July 2008

Characterizing Temporal SNR Variation in 802.11 Networks

Ratul K. Guha
University of Pennsylvania

Saswati Sarkar
University of Pennsylvania, swati@seas.upenn.edu

Recommended Citation

Ratul K. Guha and Saswati Sarkar, "Characterizing Temporal SNR Variation in 802.11 Networks", . July 2008.

Characterizing Temporal SNR Variation in 802.11 Networks

Abstract
The analysis and design of wireless medium access control (MAC) protocols, coding schemes, and transmission algorithms can significantly benefit from an understanding of the channel quality variation.We attempt to represent channel quality variation using a finite-state birth–death Markov model. We outline a method to compute the parameters of the model based on measured traces obtained using common wireless chipsets. Using this Markov chain, we statistically evaluate the performance based on the channel quality, long-term correlations, and burst length distributions. Such a model significantly performs better than a traditional two-state Markov chain in characterizing 802.11 networks while maintaining the simplicity of a birth–death model. We interpret the variation of the model parameters across different locations and different times. A finite-state stationary model is amenable to analysis and can substantially benefit the design of efficient algorithms and make simulations for wireless network protocols faster.

Keywords
analytical models, channel modeling, wireless LAN

Comments

Characterizing Temporal SNR Variation in 802.11 Networks

Ratul K. Guha and Saswati Sarkar, *Member, IEEE*

***Abstract*—The analysis and design of wireless medium access control (MAC) protocols, coding schemes, and transmission algorithms can significantly benefit from an understanding of the channel quality variation. We attempt to represent channel quality variation using a finite-state birth–death Markov model. We outline a method to compute the parameters of the model based on measured traces obtained using common wireless chipsets. Using this Markov chain, we statistically evaluate the performance based on the channel quality, long-term correlations, and burst length distributions. Such a model significantly performs better than a traditional two-state Markov chain in characterizing 802.11 networks while maintaining the simplicity of a birth–death model. We interpret the variation of the model parameters across different locations and different times. A finite-state stationary model is amenable to analysis and can substantially benefit the design of efficient algorithms and make simulations for wireless network protocols faster.**

***Index Terms*—Analytical models, channel modeling, wireless LAN.**

I. Introduction

Wireless networks are being rapidly deployed all over the world. In the USA, several companies such as Boingo, Cometa, and T-mobile are deploying nationwide IEEE 802.11b-based wireless local area networks. This large-scale deployment has motivated research into the design of better wireless systems. An important step in that direction is to characterize the signal-to-noise ratio (SNR) in these networks. The SNR determines several important attributes such as packet loss, which affects network performance and design at all levels of the network stack. The challenge in this characterization is that the SNR process is a result of both interference due to simultaneous transmission of multiple users and vagaries of radio wave propagation. Several resource allocation policies in wireless local area networks require this characterization in predicting channel qualities. Also, distribution of error bursts and error-free bursts can be obtained from the characterizations of temporal variations of the SNR as the packet error processes depend on the SNR. Thus, the SNR is a more fundamental attribute of the system. Our goal is to approximately characterize the channel behavior with a stationary process that is conducive to analysis and simulation for large systems. Toward this end, we characterize the SNR variations on a packet timescale. Furthermore, we use a birth–death Markov model with a finite number of states to characterize the SNR variations. The advantage of birth–death models is that they are simple and analytically tractable.

Significant research has been directed toward modeling fading channels with Rayleigh distributed SNR [16], [21]. However, it is not clear that the same techniques would apply for modeling IEEE 802.11 channels as these experience both interference and frequency-selective fading. Most of the research for modeling IEEE 802.11 systems has been directed toward modeling protocol behavior [4], [12] or obtaining distributions for error bursts and sequences of error-free transmissions using traces of packet losses [1], [15], [18]. Much attention has been devoted to the two-state Gilbert–Eliott model [6], [7] due to its analytical simplicity. However, the two-state models have severe limitations. This has been observed by Willig *et al.* [19], who use a bipartite model to characterize packet losses, whereas Khayam and Radha [10] use hidden Markov models for modeling losses. Regression and learning-based models have also been proposed for link quality estimation [11], [20]. These models are nonstationary and, as a result, are more suitable for simulation and trace generation rather than analysis. Our goal, on the other hand, is to present a stationary birth–death model that is simple and rapid for computation without significantly compromising the model performance.

We characterize the temporal variation of the SNRs of IEEE 802.11 channels and explain several attributes of the wireless systems using the characterization. Note that the variation of the SNR will be a nonstationary process as this is effected by the traffic, which is not a stationary process. Hence, no stationary model will provide an exact statistical match. However, nonstationary models exclude many analytical techniques, and simulations using nonstationary models can take a significant amount of time, particularly in large-scale systems.

Drawing from the literature known for fading channels, we propose a framework for determining the parameters of the birth–death Markov model for IEEE 802.11 channels (Section II). We discuss two forms of measurements, namely passive and active. Passive measurements involve collecting traces from the existing network traffic. Active measurements

Manuscript received October 5, 2006; revised April 14, 2007, August 6, 2007, and October 16, 2007. This work was supported in part by the National Science Foundation under Grant ECCS-0621782, Grant NCR-0238340, Grant CNS-0435306, Grant CNS-0435141, and Grant CNS 0721308. This paper was presented in part at the IEEE Wireless Communications and Networking Conference, Las Vegas, NV, April 2006. The review of this paper was coordinated by Prof. M. Guizani.

R. K. Guha was with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA. He is now with Telcordia Technologies, Piscataway, NJ 08854 USA (e-mail: rguha@research.telcordia.com).

S. Sarkar is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: swati@ee.upenn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2007.912599

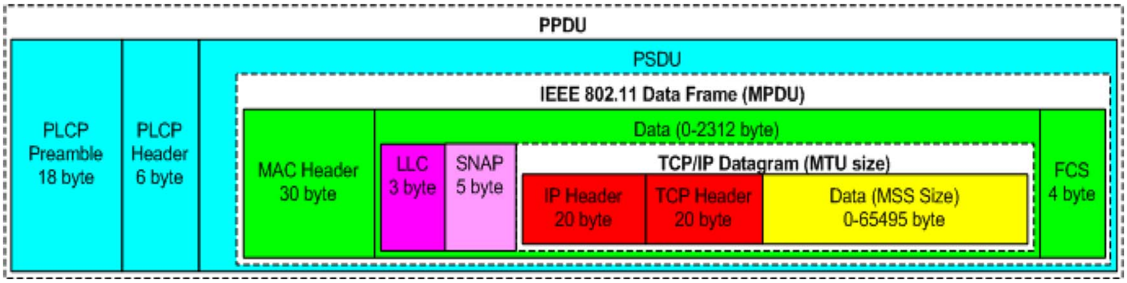

Fig. 1. 802.11 frame format.

involve sending probe packets. Active measurements are often useful since probe packets can be periodically generated to update the parameters of the model. However, active measurement has the disadvantage of affecting the very channel that is being measured. This happens because of the interference generated by the probe traffic. Passive measurement, on the other hand, requires longer timescales for trace collection. However, various user characteristics such as mobility and other nonstationary behavior of the users can be studied using passive measurements. In this paper, we primarily focus on passive measurements.

In Section III, we discuss the passive measurement setup and broadly classify the environments in which we collected the traces. We subsequently validate our model by a statistical comparison with passively collected SNR traces over wireless fidelity (Wi-Fi) channels (Sections IV and V). This validation demonstrates that a simple birth–death Markov model reasonably accurately characterizes the SNR variations in IEEE 802.11 channels. In Section VI, we evaluate the effectiveness of active measurements and compare the model with that obtained from the passive measurements. We notice that active measurements can be leveraged to construct the channel model without significantly disrupting the network. Finally, using our model, we investigate and explain several attributes of the wireless systems, such as statistics of error bursts, in Section VII. For example, in [1] and [19], the authors conclude that the two-state Gilbert–Eliot model is not suitable to characterize packet losses, whereas the authors in [10] claim that two-state models are sufficient to capture the packet loss process but not the bit losses. Using our characterization, we explain the difference between these conclusions and provide several general guidelines about the characteristics of IEEE 802.11 channels. We conclude this paper in Section VIII.

II. Modeling the 802.11 Channel

We first outline the 802.11 frame format and describe how to evaluate the packet error probabilities based on bit probabilities (Section II-A). The bit error probabilities are functions of the signal-to-interference-and-noise ratio. We will then present a framework for modeling the SNR using a finite-state Markov model and present a methodology for computing the parameters of the model based on measured traces in Section II-B. Henceforth, the ratio of the signal power to the interference and ambient RF power is referred to as the SNR.

A. 802.11 PHY

The IEEE 802.11b physical layer (PHY) is an extension of the original direct-sequence spread-spectrum (DSSS) PHY. It operates in the 2.4-GHz industrial, scientific, and medical band and provides PHY data rates of 5.5 and 11 Mb/s, in addition to the 1- and 2-Mb/s rates supported by the original DSSS. The 1-Mb/s rate is based on binary phase-shift keying (BPSK), whereas the 2-Mb/s rate is based on quadrature phase-shift keying modulation. They are encoded using DSSS based on the 11-bit Barker chipping sequence that results in a signal spread over a wider bandwidth at a reduced RF power. For 5.5 and 11 Mb/s, IEEE 802.11b uses the complementary code keying modulation scheme, which is a variation of the m-ary orthogonal keying modulation. For each word, the 5.5-Mb/s rate encodes 4 bits, whereas the 11-Mb/s rate encodes 8 bits. The spreading maintains the same chipping rate and spectrum shape as the original 802.11 DSSS and, hence, occupies the same channel bandwidth. There are 11 channels, and each channel occupies 22 MHz around the center frequency. This allows for three nonoverlapping channels (i.e., 1, 6, and 11) in the band.

The IEEE 802.11b frame format is shown in Fig. 1. When a higher layer frame, which is also called the medium access control (MAC) service data unit, arrives at the MAC layer, it is encapsulated in a MAC protocol data unit (MPDU) by adding a MAC header and a frame control sequence (FCS). The MAC header is 24 bytes, and the FCS is 4 bytes. We now evaluate the packet success probability. The MPDU is passed down to the PHY, which attaches a 6-byte physical layer convergence protocol (PLCP) header and an 18-byte preamble. The PLCP header and preamble are transmitted at 1-Mb/s BPSK. Suppose that an L-byte MPDU is to be transmitted using PHY rate r, where r can be 1, 2, 5.5, or 11 Mb/s. A packet is successful if both the PLCP header and the MPDU are correctly received. Hence, the success probability of a packet $P_r(L)$ with an L-byte-long MPDU can be written as

$$P_r(L) = P_1(24)P_r^{\mathrm{MPDU}}(L) \tag{1}$$

where

$$P_1(24) = \prod_{i=1}^{192} P_1^{b,i} \text{ and } P_r^{\mathrm{MPDU}}(L) = \prod_{i=1}^{8L} P_r^{b,i} \tag{2}$$

where $P_r^{b,i}$ is the bit success probability shown by the ith bit when the transmission occurs at rate r. The equations assume

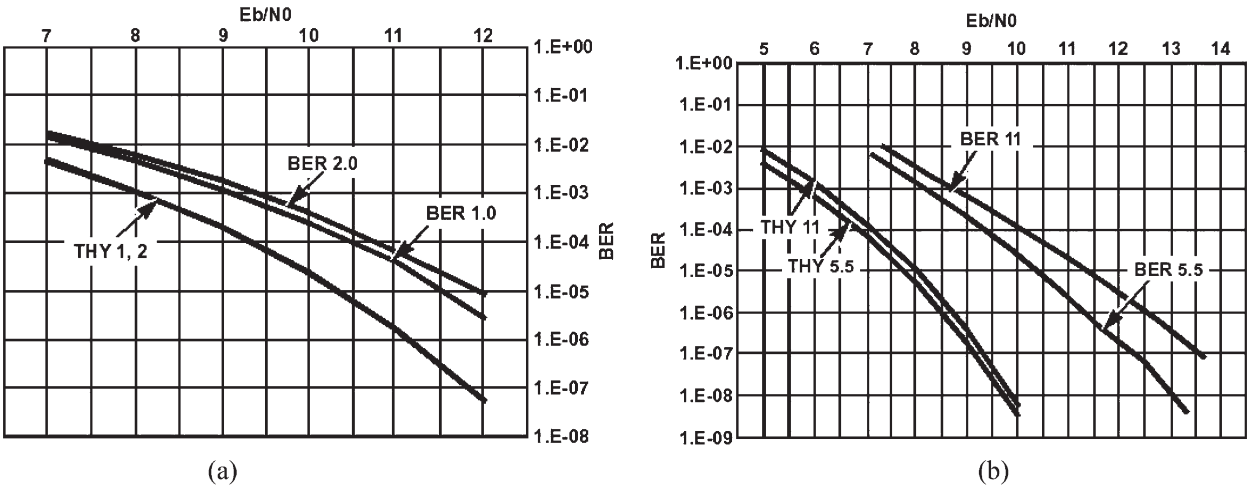

Fig. 2. SNR–BER curves. The curved marked BERs are the actual curves for the chipset, whereas the curves marked THY are the theoretically obtained curves. (a) SNR-BER for PSK modes. (b) SNR-BER for CCK modes.

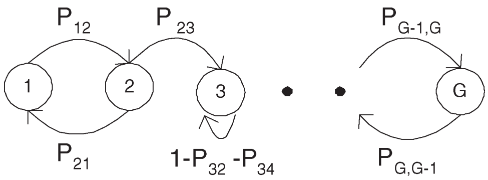

Fig. 3. Channel model.

conditional independence of the bit success probabilities, given the SNR at a bit. $P_r^{b,i}$ can be obtained from the actual chipset SNR–bit error rate (BER) curves in Fig. 2.

Typical off-the-shelf wireless chipsets [2], [9] report SNR measurements in terms of the received signal strength indication (RSSI) value once per packet. Using the available data, we now present the procedure to evaluate the parameters of the Markov chain based on measurements that can be obtained from common hardware.

B. Finite-State Model for SNR Variation

We next present a methodology to model SNR variations using a finite-state Markov chain with state space $\mathcal{S} = s_1, s_2, \ldots, s_G$ (Fig. 3). Each of the G states corresponds to a certain channel quality and an associated packet success probability. G is a parameter and needs to be fixed. We will later discuss how to select an appropriate value for G. We will obtain the transition rates and the steady-state probabilities for the Markov chain. We assume that the transitions happen between the adjacent states. Such a choice is intuitive because normally the channel transitions are not abrupt but continuous, and then, if the SNR intervals are chosen with a high granularity, then transitions between nonadjacent states can be avoided. This allows us to use birth–death models that are amenable to analysis. We adapt the methodology used for fading channels in [16] and [21]. In [16], the model parameters are analytically obtained based on a fading model. The main difference between [16] and our paper is that the authors in [16] derive the values of the model parameters from fading models, whereas we compute the values of the model parameters from measurement traces. The reason we adopt this approach is that, in our case, the reception quality is determined both by fading and interference, and models used for fading may not be appropriate for interference. We will demonstrate that the framework for fading channels is a good model for Wi-Fi channels. The SNR trace, i.e., the SNR recording for every received packet, is to be partitioned into G states based on time duration. A received packet is said to face state s_k if the SNR values of the bits comprising the packet face are in the range $[\Gamma_k, \Gamma_{k+1})$. We will compute the SNR thresholds $\vec{\Gamma} = [\Gamma_1, \ldots, \Gamma_{G+1}]$, $\Gamma_1 = 0$, $\Gamma_{G+1} = \infty$, and $\Gamma_k < \Gamma_{k+1}$, to partition the SNR process. Let P_{ij} and π_i be the state transition and steady-state probabilities, respectively. Since the transitions happen between the adjacent states, $P_{k,i} = 0$ for $|k - i| > 1$.

Let $N(\Gamma)$ be the level crossing rate of the SNR process in the positive direction only (or in the negative direction only). Essentially, this is the number of times the SNR crosses a level Γ in a particular direction divided by the total time.

Let $\bar{\tau_k}$ be the average duration of the SNR interval $[\Gamma_k, \Gamma_{k+1})$. This is the average time the SNR process continuously remains between Γ_k and Γ_{k+1} over a measurement interval T. For simplicity of exposition, we assume a fixed packet size with transmission time T_p to outline the procedure. We require that the average duration of a state is some constant times the packet transmission time T_p, i.e., $\bar{\tau_k} = c_k T_p$. We let $c_k = c\ \forall k$. In other words, each state has the same average duration. We justify such an assumption. The states in which the system spends more time are more important to characterize the system. Thus, these states should be represented with a higher SNR granularity. This can be achieved by subdividing these states such that the time duration assigned to each substate is reduced, which, in turn, means that all states have equal duration.

The steady-state probability π_i is the total time during which the SNR level is between Γ_i and Γ_{i+1} divided by the total time T, i.e., $\pi_i = \text{Prob}(\Gamma_i \leq \Gamma < \Gamma_{i+1})$. Note that π_i can be evaluated from the trace if Γ_i and Γ_{i+1} are known. $\bar{\tau_k}$ is then the ratio of the total time the signal remains between Γ_k and Γ_{k+1} and the number of these signal segments. The number of such signal segments differs from $(N(\Gamma_k) + N(\Gamma_{k+1}))T$ by at most 1 since the number of upcrossings and downcrossings can differ by 1, and this error becomes insignificant for large traces

$$\bar{\tau_k} = \frac{\pi_k T}{(N(\Gamma_k) + N(\Gamma_{k+1}))\, T} = \frac{\pi_k}{N(\Gamma_k) + N(\Gamma_{k+1})}.$$

Thus, we have

$$cT_p = \frac{\pi_k}{N(\Gamma_k) + N(\Gamma_{k+1})}, \qquad \text{for } k \in [1, \ldots, G]. \qquad (3)$$

We require our solution to satisfy G equations in (3). The variables are c and $\vec{\Gamma} = [\Gamma_1, \ldots, \Gamma_{G+1}]$ with $\Gamma_1 = 0$ and $\Gamma_{G+1} = \infty$. Hence, there are G variables. One way to solve this system is to take an initial guess for c and subsequent iteration. $\Gamma_2, \Gamma_3, \ldots, \Gamma_G$ can be successively obtained using $k = 1, 2, \ldots, G$, respectively, in (3). Note that $N(\Gamma)$ can be evaluated from the trace for every Γ.

Once $\vec{\Gamma}$ is determined, the transition probabilities can be computed as the ratio of the level crossing rate and the average number of packets staying in that state, i.e.,

$$P_{k,k+1} = \frac{N(\Gamma_{k+1})T_p}{\pi_k}, \qquad k = 1, 2, \ldots, G-1$$

$$P_{k,k-1} = \frac{N(\Gamma_k)T_p}{\pi_k}, \qquad k = 2, \ldots, G.$$

Using the aforementioned transition probabilities and (3), the memory of a state, i.e., $1 - P_{k,k+1} - P_{k,k-1}$, is $1 - (1/c)$.

We describe the method to obtain the average success probabilities for the states. The SNR–BER mappings are obtained from the characteristics in Fig. 2 [9]. Let $P(x)$ be the packet success probability for SNR level x. Also, let $n(x)$ be the fraction of time SNR level x is observed in the SNR trace. Then, the average probability of success conditioned on state s_k is $\alpha(k) = \sum_{x=\Gamma_k}^{\Gamma_{k+1}} P(x)n(x)/\pi_k$, where the summation is over the discrete SNR values that comprise state s_k.

Next, we present our measurement setup and trace collection scenarios. We will use the data from the trace recordings and evaluate the Markov chain based on the preceding discussion.

III. Measurement Setup and Trace Collection

We now describe our measurement setup and the trace collection procedure. Our hardware comprises two Dell Latitude X200 laptops with internal Agere cards and a PCMCIA slot. The wireless network interface cards are Cisco CB21AG-AK9 and the RFGrabber [17] with an Intersil chipset. A host Windows XP workstation running AiroPeek NX [17] connects to the RFGrabber to store packets. We use the AiroPeek NX [17] packet analyzer for measurement purposes.[1] We capture all IEEE 802.11 management packets (e.g., beacons), data packets [e.g., Transmission Control Protocol and User Datagram Protocol (UDP)], and control packets [e.g., acknowledgements (ACKs)], which comprise 98% of the traffic. For each packet from the access point (AP), the AiroPeek NX trace records the time stamp, signal strength, channel number, noise level, packet size, transmission speed, protocol, and packet error flag. Erroneous packets are flagged according to the radio error, decryption error, or cyclic redundancy check error.

We conduct our measurement in different kinds of Wi-Fi environments and at different time instants to obtain a variety of possible channel characteristics. We broadly describe the different settings that we believe are representative for recording the data:

1) locations close to the AP with few users (e.g., classrooms);
2) locations close to the AP with a large number of users (e.g., hotspots and conference rooms);
3) locations distant from the AP in relatively less crowded areas (e.g., parks);
4) locations distant from the AP in crowded areas.

In the first case, the channel is referred to as *good*. Here, the signal strengths are high, and the error probabilities are small ($< 3\%$) with SNR > 25 dB. The second and third cases are what we refer to as *intermediate* channels. In the second case, the interference is higher than that of the first. For the third case, the signal strength is lower than that of the first. As compared to the first case, both the second and third cases experience a lower SNR of 10–20 dB and a higher packet error with error rates of 5%–10%. The fourth case is referred to as a *poor* channel, with the signal strength lower than that of the first and the interference higher than that of the first. Here, the packet error rates are as high as 50% with SNR < 10 dB. The first three cases are more likely to arise in practice, but we consider all cases for completeness.

The trace recording is carried out for a duration of 20 min at a 2-h interval for a period of seven days. We observe that a majority of the users have short session times that are less than 10 min and that the duration of portions when the usage pattern remains similar is approximately around 6–7 min. This observation is consistent with reports in [3]. We continuously record data for 20 min at any given time.

Next, we evaluate the effectiveness of the finite-state Markov model for the temporal SNR variations.

IV. Testing of the Model Based on Traces

We proceed to evaluate the efficacy of the finite-state Markov model. We first evaluate the performance of the model in terms of tracking the SNR directly obtained from the measurement. We also evaluate the model in terms of capturing statistics that are relevant from the perspective of higher layer wireless protocols. The statistics of interest typically comprise the moments of the error-free lengths and error burst lengths. Also, long-term correlation of the packet error process is of interest in predicting the future channel behavior. Since different performance attributes are of interest for different systems, we evaluate our model based on several different factors: 1) the values of the maximum and average differences between the model parameters obtained from our methodology and those obtained from the model traces (Section IV-A); 2) the statistical divergence between the distribution of the channel states obtained from our model and the distribution directly

[1] We use Network Stumbler (NS) [14] for cross verification of signal levels. NS is an active scanner that scans all 11 channels sending probe requests.

obtained from the traces (Section IV-A); and 3) the differences in performance attributes (e.g., error burst lengths and error-free burst lengths) (Sections IV-B–D). We make our conclusions based on all the aforementioned factors. We specify the values of the maximum and average differences between the model parameters so that the system designers can decide whether this difference leads to the accuracy they desire for the performance attribute that is important to them.

We now describe the statistical measure that we use to quantify the performance of the model.

Let $p(x)$ and $q(x)$ be two probability mass functions defined over a common set $\mathcal{X}$. We now describe a commonly used statistical measure that quantifies the "distance" or the relative entropy between two probability distributions. This comprises a general measure and allows us to compare the statistics of all orders for two distributions.

The *Kullback–Leibler divergence* (KLD) [5] is defined as

$$D\left(p(x)\|q(x)\right) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}.$$

The KLD is zero when the distributions are identical, and it is strictly positive otherwise. The KLD is a measure of the "distance" between two distributions. However, the measure is not symmetric and does not satisfy the triangle inequality. The definition of the divergence measure carries a bias. This discrimination is larger for random variables with higher entropy. The entropy $H(p(X))$ of the random variable X with distribution $p(X)$ is $\sum_{x\in\mathcal{X}} p(x)\log(1/p(x))$. Hence, to evaluate the model, it is important to weigh in the entropy of the source, which could be large. Hence, we use the normalized KLD (NKLD) [13], i.e.,

$$\text{NKLD}\left(p(x)\|q(x)\right) = \frac{D\left(p(x)\|q(x)\right)}{H\left(p(X)\right)}.$$

Since the order of the distributions is important, we consider the distributions derived from the measured trace as $p(x)$ and those derived from the Markov model as $q(x)$ in the subsequent discussion.

A. SNR Variation

To test the model, we first examine the SNR values from the trace at equal intervals of time for the purpose of developing the Markov chain. The interval corresponds to transmission times of data packets that are longer than the control and management packets. We used the transmission time for a 1000-byte packet (0.73 ms) as the interval. The computation time for model generation depends on the length of the available trace. In our case, generating the model from a 7-min trace required approximately 15 s on a 750-MHz laptop with 256-MB random access memory. From the Markov chain, we generate an artificial channel trace for the SNR variation.

The difference between the SNR values from the actual trace and the trace obtained from the channel model is examined. The root-mean-square percentage difference is observed to be less than 4%. We also evaluate the NKLD between the probability mass function of the SNR obtained from the measured trace and the Markov model. The probability mass function defines the probability of the SNR lying in certain intervals. The setup is shown in Fig. 4. The NKLD measure is plotted for three kinds of channels in Fig. 5 as the number of states increases. The low value of the NKLD measure signifies the proximity of the two distributions. To get a feel of how rapidly the NKLD increases as the distributions become separate, we use a naïve channel model and evaluate its performance. We partition the entire SNR range into G states of equal SNR length that are equiprobable. The NKLD value will increase to more than 0.7 for all values of G. On the other hand, using a time-varying distribution using phase-type models reduces the NKLD to below 0.009. This improvement comes at the expense of using a nonstationary model.

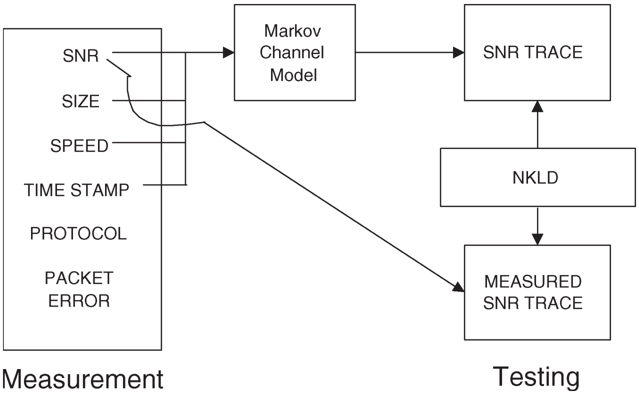

Fig. 4. Measurement and testing setup for an SNR trace.

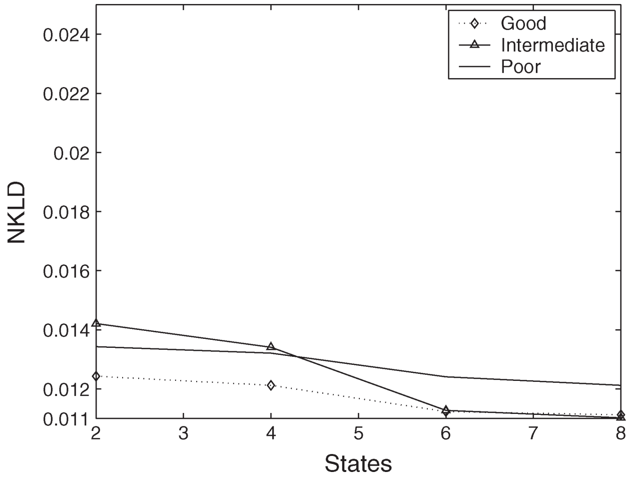

Fig. 5. SNR distribution with a birth–death Markov chain.

We notice a small value of the divergence between the actual SNR trace and the trace obtained from the channel model that does not further decrease with the increase in the number of states. We first investigate whether this divergence is lower if the channel is modeled by a more general Markov chain. For this, we use a general Markov model to characterize the SNR process where the transitions can occur between any two states. The state space is defined in a way similar to the birth–death model. The transition probability from a state X to any other state Y is obtained from the SNR trace by calculating the ratio of the total number of transitions from X to Y and the

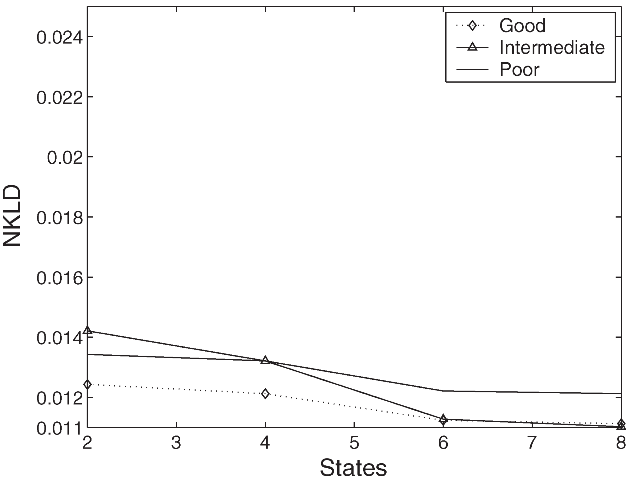

Fig. 6. SNR distribution with a Markov chain having arbitrary transitions.

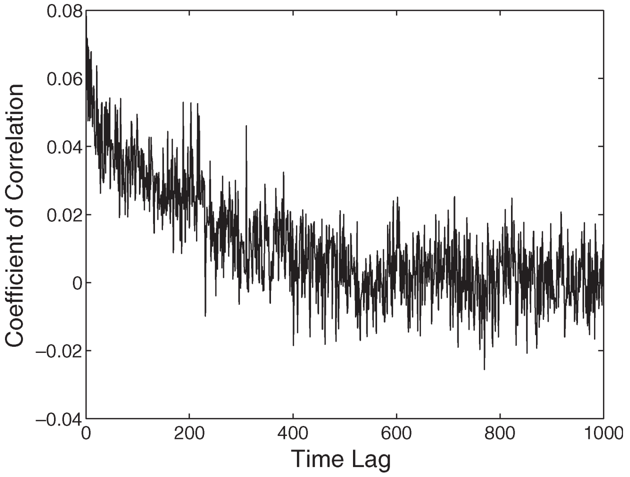

Fig. 7. Coefficient of correlation with time lag.

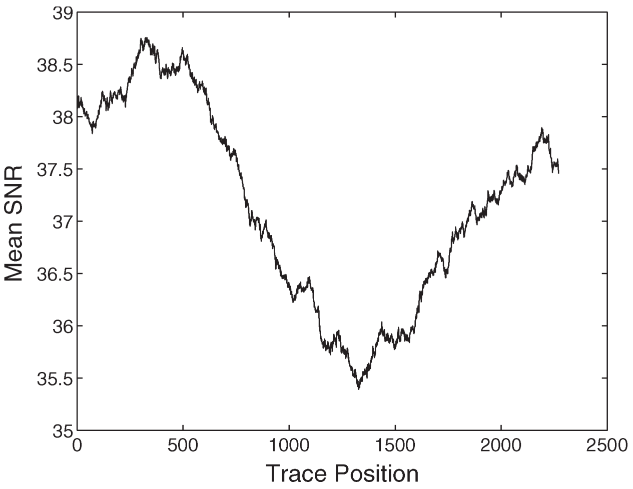

Fig. 8. Mean values over different windows. The y-axis value corresponding to the x-axis point k denotes the mean from position k to $k + 1000$ in the trace.

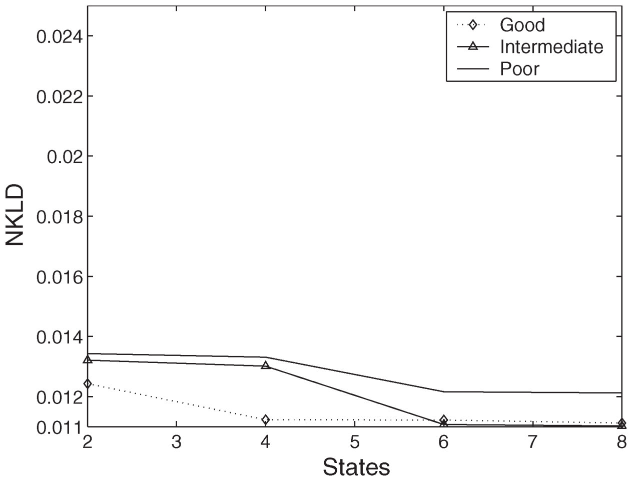

Fig. 9. Channel model using beacon packets.

total number of transitions from state X. The total number of transitions from state X includes all the transitions out of X and those that go back to themselves. The performance using this model is shown in Fig. 6. We notice that a Markov model with arbitrary transitions does not yield any significant improvement in the model performance. The intuition is that the model represents states with a high time granularity, and within this timescale, the SNR values gradually change. Hence, transitions to faraway states rarely occur. Hence, among Markov models, a birth–death model is good enough.

We believe that the reason for the residual divergence is that we use an irreducible, aperiodic, finite-state Markov chain to model the SNR process. Such a model has a stationary distribution. However, the SNR process is expected to be nonstationary due to physical reasons such as node mobility, traffic variations, fading, etc. We establish the nonstationarity of the process using a statistical analysis of the trace. The stationarity of the trace is investigated using a derived time lag. In Fig. 7, we plot the coefficient of correlation as the time lag W is increased, that is, we select vectors from the trace shifted by W. We note that for $W > 800$, the correlation goes down to zero, ensuring that we have a sufficiently large lag such that the transients have died down. Hence, to determine the mean, we use a time lag of 1000. We note that the mean over time lags of 1000 plotted in Fig. 8 changes by almost 10% with time, indicating nonstationarity. This nonstationary behavior cannot be captured by stationary ergodic models, resulting in the observed divergence. Nevertheless, the residual divergence is small with a stationary ergodic birth–death model, which, unlike nonstationary models, is amenable to analysis.

We also develop the Markov chain based on SNR information obtained from 802.11 beacon packets in Fig. 9. Although the beacon packets are relatively infrequent (100 ms), the Markov chain well approximates the channel behavior, even in this case. Furthermore, in Sections IV-B and C, we also demonstrate that the statistical characteristics of empirical observations for several attributes such as burst length processes match those obtained from the birth–death model, which corroborates our conclusion that a birth–death Markov chain models the SNR process reasonably well.

B. Packet Error Model and SNR Sampling

We now study various attributes of the system for, e.g., the packet error process that can be derived from the SNR

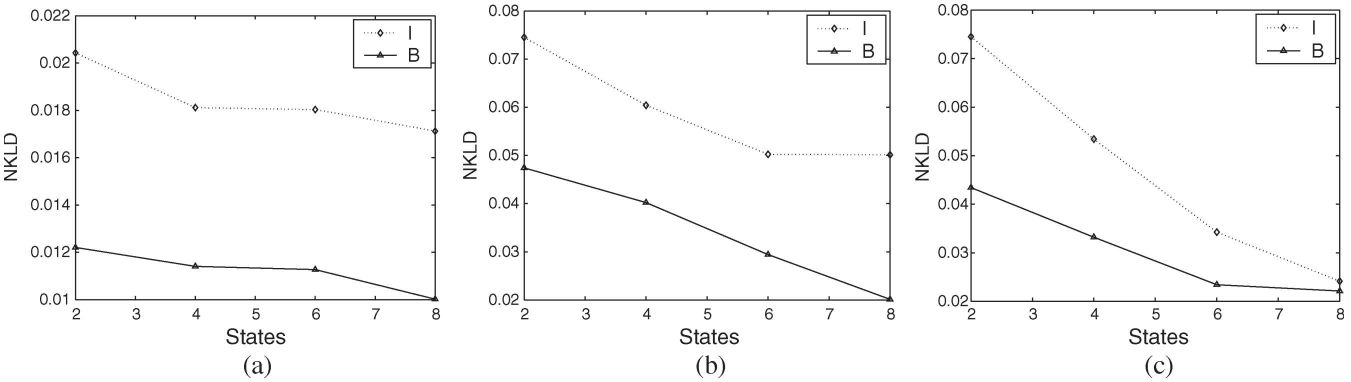

Fig. 10. NKLD measure for burst lengths from the actual error trace and the packet trace from the SNR for different kinds of channels. (a) Good channel. (b) Poor channel. (c) Intermediate channel.

process. Toward this, we first study and evaluate the effect of fundamental measurement limitations. We will then evaluate the efficacy of the SNR measurements in characterizing the packet errors.

We proceed to evaluate the effect of two fundamental measurement limitations imposed by software and hardware. First, common Wi-Fi hardware records the signal strengths and the noise levels once per packet. However, any of the intermediate bits of a packet can get corrupted, and the predicted packet errors based on the reported SNR could be significantly different from the actual packet errors. We will investigate whether the memory in the channel is high enough for an SNR interpolation between packet boundaries to be useful.

Second, the SNR–BER characteristics reported by card manufacturers, as shown in Fig. 2, are obtained under an additive white Gaussian noise environment, which might not be the case in practice. Also, the RSSI values, i.e., the units in which the cards report the signal levels and signal values (in decibel milliwatts), do not have a one-to-one correspondence. For example, in Cisco cards, RSSI values of 63 and 64 both correspond to −44 dBm. These factors can result in discrepancies between the calculated packet errors based on SNR measurements and the observed packet error process. We will check how closely the packet error model that we use mimics the behavior of the actual channel. Specifically, we would be focusing on the distributions of the packet error burst lengths and error-free lengths. The random variable depicting the error-free length I denotes the number of good packets received between error packets. The error burst length B is the length of consecutive packet errors. Packet error bursts are responsible for degrading the throughput in wireless transmission because they are harder to recover from. We mostly recover from isolated packet errors using error correcting and redundancy coding schemes.

We now present an outline of the procedure. From the per-packet SNR values of the captured packets, a first-order interpolated SNR sequence for each bit of the packet is generated. The hypothesis is that the interpolated SNR values are those that bits of the packets face. We then use the SNR–BER curves [9], as shown in Fig. 2, to get the bit error probability for each bit of the packet. This results in a packet error trace that is obtained after SNR interpolation and SNR–BER conversion. This trace is compared with the actual packet error trace obtained from the error flags. For this comparison, we plot the NKLD in Fig. 10. The low value of the divergence validates such an SNR-to-packet-error-conversion procedure.

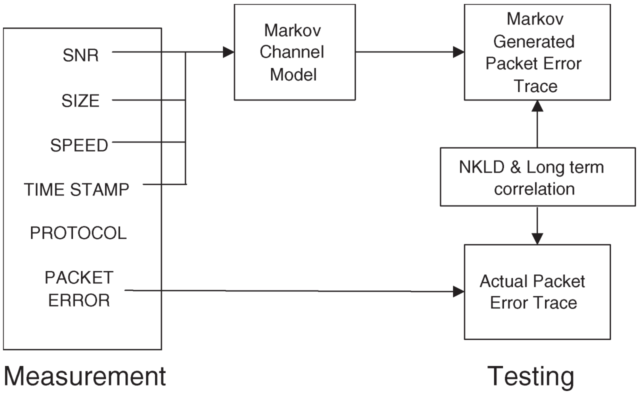
SNR
SIZE
SPEED
TIME STAMP
PROTOCOL
PACKET ERROR
Markov Channel Model
Markov Generated Packet Error Trace
NKLD & Long term correlation
Actual Packet Error Trace
Measurement
Testing

Fig. 11. Measurement and testing setup for a packet loss trace.

C. Burst Length Process

Now, we study the burst length process that can be derived from the channel model. Using (1) and Fig. 2, the SNR values yield the packet success probabilities that result in a packet loss trace, which we refer to as the Markov-generated packet trace. The setup is shown in Fig. 11.

We obtain the probability mass function of I and B from the observed packet error trace. Again, we compute the probability mass function of I and B from the Markov-generated packet error trace. The setup is shown in Fig. 11.

We compare the NKLD between the actual trace and the Markov trace. Specifically, we observe the variation in the NKLD as the number of states in the model is increased. In Fig. 12(a), we observe a very slight reduction in the NKLD measure as the number of states is increased. This is because, although the variation in the SNR is significant, the actual SNR values are high, so that the packet success rate is high. However, for intermediate and poor channels, the SNR variation is in a range where the packet success probabilities vary. It is in these regions that we need a higher granularity to define the state space of the SNR process.

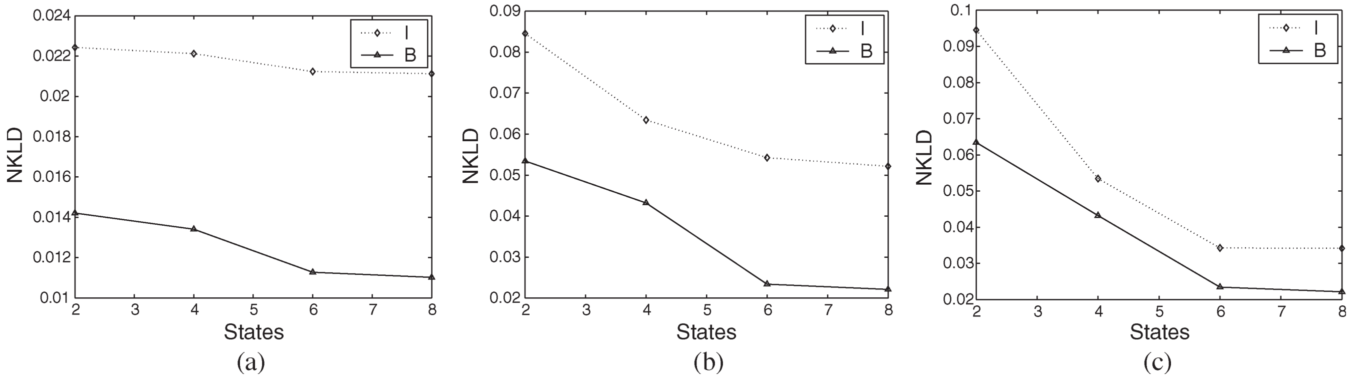

Fig. 12. NKLD measure for burst lengths from actual and Markov-generated packet traces for different kinds of channels. (a) Good channel. (b) Poor channel. (c) Intermediate channel.

The NKLD measure in Fig. 12 now includes the discrepancy accrued due to the SNR-trace-to-packet-error-trace conversion, as well as the model for the SNR variation. We attempt to identify the contribution of each of these two factors toward the NKLD. We note the packet error trace directly obtained from SNR values and the actual packet errors observed in Fig. 10. The NKLD values do not significantly increase in Fig. 12 when compared with Fig. 10. Hence, this indicates that it is the SNR-trace-to-packet-error-trace evaluation that results in the discrepancy rather than the model for the SNR variation.

D. Long-Term Correlation

We study the performance of the Markov model in terms of capturing long-term correlation, i.e., the conditional probability that packet $n + k$ is in error given that packet n is in error. In Fig. 13, we plot the performance of the Markov model in tracking long-term correlation. As shown in the plot, a Markov chain with eight states performs significantly better than that with two states. However, both models are not able to capture the long-term correlation with a discrepancy of more than 7%, which occurs because the actual trace is nonstationary and no stationary model can capture the variations in a long run. The empirical average does not deviate from the actual average by more than 2% with 95% confidence.

We now present evaluations for studying the performance of the channel model on traces collected from traffic sent by the users.

V. Testing of the Model Based on User Traces

We proceed to evaluate the efficacy of the finite-state Markov model in modeling the reverse channel, i.e., the users to the AP. So far, the study has been carried out based on traces collected from the traffic sent by the AP. The forward and reverse channels are known to be asymmetric. We now perform a study when user traces are collected at the AP to investigate the behavior of the reverse channel. Here, packets received from a specific user MAC address are examined for evaluation purposes. We demonstrate that similar conclusions hold when the user traffic is collected as compared to when the AP traffic is collected. As expected, the parameters of the model obtained in this case are different from those for traffic from the AP. Nevertheless, the statistical match between the model and the empirical traces is still satisfactory, and other general conclusions are also the same.

The difference between the SNR values from the actual trace and the trace obtained from the channel model is now examined. The root-mean-square percentage difference is observed to be less than 4%. We also evaluate the NKLD between the probability mass function of the SNR obtained from the measured trace and the Markov model. The probability mass function defines the probability of the SNR lying in certain intervals. The setup is shown in Fig. 4. The NKLD measure is plotted for three kinds of channels in Fig. 14 as the number of states increases. The low value of the NKLD measure signifies the proximity of the two distributions.

Using a Markov model with arbitrary transitions, as shown in Fig. 15, does not yield any significant improvement in the model performance. Here, unlike in Section IV, we do not perform error burst testing. This is because the SNR–BER curves of the AP chipset and the AP antenna characteristics are not available. We now present the evaluations.

We note that network events such as load variation and mobility influence both the forward (to the AP) and reverse (from the AP) channels. The channel models obtained for the forward and reverse channels have different parameters. However, the effectiveness of the corresponding Markov models in terms of the divergence measure is not different.

VI. Probe-Based Measurement

In the previous sections, we have focused on traces collected as a result of passive monitoring of the network. Such measurements require monitoring the network on a large timescale due to highly varying nature of the traffic. Additionally, the traffic rate can be very low for the collected trace to be meaningful in capturing the channel statistics. In this section, we investigate the behavior of our model when active measurements are used for trace collection. Here, probe traffic is generated in different ways, and corresponding traces are collected. Although active measurements influence the channel due to the interference generated by the probe traffic, such measurements can be very useful for quick channel evaluation due to its adaptability.

The difference between the active and passive measurements will result because of the presence of a probe traffic in the active

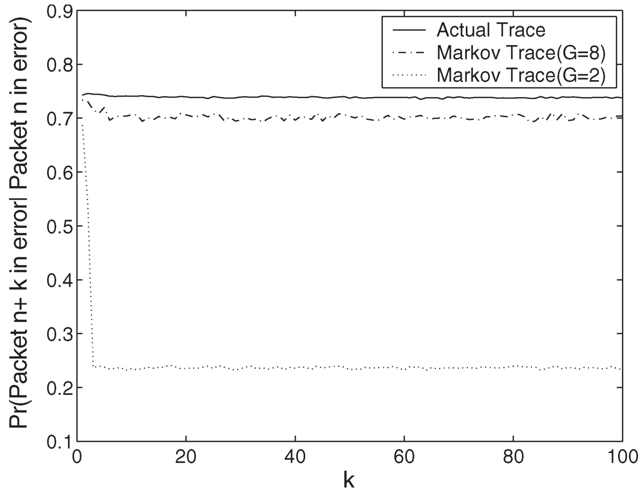

Fig. 13. Conditional probability that packet $n + k$ is in error given that packet n is in error.

measurements. To investigate the effect of the probe traffic and how they influence the Markov channel model, we consider three scenarios. The first scenario is an isolated setup to perform the measurement under controlled conditions. Thereafter, we consider two setups to collect traces in the presence of traffic from other user.

The isolated setup is shown in Fig. 16. The wired workstation T uses multi-generator to generate UDP traffic destined for L. The client C is receiving UDP traffic from the AP at the rate of 2 Mb/s (500 packets/s with 512-byte packets).

We first observe the influence of the packet rate on T–AP–L on the traffic from AP–C. As the rate on T–AP–L is increased, there is an increase in the interpacket time observed for the traffic from AP to C (Fig. 17). However, the throughput suffers only when the total packet load exceeds a certain threshold, as shown in Fig. 18. We want the probe rate to be sufficiently low so as not to influence the existing user sessions.

We then develop the model based on the packet trace from link AP–L. The packet transmission time, i.e., T_p, is the same, irrespective of the number of packets that are generated on the link T–AP–L since the transmission rate is 11 Mb/s. We investigate the sensitivity of the Markov model when the probe traffic rate is changed. The investigation is essential since, if the model parameters were to substantially change with the change in the probe rate, then the probe rate ought to be carefully determined. We vary the probe rate in the range 1–6 Mb/s with steps of 1 Mb/s and develop the Markov model for the SNR measurements obtained at each rate. We next summarize our observations. The SNR thresholds for the models are identical, with less than 1% change in the transition probabilities. The maximum difference between the NKLD is less than 2%. Our measurements, however, indicate that the model parameters do not significantly change with the change in the probe rate.

We now explain our observations. The model parameters may be different for different rates for the following two reasons. First, the SNR process could itself change as the rate is changed. This is because when the probe packets are more frequently transmitted, other users will also more frequently back off. We now explain why this effect is not pronounced.

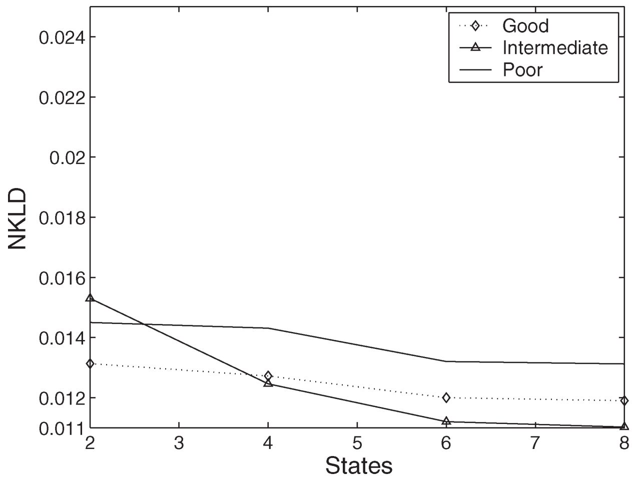

Fig. 14. SNR distribution.

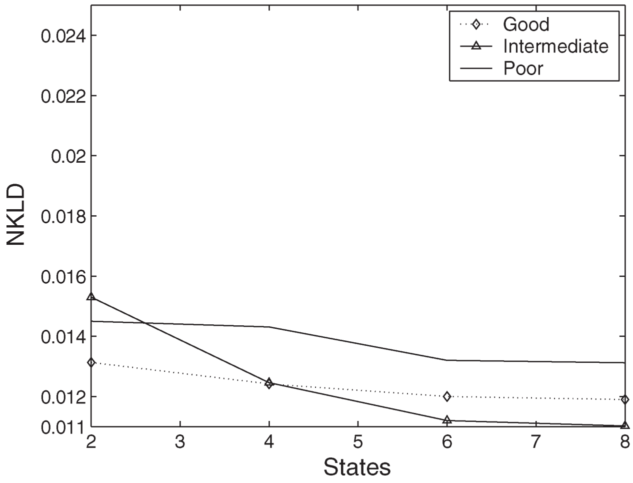

Fig. 15. SNR distribution with a Markov chain having arbitrary transitions.

First, note that when the existing aggregate rate on the channel is high, an increase in the probe traffic generation rate does not translate to a higher injected probe rate. This is because, although the traffic generation rate is increased, the probe traffic injected into the network, in the presence of an already existing user, does not increase beyond 3 Mb/s. For a higher number of users when the aggregate existing traffic is more than 4 Mb/s, the change in the probe rate injected is even lesser. On the other hand, when the existing aggregate rate in the channel is low, the injection of probe traffic does not significantly disrupt the transmission of other users and, hence, does not substantially alter the SNR process.

Second, even if the process does not change, the model statistics can change since the memory is likely to increase as the SNR is more frequently measured with a higher probe rate. However, even for a moderate probe rate of 1 Mb/s, the model exhibits high memory (> 0.95). Therefore, the scope for a substantial increase in memory with an increase in the probe rate is limited. Thus, this effect is not pronounced either.

In this situation, the performance of the model in characterizing the burst process is shown in Fig. 19.

In APs with connected users, sending additional streams will influence the performance observed by the other users. In

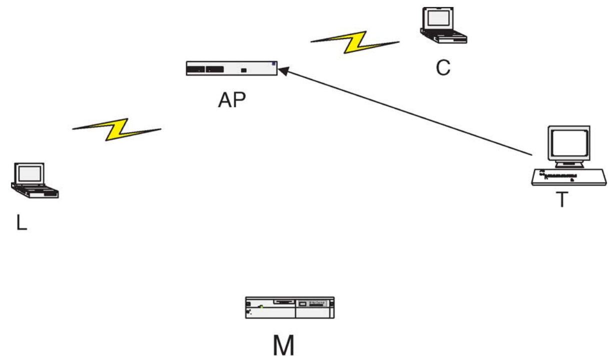

Fig. 16. Client C is associated with AP and receiving UDP traffic. Laptop L is in promiscuous mode associated with AP. The wired workstation T is sending UDP traffic to L. M is in monitor mode, listening to all the wireless traffic.

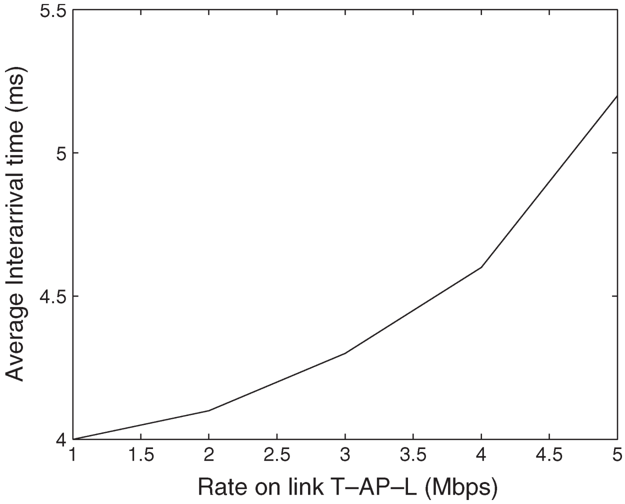

Fig. 17. Variation of the interarrival time for AP–C.

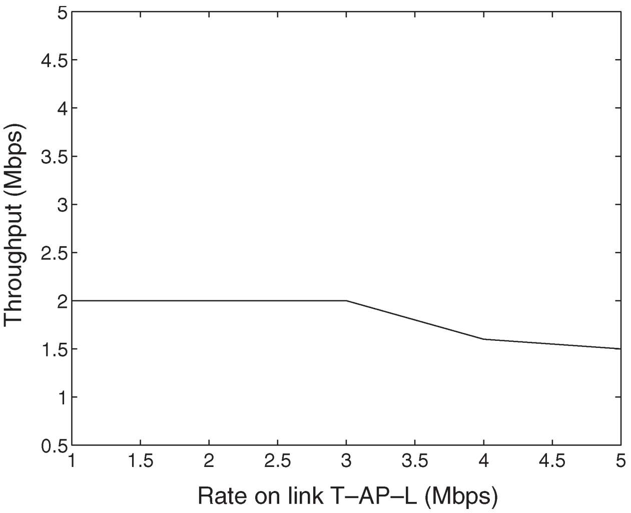

Fig. 18. Variation of the throughput for AP–C.

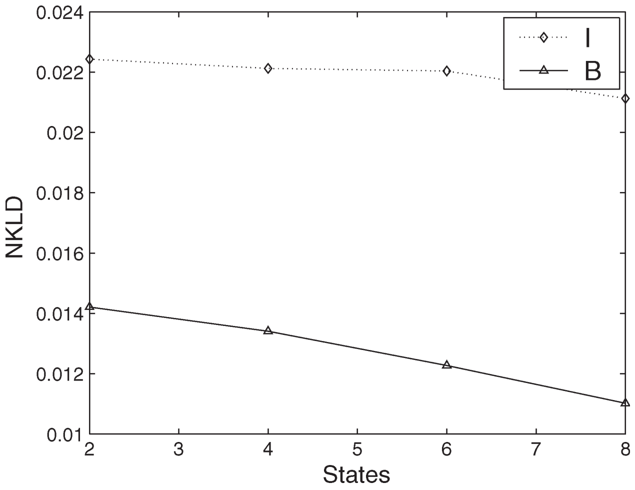

Fig. 19. Measurement from the probe packet stream.

practice, there are two ways to actively measure the channel characteristics using probe packet streams, and we investigate these methods. The first method is to send packets to an AP and measure the channel characteristics from the response packets received from the AP measured by the monitor placed near L. In Fig. 16, this corresponds to the case when the node L communicates with the AP and receives ACK packets. In this scenario, we vary the traffic rate from 1 to 6 Mb/s and develop the Markov model for each probe traffic rate. The transmission rate does not increase beyond 3 Mb/s due to the already-existing users on the network. The difference in the model parameter values is less than 5% as compared with the models developed in Section IV, and the divergence measure between the models is close to 0.01. From the perspective of the Markov model, the influence on the channel because of the probe traffic sent by the measuring station is not any different than the effect of other users on the network. The other users could be changing their rates, or they could be leaving and joining the network.

Another method is to make the AP send packets to a mobile node using a stream generated from the wired side of the AP. This scenario corresponds to the link T–AP–L in Fig. 16. We vary the rate of the traffic sent from the wired side from 1 to 6 Mb/s and observe the channel characteristics measured by the monitor M placed near L. The transmission rate does not increase beyond 3 Mb/s due to the already existing users on the network. This method of obtaining the channel characteristics yields identical results as compared to the case when the traffic destined for an arbitrary user is captured and used to generate characteristics (Fig. 20). We now explain the similarities. In Fig. 20, we evaluate the difference in the channel model when an additional probe stream is introduced into the network with a certain number of existing users as compared with the models derived from passive measurements. We observe that the NKLD increases with the number of users. This can be explained as follows: For few users, the probe traffic can easily be accommodated, resulting in a little change for traffic patterns of the existing users. When the wireless AP is functioning close to capacity with a large number of users, the effect of adding an additional probe stream is more pronounced. However, even in this case, with a large number of users, although the overall influence is high, the effect per user is mitigated, as can be observed from the slight increase in divergence in Fig. 20. The effect of varying the rate is similar to the previously explained observations. We conclude that in common APs, additional probe traffic does not significantly influence the statistics as captured by the Markov model.

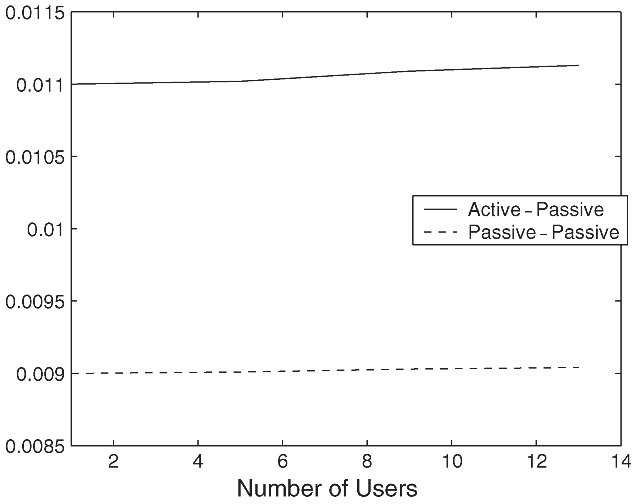

Fig. 20. Influence of the probe packet stream on the channel model. The active–passive shows the divergence between the channel models developed from the probe stream and the original passive stream collected from a particular user. The passive–passive shows the divergence between the traffic collected from the user with and without the probe traffic collected within a 5-min window.

We now discuss some inferences that we make from the model in different 802.11 scenarios.

VII. Inferences From the Model

Based on our measurements, we proceed to identify some features of the model that make it useful for channel representation. Our work is complementary to efforts of Henderson *et al.* [8], who have conducted measurements primarily from an application-layer perspective. They study large-scale characteristics such as application mix, building traffics, and mobility patterns, whereas we focus more toward the PHY aspects and their characterization.

An important parameter of the model is G, i.e., the number of states to represent the channel. Specifically, the value of G, i.e., the number of states, needs to be high enough, and using $G = 8$ resulted in a good match for all the three types of channels that we identified in Section III. High values of G would, however, increase the computational complexity. Based on model generation from traces collected on different days and times, we notice some broad trends in terms of the 802.11 channel behavior. SNR variation is a reasonably good indicator of the channel quality and packet loss in an 802.11b AP-type network. We obtained the following channel state success probabilities α: For poor channels $(\text{SNR} < 10\ \text{dB})$, $\alpha = [0.0000\ \ 0.1057\ \ 0.1904\ \ 0.3019\ \ 0.4297\ \ 0.5595\ \ 0.6782\ \ 1.0000]$ with error rates of 13%; for intermediate channels (SNR 10–20 dB), $\alpha = [0.0000\ 0.6785\ 0.7772\ 0.8535\ 0.9083\ 0.9453\ 0.9689\ 1.0000]$ with error rates of 8%; and for good channels $(\text{SNR} > 25\ \text{dB})$, $\alpha = [0.0002\ \ 0.9980\ \ 0.9980\ \ 0.9991\ \ 0.9996\ \ 0.9996\ \ 0.9999\ \ 1.0000]$ with error rates of 1.5%. These data can be used to explain the discrepancy between the results reported in prior works. Recall that in [1] and [19], the authors conclude that the two-state Gilbert–Eliot model is not suitable for characterizing packet losses, whereas the authors in [10] claim that two-state models are sufficient for capturing the packet loss process but not the bit losses. Results in [1] and [19] are based on measurements in error-prone channels, whereas those in [10] are based on good channels. We note from α for good channels that using fewer states (e.g., 2) can result in a close match since most of the packet success levels are high, as has been independently observed in [10]. However, this is not true for intermediate and poor channels since variation in the success probabilities of different states is significant. Therefore, Arauz and Krishnamurthy [1] and Willig *et al.* [19], who have studied channels with higher mean burst lengths and error rates, have not observed a good match with a two-state model.

Memory	8-10 a.m	10-12 a.m	12-2 p.m	2-5 p.m
Good	0.8834	0.9327	0.9114	0.9234
Intermediate	0.8815	0.9215	0.9005	0.9107
Poor	0.8231	0.8145	0.8576	0.8435

Fig. 21. Variation in memory with an eight-state model. The locations for good, intermediate, and poor are a classroom, a hotspot, and a park, respectively.

Memory	Monday	Wednesday	Friday	Saturday
8-10 a.m	0.8834	0.8823	0.8814	0.9435
12-2 p.m	0.9114	0.9232	0.9146	0.9431
2-5 p.m	0.9234	0.9217	0.9132	0.9482

Fig. 22. Variation in memory at a classroom AP.

Memory	Monday	Wednesday	Friday	Saturday
8-10 a.m	0.8815	0.8807	0.8814	0.9234
12-2 p.m	0.9005	0.8984	0.9034	0.8914
2-5 p.m	0.9107	0.9136	0.9064	0.8642

Fig. 23. Variation in memory at a T-mobile hotspot.

Memory	Monday	Wednesday	Friday	Saturday
8-10 a.m	0.8231	0.8315	0.8326	0.8444
12-2 p.m	0.8576	0.8424	0.8616	0.8329
2-5 p.m	0.8435	0.8623	0.8458	0.8224

Fig. 24. Variation in memory at a public park.

We studied the variation in channel memory, i.e., $1 - P_{k,k+1} - P_{k,k-1}$ (Fig. 3), with different times of the day and across different days. The channel memory would indicate how rapidly a wireless channel varies. Depending on the variation speed of the channel, algorithms can possibly adjust recomputation to minimize the overhead. The memory of the channels is observed to vary in the range 0.8–0.95. In Fig. 21, we tabulate the memory of the eight-state model at different environments with time on a weekday (Monday). Over time (3–4 h) on the same day, the channel quality significantly varies, i.e., shifts from "good" to "intermediate," and *vice versa*, as the user loads are altered. This indicates that the number of users and usage patterns plays a major role in determining the channel characteristics.

In general, a strong similarity between parameters (variation $< 2\%$) was observed in terms of the same time at different days at a given location (Figs. 22–24). This can be attributed to similar usage patterns. In addition, because of recurring user patterns at the same time over different days, models with similar parameters can again be utilized. A memory value of 0.8 is shown in locations with a high number of users, whereas very high memory is shown in "good" channels with very low loss rates. Due to the high memory of the channel, resource allocation algorithms can possibly make decisions less frequently to reduce overhead.

Apart from user patterns, different environments also play a major role. In open environments with high delay spread, we observe a loss rate increase ("good" to "intermediate") with increasing distance that cannot be explained simply based on attenuation of signal levels. Here, the loss rate increases as a result of multipath effects, as has been pointed before, and the channel attains "intermediate" loss rates, despite the low number of users.

Overall, characterizing the SNR gives a good insight of the channel characteristics and helps explain behavior that are not directly answered from packet error traces. The proposed model for characterizing the SNR captures the channel variation with a reasonable degree of accuracy in terms of SNR and burst length distributions. It can serve as a useful model for analytical scenarios that require tractable models and for computational scenarios.

VIII. Conclusion

We have investigated a model for characterizing SNR variations of an 802.11 channel. The model is simple, analytically tractable, and easy to characterize using measured traces. We have discussed an approach to gather the information required to parameterize such a model based on measurements taken from 802.11b AP networks using common hardware. The model is found to represent the packet loss process with reasonable accuracy. The model maintains the birth–death flavor of a two-state model and, at the same time, significantly improves the performance. Such a model that tracks the temporal variation of the SNR can be useful for a variety of resource allocation algorithms and large-scale simulations that might require low complexity models.

References

[1] J. Arauz and P. Krishnamurthy, "Markov modeling of 802.11 channels," in *Proc. IEEE Veh. Technol. Conf.*, 2003, pp. 771–775.

[2] Atheros, *Atheros WLAN Chipset*. [Online]. Available: http://www.atheros.com

[3] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," *Perform. Eval. Rev.*, vol. 30, no. 1, pp. 195–205, Jun. 2002.

[4] G. Bianchi, "Performance analysis of IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 1991.

[6] E. O. Elliott, "Estimates for error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, no. 9, pp. 1977–1997, Sep. 1963.

[7] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 9, pp. 1253–1265, Sep. 1960.

[8] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proc. ACM Mobicom*, 2004, pp. 187–201.

[9] Intersil, *ISL3874A; Wireless LAN Integrated Medium Access Controller with Baseband Processor With Mini-PCI Features*, Aug. 2001.

[10] S. A. Khayam and H. Radha, "Markov-based modeling of wireless local area networks," in *Proc. 6th ACM Workshop Modeling Anal. Simul. Wireless Mobile Syst.*, 2003, pp. 100–107.

[11] M. Koes, B. P. Sellner, B. Lisien, G. Gordon, and F. Pfennig, "A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data," in *Proc. IJCAI*, 2003, pp. 1427–1428.

[12] Z. Kong, D. H Tsang, B. Bensaou, and D. Gao, "Performance analysis of IEEE 802.11e contention-based channel access," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 10, pp. 2095–2106, Dec. 2004.

[13] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.

[14] M. Milner, *NetStumbler*. [Online]. Available: http://www.netstumbler.org

[15] G. T. Nguyen, B. Noble, R. H. Katz, and M. Satyanarayanan, "A trace-based approach for modeling wireless channel behavior," in *Proc. Winter Simul. Conf.*, 1996, pp. 597–604.

[16] H. S. Wang and N. Moayeri, "Finite-state Markov channel—A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[17] WildPackets, *AiroPeek NX 2.0.5*. [Online]. Available: http://www.wildpackets.com

[18] A. Willig, "A new class of packet- and bit-level models for wireless channels," in *Proc. IEEE PIMRC*, 2002, pp. 2434–2440.

[19] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Trans. Ind. Electron.*, vol. 49, no. 6, pp. 1265–1282, Dec. 2002.

[20] Y. Xu and W. Lee, "Exploring spatial correlation for link quality estimation in wireless sensor networks," in *Proc. IEEE PERCOM*, 2006, pp. 200–211.

[21] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

Ratul K. Guha received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, New Delhi, India, in 2002 and the M.S. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania, Philadelphia, in 2004 and 2007, respectively.

He is currently a Senior Research Scientist with Telcordia Technologies, Piscataway, NJ. His research interests include wireless networks, vehicular communications, computer security, and distributed control.

Saswati Sarkar (S'98–M'00) received the M.Eng. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1996 and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in 2000.

She is currently an Associate Professor with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia. Her research interests include resource allocation and performance analysis in communication networks.

Prof. Sarkar served as an Associate Editor for the IEEE Transactions on Wireless Communications between 2001 and 2006. She was the recipient of the Motorola Gold Medal as the Best Master's Student with the Division of Electrical Sciences, Indian Institute of Science, in 1996 and the National Science Foundation Faculty Early Career Development Award in 2003.