A TRACE-BASED APPROACH FOR MODELING WIRELESS CHANNEL BEHAVIOR

Giao T. Nguyen
gnguyen@cs.berkeley.edu
University of California at Berkeley

Randy H. Katz
randy@cs.berkeley.edu
University of California at Berkeley

Brian Noble
bnoble@cs.cmu.edu
Carnegie Mellon University

Mahadev Satyanarayanan
satya@cs.cmu.edu
Carnegie Mellon University

ABSTRACT

The loss behavior of wireless networks has become the focus of many recent research efforts. Although it is generally agreed that wireless communications experience higher error rates than wireline, the nature of these lossy links is not fully understood. This paper describes an effort to characterize the loss behavior of the AT&T WaveLAN, a popular in-building wireless interface. Using a trace-based approach, packet loss information is recorded, analyzed, and validated. Our results indicate that WaveLAN experiences an average packet error rate of 2 to 3 percent. Further analysis reveals that these errors are not independent, making it hard to model them with a simple two-state Markov chain. We derive another model based on the distributions of the error and error-free length of the packet streams. For validation, we modulate both the error models and the traces in a simulator. Simulations based on traces yield an average TCP throughput of about 5 percent less than simulations based our best error model.

1. INTRODUCTION

With the proliferation of portable computers and wireless networks in recent years, many researchers have focused on designing better mobile systems. This has led to a growing interest in characterizing the loss behavior of many wireless technologies, including wireless LANs and packet radio networks. While it is well known that wireless links have higher error rates than their wired counterparts, the detailed characteristics of wireless errors are not well understood due to their inherent dependence on complex radio wave propagation. Characterizing the loss behavior is an important problem since it is one of the few key parameters that affect all levels of the network stack.

Due to the large number of protocols at different network layers, it is often infeasible to build and measure all of them. A better alternative is to trace the behaviors of existing implementations and apply the lessons learned to evaluate new designs. From wireless network traces (Noble, Nguyen, Satyanarayanan, and Katz 1996), we can improve our understanding of the wireless channel behavior, develop realistic models, and validate them. Traces also provide us a way to reproduce realistic network and mobile environments for comparing protocols and algorithms.

The focus of this paper is on the tracing and modeling of wireless channel errors. Although the uniform bit error rate model has often been used to simulate lossy links, it is inadequate to capture our measurements of in-building wireless errors. Our goal is to produce a realistic model that is not too difficult to implement. The resulting model contains several mathematical expressions for the error and error-free length extracted from many packet error traces. Validation of this model involves comparing the simulated TCP throughput with other models and traces.

The remainder of this paper is organized as follows. In section 2, we provide some background on wireless errors and the testbed for this study. Our trace-based approach consists of three phases, described in three separate sections along with the associated data. Section 3 describes the trace collection and presents the packet error rate data. In section 4, the errors are further analyzed and modeled. Finally, we discuss the validity of our models in section 5.

2. BACKGROUND

Wireless errors are mainly caused by the inability of the receiver to distinguish the transmitted signal from the background noise. The radio propagation patterns and their effect on receiver signal-to-noise ratio (SNR) have been studied for both the in-building and outdoor environments (Andersen, Rappaport, and Yeshiva 1995), (Cox and Leck 1975), (Eckhardt and Steenkiste 1996). The detrimental effect of wireless errors on reliable transport protocols has resulted in many efforts to improve the performance of TCP (Bakre and Badrinath 1995), (Balakrishnan, Seshan, and Katz 1995).

Since wireless errors highly depend on the environment and the network device, characterizing them requires some knowledge of the testbed. The following

subsections describe our testbed and the wireless interface that we use in this study.

2.1 Testbed

The measurements presented in this paper were collected from the BARWAN wireless overlay network, which is built from many commercially available networking technologies. The available network interfaces include 915 MHz and 2.4 GHz AT&T WaveLAN, Metricom, CDPD, infrared, and DirecPC. The mobile hosts (MH) are IBM Thinkpad laptops and the base stations (BS) are Dell PCs. Both the mobile hosts and base stations are x86 computers running a common UNIX operating system, BSD/OS version 2.1. Although the base stations and mobile hosts can have multiple wireless interfaces, we only evaluate the 915 MHz WaveLAN in this study. Having complete control over this network allow us to obtain many device-specific data that are valuable for channel modeling.

2.2 WaveLAN

WaveLAN is a commercial wireless network interface operating in the 902-928 MHz ISM (Industrial, Scientific, and Medical) band (AT&T), (Kohno, Meidan, and Milstein 1995). This direct sequence spread spectrum modem has a spreading factor of 11 chips per bit and a maximum bandwidth of about 2 Mbps. WaveLAN does not employ any forward error correction or retransmission at the device level, which makes it appropriate for studying wireless errors.

Medium access control (MAC) in WaveLAN resembles that of an Ethernet. Based on CSMA, it constantly monitors the state of the wireless channel. Before each transmission, the MAC must wait until the end of the current transmission plus a small delay known as the WaveLAN Inter-Frame Space (WIFS). Since it is difficult to detect a collision while transmitting, WaveLAN implements a collision avoidance scheme, which is different from the CSMA/CA being proposed IEEE 802.11 committee (Chen 1994). After the WIFS, the MAC waits for a number of *antenna slots* less than an initial back-off value. If the channel becomes busy before its slot, the MAC doubles its back-off window and continues to monitor the channel for the next contention period. This back-off process is repeated for a maximum of 15 times before the packet is dropped.

The WaveLAN link is slightly asymmetric due to differences in implementation of the base station's ISA interface and the mobile host's PCMCIA counterpart. The ISA adapter has an Intel 82586 CSMA/CD LAN controller, while the PCMCIA version has the newer 82593 chip. Although both interfaces have 0.5 nanosecond bit-times and 32-bit inter-frame space, the length of the antenna slot for back-off purpose is significantly different: 12 bit-times for ISA and 32 bit-times for PCMCIA. To partially make up the longer antenna slot, the PCMCIA controller sets its initial back-off window to 16 slots (512 bit-times) instead of the ISA's 32 (384 bit-times). The larger average delay for PCMCIA leads to lower throughput.

Like many other wireless interfaces, WaveLAN provides access to the receiver signal strength information. The three signal measurements for each packet are signal level, noise level, and signal quality. The signal level is a 6-bit measurement derived from the receiver's automatic gain control (AGC) setting at the beginning of each transmission. Similarly, the noise level is measured immediately after the end of the transmission, when no WaveLAN interface within detectable range can transmit. The signal quality is the 4-bit measurement derived from the information used for antenna selection at the beginning of packet reception.

3. TRACE COLLECTION

Reproducing accurate loss behavior for wireless network is challenging because the quality of the wireless channel can vary dramatically over time and space. To handle this problem, we use a trace-based approach consisting of three phases: trace collection, analysis, and validation. In the trace collection phase, a large number of traces are collected for many different scenarios. The analysis phase involves extracting the data of interest, such as packet errors, and modeling them. In the validation phase, the models are simulated and compared with the traces and measurements.

This section describes the collection phase. To collect traces for many different scenarios, we implement a general purpose trace engine and workload generator. The following two subsections describe the trace recording process and workload generation. We then investigate the effects of traffic parameters including packet size and transfer rates. Finally, we examine the effect of distance on packet error rate.

3.1 Trace Recording

The network tracing facility for both the base station and the mobile host is transparent to the applications, as shown in Figure 1. It contains two key components: the *trace agent* and *trace collector*. The trace agent resides in the kernel where it can record data that is either inaccessible or expensive to obtain at the user level. Kernel hooks are added to the network device drivers to pass each received packet and device information to the trace agent. These data are periodically extracted from the kernel buffer and saved to disk by the user-level trace collector.

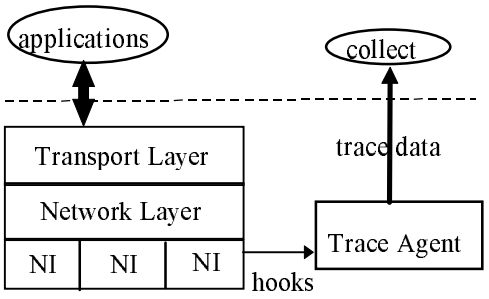

Figure 1 Network trace collection

When the trace collection program is executed, it opens a network tracing pseudo-device to start the kernel trace agent. The kernel trace agent then puts the wireless interfaces into promiscuous mode to capture all packets. Device-specific data such as signal level, noise level, signal quality, and error status are also recorded for post-processing and analysis.

3.2 Workload Generation

To generate traffic for error measurements, UDP is preferred over TCP because it has no error-recovery and connection establishment mechanisms. Each UDP packet is specially formatted to include information for error detection, such as a sequence number. For our modeling of packet errors, we decided to hold transmission rate and packet size constant such that our models can be converted into time-based ones. The effects of these two parameters are discussed in the following subsections.

To simplify the analysis process, each trace captures a single traffic stream. We also pay special attention to reduce the interference of the trace workload and other traffic. Although the UDP packets need not be received for the sender to make progress, a standard kernel would send out a "port unreachable" error message in reply to each UDP packet whose destination port has no user-level listener. A kernel hook is added to suppress these interfering packets while collecting traces.

3.3 Effect of Transmission Rate

Due to the asymmetric nature of WaveLAN, and most wireless networks in general, each direction of the link is traced independently. Our measurements indicate that the base station can send large UDP packets of more than 1000 bytes at the maximum throughput of about 1.6 Mbps. Using a similar packet size, the PCMCIA interfaces only achieve the maximum throughput of about 1.2 Mbps. This lower transmission rate is mainly due to the differences in implementation of the radio interfaces discussed in section 2.2. Due to limited space, we only present the analysis for the higher bandwidth down-link (base station to mobile host).

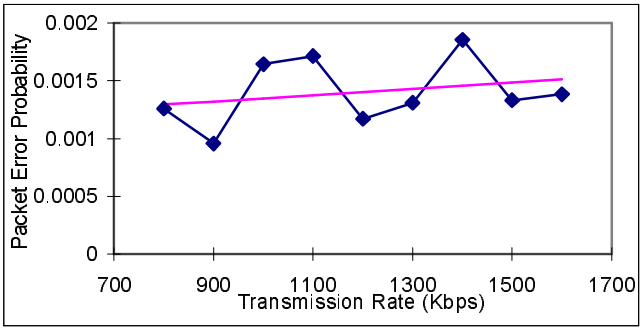

Figure 2. Packet error rate vs. transmission rate

To investigate the effect of the transmission rate, we use UDP streams with rates from 0.8 to 1.6 Mbps. The packet size and the distance are fixed at 1400 bytes and about 70 feet, respectively. Each data point in Figure 2 represents a single 1000-second trace. This plot reveals little correlation between the packet error probability and the transmission rate. Since the higher transmission rate does not lead to an increase in error probability, we will use nearly maximum transmission rates to capture more channel errors.

3.4 Effect of Packet Size

A similar experiment is conducted to examine the effect of packet size on error rate. In this experiment, the transmission rate is held constant at 1.5 Mbps while the packet size is varied from 100 bytes to 1400 bytes. To avoid fragmentation, packet size is chosen to be less than WaveLAN's maximum transfer unit of 1500 bytes.

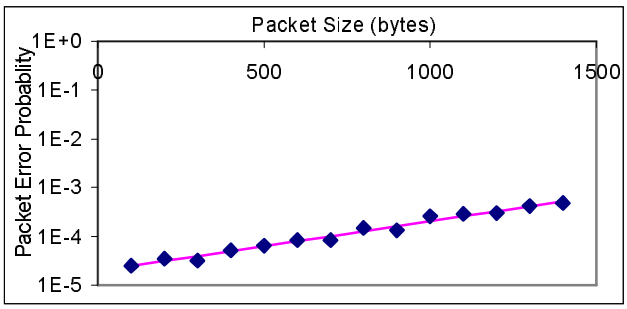

Figure 3. Packet error rate vs. packet size

The resulting packet error rates are plotted on the semi-log graph in Figure 3. Since the points in the graph lie on a straight line, we infer that the packet error rate increases exponentially with the packet size. Our regression analysis shows that packet error rate doubles for every 300-byte increment of the packet size.

The results of this experiment does not directly influence the choice of packet size for error modeling in section 4. For the data to be modeled, we choose to fix the packet

size at 1400 bytes because we believe it is more representative of our WaveLAN network.

3.5 Effect of Distance

Another parameter that is generally believed to have high correlation with signal level and error rates is the distance between the sender and receiver. In free space, the power of electromagnetic radiation varies inversely with the square of distance, making distance an ideal indicator of signal level as well as loss rate. In practice, this relationship is not exactly the inverse square due the imperfect propagation environment.

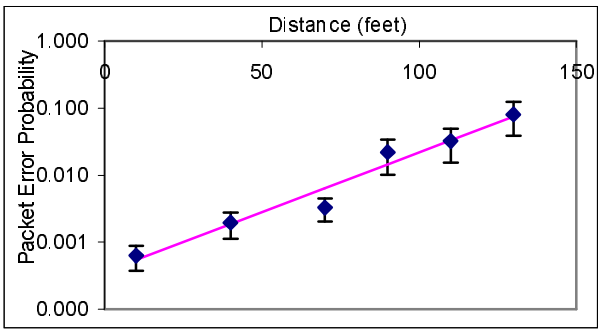

Figure 4. Packet error rate vs. distance

Figure 4 plots the packet error rate measured at various distances in our building. Each data point represents UDP transfer totaling about 10^8 bytes over three 1000-second traces for a single room. We infer from the graph that packet error rate is an exponential function of distance. The regression analysis reveals that packet error rate doubles for every increase of 17 feet.

Table 1 summarizes the data for the distance experiment. It contains the mean and standard deviation of the three key packet error characteristics: error rate, error length, and error-free length. The distance of less than 20 feet represents the single room scenario, where the sender and receiver have line-of-sight (LOS). The "average" row contains the averages of the data collected for all the distances. Data in the "mobile" row are collected while the experimenter is moving at the speed of about 5 feet per second inside our building. The packet error probability for this case is about 30% higher than the average, presumably because the presence of mobility produces more errors.

	packet error prob		error length		error-free length	
Distance	mean	stdev	mean	stdev	mean	stdev
10	0.000634	0.000229	1.000	0.000	1749.59	732.72
40	0.001955	0.000725	1.848	0.227	1023.43	334.40
70	0.003271	0.001086	2.005	0.210	574.56	148.45
90	0.021822	0.010412	2.428	0.018	122.89	58.99
110	0.032171	0.014786	2.540	0.482	87.26	39.75
130	0.080431	0.037123	2.298	0.290	28.90	8.82
average	0.024735	0.031554	2.054	0.564	549.37	699.39
mobile	0.032711	0.019324	2.365	0.089	84.60	37.36

Table 1. Error parameters for varying distances

The error length is defined to be the number of packets that are lost consecutively. Similarly, the error-free length is number of packets that are successfully received between two adjacent bursts of error. This can also be translated into the inter-arrival time of errors. The average error length is 2 to 3 packets for most distances. The only noticeable exception is the same-room LOS scenario, which produces only single-packet error bursts. The average error-free lengths for different distances vary by two orders of magnitude and have large standard deviation.

4. MODELING WIRELESS ERRORS

The loss characteristics of wireless channels have been empirically observed to be bursty due to various fading effects. As the result, evaluating wireless network protocols with a uniform error model will likely produce inaccurate results. We investigate several models to capture the burstiness of wireless errors. Starting with the Fritchman binary error model (Fritchman 1967), we develop two models for our trace data. In the following subsection, we describe the two-state Markov model. Following that is the analysis of the error and error-free distributions to improve the two-state model.

4.1 Two-state Markov Model

The basic error model contains two states: error and error-free, each having its own distribution. When the channel is in the error state, any packets sent would be either lost or corrupted. The opposite is true for the error-free state. Being a Markov model, the duration of staying in each state can be expressed in term of transitional probabilities, as shown in Figure 5. This is the most simple form of the multiple-state Fritchman model.

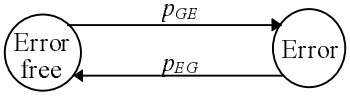

Figure 5. Two-state Error Model

In our derivation of parameters for the two-state Markov model, we label the error state E and error-free state G.

The associated transitional probabilities, p_{GE} and p_{EG}, can be computed from the average error-free length L_G and error length L_E. Since both L_E and L_G are geometrically distributed, the transitional probabilities can be expressed by the following formulas (Jain 1991):

$$p_{EG} = \frac{1}{L_E} \quad \text{and} \quad p_{GE} = \frac{1}{L_G}$$

The state transitions in the Markov model are memory-less. This nice property allows the transitions to be decided on an individual packet basis. If the length or duration of staying in each state is needed, the inverse cumulative distribution function of the geometric distribution can be used. Let u be a number uniformly distributed from 0 to 1, $U(0,1)$, the length x of staying in a state with the leaving probability p is:

$$x = \frac{\log(u)}{\log(1-p)}$$

	Error	Error-free
probability	0.382	0.006
length (pkts)	2.618	166.394
stdev of length	2.959	496.900

Table 2. Two-state Markov parameters

Table 2 provides the error and error-free length averaged from all the traces that are collected for the modeling purpose. Since the standard deviations for these parameters are large relative to the means, this model will likely be inaccurate. This motivates us to find other more accurate models without sacrificing complexity. The following section describes one such model.

4.2 Improved Two-state Model

Although the two-state Markov model can describe burstiness more accurately than the uniform error model, it cannot capture all loss behavior. The main limitation is the underlying assumption that the length distribution for each state is geometric. Our results indicates that neither the error and error-free length distributions is geometric.

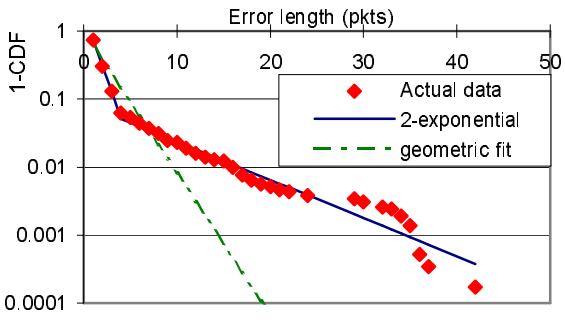

Figure 6. Error length distribution

Figure 6 plots the 1-CDF distribution on a semi-log graph. The dash line is the geometric fit for this data set. Our key observation is that almost 90 percent of all error bursts have length of less than 4 packets. This suggests that we should split error length distribution into two segments. Each segment is fitted with an exponential curve (straight lines on the semi-log graph). Since the best-fit curves do not intersect the coordinate (0,1), exponential terms must be multiplied by a constant factor e^k. The resulting CDF and its inverse are:

$$1 - F(x) = e^{-x/a} e^{k}$$

$$x = \lceil a(k - \ln(1 - F(x))) \rceil$$

Since these expressions are derived from those of the exponential distribution, we will label them *Exponential**. The parameters a and k for each segment can be separately obtained using linear regression on x and natural log of $1\text{-}F(x)$. For this type of segmented curves, the splitting points must also be specified. These are the 1-CDF values since we are interested it the inverse function. The parameters for the error length distribution are presented in Table 3.

The characteristic error-free length distribution is significantly different from that of the error length. In Figure 7, the 1-CDF is plotted on a log graph to provide more detail of the at the lower range. To avoid having an excessively complicated model, we decided to break this curve into 3 segments: 1 to 37, 37 to about 330, and the rest. The data points for the first two segments seem to fall on two different straight lines. Since the points of these segments fall on straight lines in this graph, they can be expressed by the following formula:

$$1 - F(x) = x^{-a} k$$

This expression describes the Pareto distribution. The inverse function that describes the error-free length is:

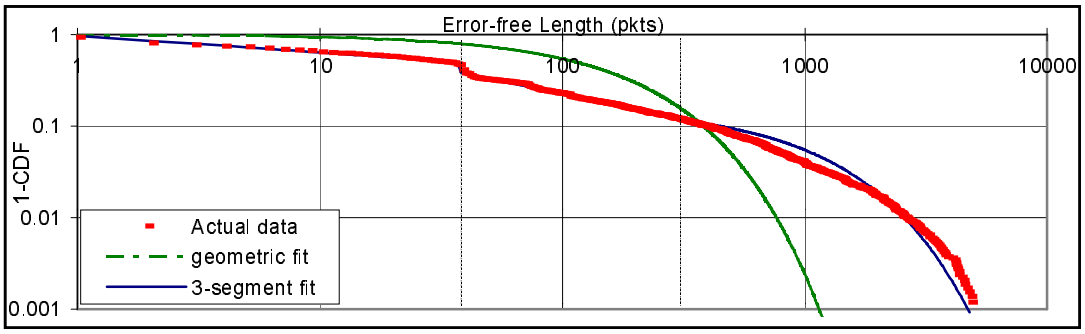

Figure 7. Error-free length distribution

BS to MH	segment	Distribution	1-CDF	a	k	R2
Error-free	1 to 37	Pareto	1.000	0.183	0.978	0.974
	38 to 330	Pareto	0.504	0.567	3.053	0.996
	331+	Exponential*	0.113	925.719	0.110	0.934
Error	1 to 3	Exponential*	1.000	1.153	0.563	1.000
	4+	Exponential*	0.051	7.753	-2.461	0.938

Table 3. Parameters of the Improved model for the down-link

MH to BS	segment	Distribution	1-CDF	a	k	R2
Error-free	1 to 36	Pareto	1.000	0.217	1.033	0.911
	37 to 300	Pareto	0.474	0.781	7.011	0.993
	301+	Exponential*	0.079	441.525	0.667	0.936
Error	1 to 3	Exponential*	1.000	1.153	0.563	1.000
	4+	Exponential*	0.042	5.145	-2.392	0.929

Table 4. Parameter of the Improved model for the up-link

$$x = \left\lceil \frac{1}{k}(1 - F(x))^{-\frac{1}{a}} \right\rceil$$

The parameters a and k for the first two segments can be separately obtained by applying linear regression on the natural log of both x and $(1\text{-}F(x))$.

The third segment of the error-free length distribution is a concave curve. On a log graph, this type of curve implies exponential decaying function. Therefore, we use the *Exponential** expressions presented earlier to describe this segment. In Figure 7, our best fit curve with 3 segments is close enough to the actual distribution that it is almost invisible. The dash curve represents the geometric distribution, which does not show a very good fit.

Table 3 summarizes all the computed parameters for our improved model. This model requires 13 parameters: five pairs of a and k, and three 1-CDF values. For each pair a and k obtained from the regression of the 1-CDF values, there is an associated coefficient of determination R^2. This value is the ratio of the regression sum of squares (SSR) to the total sum of squares (SST), which is an indication for the quality of fit. Our model fits the 1-CDF of the experimental data with coefficients of determination greater than 90%.

Wireless loss behavior for the up-link closely resembles that of down-link. The improved two-state model for the up-link channel also contains 2 *Exponential** curves for the error length and a combination of 2 *Pareto*, 1 *Exponential** for the error-free length. Table 4 provides the model parameters computed for the up-link. Since the error and error-free distributions are very similar for both directions, we have higher confidence that the errors really occur in the wireless channel.

5. VALIDATION

Validation is the last of the three phases for our trace-based approach to modeling. Since our primary use of the channel models is evaluating the impact of the errors on higher-layer network protocols, we choose TCP throughput as the metric for validating our models. The approach that we take is validating the models in the network simulator *ns* (McCanne and Floyd 1996). In the following subsections, we describe the simulation of errors in *ns* and provide a quantitative comparison of all the results.

5.1 Simulating packet errors

To evaluate the error models for WaveLAN, we made several key enhancements to *ns*, in particular, the wireless channel and WaveLAN MAC protocol. We also implement the error models to generate errors for the channel.

Our simulation setup consists of 2 WaveLAN nodes. A 1000-second file transfer using TCP is done from the sender to the receiver. In this setup, we simulate three error models: uniform packet error rate, two-state Markov, and our improved two-state. The uniform packet error rate of 0.01549 is computed from the average error and error-free length. Other parameters for the two-state Markov and the improved model are taken from Table 2 and Table 3.

Trace modulation contains two steps. The first step is extracting the time and duration of the errors from the traces. In the second step, the errors are replayed in the channel at the simulated time corresponding to the traces. For the validation purpose, we collect 10 new traces from various locations and replay the errors in the simulation setup described above. Next, we average the TCP throughputs for all the trace-driven simulations and compare it with the throughputs of the error models.

5.2 Quantitative Comparison

	Throughput (Mbps)	relative difference
Uniform	1.5134	0.213429
Markov	1.4054	0.126841
Improved	1.3071	0.047995
Trace	1.2472	0.000000

Table 5. TCP throughput of models

Table 5 lists the TCP throughputs for the three models and the trace modulation. Having the trace-driven simulation throughput as the reference, the relative difference is computed for each model. The improved model produces throughput of about 5% higher than that of trace-driven simulation. The Markov and uniform error models differ by 13% and 21%, respectively. The higher TCP throughput for the Markov model confirms that the error length generated by the this model (shown in Figure 6) are shorter than the real ones.

6. CONCLUSION

We have showed that trace-based approach is valuable for modeling wireless errors. The large number of traces that we have collected provide a good basis for developing and validating several wireless error models. From the trace data, we derive the parameters for two variations of the two-state error model. Although the Markov model is much simpler, it produces TCP throughput that differs by 13% from the traces. The improved model with 13 parameters significantly improves the accuracy, only 5% from the traces. While the simple Markov model may be adequate for some applications, the higher accuracy of the improved model will justify its complexity for others.

Although it is our main goal to provide the best channel error model to study all network protocols, the results obtained thus far are more suitable for evaluating higher-level network protocols. This is partly due to our immediate interest in TCP performance evaluations. We believe it is possible to improve this model to study link-level and MAC protocols by tracing smaller packets.

7. FUTURE WORK

Using a similar trace-based approach, we intend to characterize the behavior of the 915 MHz Metricom packet radio network and its interactions with the co-located WaveLAN network in the near future. Another goal is to improve our understanding of the mobility effects on various network characteristics such as errors and latency.

ACKNOWLEDGMENTS

The authors wish to thank Hari Balakrishnan, Bruce Mah, and Venkata Padmanabhan for many helpful suggestions in tracing and modeling of wireless networks. We also wish to thank Todd Hodes, who offered comments on early drafts of this paper.

This research was supported by the Defense Advanced Research Project Agency (DARPA) and the Air Force Material Command (AFMC) under contract numbers DAAB07-95-C-D154 and F196828-93-C-0193, and the State of California MICRO Program. Additional support was provided by AT&T, Hughes Aircraft, IBM Corp., Intel Corp., and Metricom. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of AFMC, AT&T, DARPA, Hughes, IBM, Intel, Metricom, Carnegie Mellon University, the University of California, the State of California, or the U.S. Government.

REFERENCES

AT&T Global Information Solutions Company. *Architecture Specification for WaveLAN Air Interface.*

Andersen, J. B., T. S. Rappaport and S. Yeshiva. 1995. Propagation Measurements and Models for Wireless Communications Channels. *IEEE Communications Magazine*, 42-49.

Bakre, A., and B. R. Badrinath. 1995. I-TCP: Indirect TCP for mobile hosts. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, 136-143.

Balakrishnan, H., S. Seshan, and R. H. Katz. 1995. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 469-481.

Caceres, R., and L. Iftode. 1995. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE*, 850-857.

Chen, K-C. 1994. Medium Access Control of Wireless LANs for mobile computing. *IEEE Network*, 50-63.

Cox, D. C., and R. P. Leck. 1975. Distributions of multipath delay spread and average excess delay for 910 MHz urban mobile radio paths. *IEEE Trans. Ant. Prop.*, 23: 206-213.

Eckhardt, D., and P. Steenkiste. 1996. Measurement and analysis of the error characteristics of an in-building wireless network. In *Proceeding of the ACM SIGCOMM*.

Fritchman, B. D. 1967. A binary channel characterization using partitioned Markov Chains. *IEEE Transactions on Information Theory*, 221-227.

Kohno, R., R. Meidan, and L. B. Milstein. 1995. Spread spectrum access methods for wireless communications. *IEEE Communication Magazine*, 58-67.

Jain, R. 1991. The art of computer systems performance analysis, 491.

McCanne, S., and S. Floyd. Network Simulator. http://www-nrg.ee.lbl.gov/ns/

McCanne, S., and V. Jacobson 1993. The BSD packet filter: a new architecture for user-level packet capture. In *Proceedings of the 1993 Winter USENIX Technical Conference*.

Noble, B., G. Nguyen, M. Satyanarayanan, R. H. Katz. 1996. Mobile network tracing. *RFC Draft*.

AUTHOR BIOGRAPHIES

GIAO T. NGUYEN is a graduate student in Electrical Engineering and Computer Science at the University of California at Berkeley. His research interests are in the areas of computer networks, distributed and mobile computing. He received a Bachelor of Science in Computer Science and Engineering from the University of California at Los Angeles in 1994.

RANDY H. KATZ is a leading researcher in computer system design and implementation. His research experience has spanned numerous disciplines. He has written over 120 technical publications on CAD, database management, multiprocessor architectures, high performance storage systems, and video server architectures. Professor Katz developed the concept of Redundant Arrays of Inexpensive Disks (RAID), now a $3 billion industry segment. Katz's recent research has focused on wireless communications and mobile computing applications. He is a Fellow of the ACM and a Fellow of the IEEE.

BRIAN NOBLE is a Ph.D. candidate in Computer Science at Carnegie Melon University. His research interests include distributed systems, file systems, data repositories, mobile computing, and the measurement and evaluation of deployed systems. He received a Bachelor of Science in Electrical Engineering and Computer Science from the University of California at Berkeley in 1991, and a Master of Science in Computer Science from Carnegie Mellon in 1993.

MAHADEV SATYANARAYANAN is a professor of Computer Science at Carnegie Mellon University. His current research addresses the problem of information access in mobile computing environments. The Odyssey mobile computing platform and the Coda File System are outcomes of this work. Earlier, he was a principal architect and implementor of the Andrew File System, which focused on issues of scale, performance and security. Later versions of this system have been commercialized and incorporated into the Open Software Foundation's DCE offering. Satyanarayanan received the Ph.D. in Computer Science from Carnegie Mellon, after Bachelor's and Master's degrees from the Indian Institute of Technology, Madras. He is a member of ACM, IEEE, Usenix, and Sigma Xi, and has been a consultant and advisor to industry and government.