0003	0.0003	0.0009	0.0014	1.17	1.51

Table V shows the results of this experiment for different pairs. Column 1 lists the node pairs where the first node was the sender and the second node was the receiver. From the table we can see that the observed failure probability values typically matched the model's estimate with an average error of about 4.5% error; in the worst case the error was 11.39%.

VI. APPLICATION

By abstracting away many details, analytical models can help in understanding the core phenomena that govern the performance of protocols. Analytical models can provide an initial comparison of alternate designs to narrow down the possibilities. Validation experiments add to the basis for confidence in the results generated by the models. If models are improved enough to be applicable to real networks, then inferences, as described below, could be drawn directly from them.

Packet errors: Rate adaptation algorithms, which control the transmission rate of stations, are supposed to act on packet errors and adjust transmission rates. In practice, however, it is difficult to isolate packet errors from collision errors, and hence rate control algorithms are adversely affected by contention. A centralized solution can solve this problem. The failure probabilities of the nodes in a network can be obtained, from analysis of sniffer data, and then the Equations from Section III-A and V-A allow a server to compute packet error rates. These values can then be broadcasted by APs and individual stations could adapt their rate control algorithms accordingly.

Compliance detection: Tools based on analytical formulas derived from models, can be developed that examine captured sniffer traces and identify nodes that are not in compliance with the standard. For instance, nodes may be selfish and gain access to the medium more than they should, and cause unfairness in the medium. A tool using analytical models could observe network and identify such selfish or malicious nodes. Although it may not always be possible to fix compliance issues, it can at least be helpful to diagnose performance problems observed in production networks.

Performance diagnosis: Models can be used to diagnose a network problem or problem with any individual node. Using analytical models on the data gathered from the sniffers placed near the problem area, an administrator can determine network conditions such as congestion, saturation (load) of nodes, RF interference, poor AP placement, or misconfiguration of nodes.

VII. CONCLUSION

In this work we have sought to provide an experimental validation of analytical models for the performance of IEEE 802.11 WLANs. During the course of our work, we encountered several experimental design challenges such as packet error probabilities, which are usually ignored in analytical models, but they do matter and hence need to be measured, and non-standard behavior of WLAN hardware.

We discussed how we addressed these challenges. We have presented validation results for two node networks with saturated queues carrying UDP packets, and also with a long-lived TCP transfer with delayed ACKs. We also report an extension of a published TCP-over-WLAN performance model that incorporates channel errors. We found that the models matched experimental results with a mean error in range of 3-5%.

Setting up controlled experiments with wireless networks, in order to make precise repeatable measurements at the frame level is a challenge. Our work here has been limited to a simple case with fixed rate on 802.11b and for just two nodes. As soon as more than two contending nodes are involved we have to deal with the phenomenon of packet capture, i.e., one of several "colliding" transmissions can be successfully received by its intended receiver. Setting up repeatable controlled experiments for complex scenarios that include factors such as the capture effect, clients operating at different rates (or automatically adjusting their rates), or environmental source of interference and attenuation, has proved to be hard, and we plan to take up this challenge in our future work.

ACKNOWLEDGEMENTS

Many thanks to S.V.R. Anand, for his support and help with the experiments. We also thank Malati Hegde, Pavan Kumar, Vasudev K.R., and the reviewers for their helpful comments.

This research was supported by the Society for Innovation and Development (Indian Institute of Science, Bangalore, India), by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, by an equipment

donation from Aruba Networks, by a Fulbright Faculty Fellowship, and by Dartmouth College. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any sponsor.

REFERENCES

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar 2000. DOI 10.1109/49.840210

[2] E. Lopez-Aguilera, M. Heusse, Y. Grunenberger, F. Rousseau, A. Duda, and J. Casademont, "An Asymmetric Access Point for Solving the Unfairness Problem in WLANs," *IEEE Transactions on Mobile Computing*, vol. 7, no. 10, pp. 1213–1227, 2008. DOI 10.1109/TMC.2008.44

[3] K. Ghaboosi, B. Khalaj, Y. Xiao, and M. Latva-aho, "Modeling IEEE 802.11 DCF Using Parallel Space–Time Markov Chain," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2404–2413, 2008.

[4] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 159–172, 2007. DOI 10.1109/TNET.2006.890136

[5] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2, March-3 April 2003, pp. 836–843 vol.2.

[6] Y. C. Tay and K. C. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," *Wireless Networks*, vol. 7, no. 2, pp. 159–171, March 2001. DOI 10.1023/A:1016637622896

[7] M. Carvalho and J. Garcia-Luna-Aceves, "Delay analysis of IEEE 802.11 in single-hop networks," in *Proceedings of the 11th IEEE International Conference on Network Protocols*, Nov. 2003, pp. 146–155. DOI 10.1109/ICNP.2003.1249764

[8] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the MAC-level behavior of wireless networks in the wild," *SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 75–86, Oct 2006. DOI 10.1145/1151659.1159923

[9] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: solving the puzzle of enterprise 802.11 analysis," *SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 39–50, Oct 2006. DOI 10.1145/1151659.1159920

[10] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker, "Automating cross-layer diagnosis of enterprise wireless networks," *SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 25–36, Oct 2007. DOI 10.1145/1282427.1282384

[11] R. Murty, G. Mainland, I. Rose, A. Chowdhury, A. Gosain, J. Bers, and M. Welsh, "CitySense: An Urban-Scale Wireless Sensor Network and Testbed," in *IEEE International Conference on Technologies for Homeland Security*, 2008. Available online: http://fiji.eecs.harvard.edu/node/119

[12] D. Johnson, "Validation of wireless and mobile network models and simulation," in *DARPA/NIST Network Simulation Validation Workshop*, May 1999.

[13] A. Aziz, T. Huehn, R. Karrer, and P. Thiran, "Model validation through experimental testbed: the fluid flow behavior example," in *TridentCom '08: Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–6.

[14] V. Angelakis, S. Papadakis, V. Siris, and A. Traganitis, "Adjacent channel interference in 802.11a: Modeling and testbed validation," in *Proceedings of the IEEE Radio and Wireless Symposium*, Jan. 2008, pp. 591–594. DOI 10.1109/RWS.2008.4463561

[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)*, 2007.

[16] J. Liu, Y. Yuan, D. M. Nicol, R. S. Gray, C. C. Newport, D. Kotz, and L. F. Perrone, "Empirical Validation of Wireless Models in Simulations of Ad Hoc Routing Protocols," *Simulation: Transactions of The Society for Modeling and Simulation International*, vol. 81, no. 4, pp. 307–323, April 2005. DOI 10.1177/0037549705055017

[17] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, "Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards," in *INFOCOM 2007, 26th IEEE International Conference on Computer Communications*, May 2007, pp. 1181–1189. DOI 10.1109/INFCOM.2007.141

[18] "Aruba AP-70." Available online: http://www.arubanetworks.com/products/access-points/ap-70.php

[19] "OpenWrt Linux." Available online: http://openwrt.org/

[20] C. Newport, D. Kotz, Y. Yuan, R. S. Gray, J. Liu, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," *SIMULATION: Transactions of The Society for Modeling and Simulation International*, vol. 83, no. 9, pp. 643–661, September 2007. DOI 10.1177/0037549707085632

[21] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 588–601, June 2007. DOI 10.1109/TNET.2007.893091

[22] *IEEE Standard for Information Technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society, IEEE 3 Park Avenue NY USA, June 2007.

[23] R. Bruno, M. Conti, and E. Gregori, "Performance modelling and measurements of TCP transfer throughput in 802.11-based WLAN," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems (MSWiM)*. ACM, 2006, pp. 4–11. DOI 10.1145/1164717.1164721

[24] A. K. George Kuriakose, Sri Harsha and V. Sharma, "Analytical models for capacity estimation of IEEE 802.11 WLANs using DCF for internet applications," *Wireless Networks*, vol. 15, no. 2, pp. 259–277, February 2009. DOI 10.1007/s11276-007-0051-8

[25] V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*. Chapman and Hall/ CRC Press, 1996.