Improving Wireless Link Simulation Using Multilevel Markov Models

ANKUR KAMTHE, MIGUEL Á. CARREIRA-PERPIÑÁN, and ALBERTO E. CERPA,
University of California - Merced

Modeling the behavior of 802.15.4 links is a nontrivial problem, because 802.15.4 links experience different level of dynamics at short and long time scales. This makes the design of a suitable model that combines the different dynamics at different time scales a nontrivial problem. We propose a novel multilevel approach, the M&M model, involving hidden Markov models (HMMs) and mixtures of multivariate Bernoullis (MMBs) for modeling the long and short time-scale behavior of wireless links from 802.15.4 test beds. We characterize the synthetic traces generated from our model of the wireless link in terms of the mean and variance of the packet reception rates from the data traces, comparison of distributions of run lengths, and conditional packet delivery functions of successive packet receptions (1's) and losses (0's). Our results show that when compared to the closest-fit pattern matching model in TOSSIM, the proposed modeling approach is able to mimic the behavior of the data traces quite closely, with differences in packet reception rates of the empirical and simulated traces of less than 1.9% on average and 6.6% in the worst case. Moreover, the simulated links from our proposed approach were able to account for long runs of 1's and 0's as observed in empirical data traces.

Categories and Subject Descriptors: I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms: Algorithms, Measurement, Performance, Design, Experimentation

Additional Key Words and Phrases: Simulation, 802.15.4 low-power wireless networks, wireless channel model, hidden Markov models, mixture of multivariate Bernoulli

ACM Reference Format:
Kamthe, A., Carreira-Perpiñán, M. Á., and Cerpa, A. E. 2013, Improving wireless link simulation using multilevel Markov models. ACM Trans. Sensor Netw. 10, 1, Article 17 (November 2013), 28 pages.
DOI: http://dx.doi.org/10.1145/2529991

1. INTRODUCTION

The common denominator in all wireless sensor networks (WSNs), regardless of their underlying application, is the use of the radio to communicate information extracted from the sensed environment and, more importantly, to coordinate with other nodes. Consequently, radio communication and intelligent usage of the radio is a critical component of wireless distributed system in general and WSNs in particular. Due to the

A prior version of this manuscript "M&M: Multi-level Markov Model for Wireless Link Simulations" appeared in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*.
This work was supported in part by the Department of Energy and the California Energy Commission, and performed under U.S. Department of Energy Contract No. DE-AC02-05CH11231 and the California Institute for Energy and the Environment under contract No. MUC-09-03.

Authors' addresses: A. Kamthe, M. Á. Carreira-Perpiñán, and A. E. Cerpa, 5200 N. Lake Road, School on Engineering, University of California—Merced, CA 95344; corresponding author's email: akamthe@ucmerced.edu.

DOI: http://dx.doi.org/10.1145/2529991

low-power nature of WSNs, the radio used for communication is especially susceptible to changes in the quality of the wireless medium, resulting in packet losses which can be attributed to limited transmission power levels as well as multipath effects resulting from lack of frequency diversity. Experiments [Zhao and Govindan 2003] conducted with these low-power radio-equipped sensor nodes have shown, using empirical measurements, that there exists a “gray area” within the communication range of sensor radios where the packet reception varies widely. Data collected using SCALE [Cerpa et al. 2003] led to the following conclusions: (i) no clear correlation between packet delivery and distance in an area of more than 50% of the communication range, (ii) temporal variations of packet delivery are correlated with mean reception rate of each link, and (iii) percentage of asymmetric links in a sensor network varies from 5% to 30%. These studies [Zhao and Govindan 2003; Cerpa et al. 2003, 2005; Srinivasan et al. 2006, 2008, 2010] help confirm that low-power wireless communication is unpredictable, sensitive to changes in the environment, and known to significantly change over different time scales.

In systems research, a well-designed simulator provides users with the ability to test new ideas in an inexpensive manner. The simulator models the key elements of a given system, for example, hardware, such as the CPU, network interfaces, sensors, etc., and software, such as the operating system. This allows the user to focus his attention on the design, testing, and analysis of algorithms in a controlled and repeatable environment. Recent studies [Pawlikowski et al. 2002; Kotz et al. 2004] have indicated the presence of a wide chasm between real-world radio channel behavior and existing radio channel models in wireless simulators. This leads to significant differences in performance of a system in simulation as compared to a real-world deployment. Thus, improving wireless simulators by incorporating accurate and robust radio channel models will reduce the gap between simulation and real-world performance. To reach this goal, we believe it is required to collect data traces of packet reception information over long periods of time at fine granularity. This data would be the seed for creating radio channel models that would help simulate more realistic packet losses, thus helping application designers increase the robustness of their applications by accounting in simulation for losses in the wireless medium.

Existing sensor network simulators [Levis et al. 2003; Levis and Lee 2003; Girod et al. 2004] consider very elementary channel models, as will be described in Section 2. Studies [Lee et al. 2007] have tried to complement these models with additional information in the form of noise data, but still these models inadequately and inaccurately characterize the spatial and temporal variations in the radio channel over different time scales. The problem of existing wireless simulators is their inability to simultaneously model the effect of changes in wireless communication over short and long time-scales and over distance. To create accurate models for simulating the wireless channel, it is essential to collect data from the target deployment area. In cases where this is not possible, the simulator should be able to recreate realistic channel conditions from a comprehensive database of models created before hand.

In this article, we propose a novel multilevel approach involving hidden Markov models (HMMs) and mixtures of multivariate Bernoullis (MMBs) for modeling the long and short time-scale behavior of links in wireless sensor networks, that is, the binary sequence or trace of packet receptions (1’s) and losses (0’s) in the link. In this approach, the transition matrix of an HMM models the long-term evolution of the trace (level 1) as transitions among a set of unobserved, level-1 states. These states typically correspond to a roughly constant packet reception rate (as determined by the data) and might correspond to different regimes of the link. Within each level-1 state, the short-term evolution of the trace (level 2) is modeled by an MMB emission distribution for each state of the HMM. This captures the faster, but not random, variations of the sequence of packet receptions and losses. We characterize the synthetic traces generated from the model in terms of several statistical measures: moments (mean and variance) of the distribution of packet reception traces, run-length distributions of packet receptions and of packet losses, and conditional packet delivery functions (CPDFs). To compare run-length and CPDF distributions, we designed a new metric called the Nearest-Neighbor Distance. This metric aims to solve the problem of comparing distributions with unequal supports. In addition, a full implementation of the M&M model for the TOSSIM simulator is provided.

The rest of the article is organized as follows. A review of related work is provided in Section 2. In Section 3, we identify issues that need to be addressed to resolve the deficiencies in link models for WSN simulators. A new Markov-based modeling approach for addressing these issues is proposed in Section 4. In Section 5, we evaluate the performance of our proposed approach. Section 6 contrasts the modeling of links in existing WSN simulators against our proposed approach. In Section 7, we discuss issues related to our modeling approach and, finally, in Section 8, we summarize our results and discuss future work.

2. BACKGROUND AND RELATED WORK

Models for characterizing the behavior of wireless links have been a widely studied area in networking literature [Rappaport 2001]. These studies can be classified into radio propagation models and packet loss models. Radio propagation models predict the average received signal strength and its variability at a given distance from the transmitter. They can be further subdivided into large-scale propagation models and small-scale propagation models [Rappaport 2001]. In practice, radio propagation models are created from a combination of analytical and empirical methods based on actual field measurements. The field measurements help incorporate the effects of known and unknown factors in the measured environment. One of the most common radio propagation models is the log-normal shadowing model [Rappaport 2001]. In contrast, packet loss models try to discover the underlying bursty packet loss distribution. Errors in packet reception can be attributed to causes, such as interference in the channel and fading effects which lead to irrecoverable bit errors. Packet loss models can be broadly classified into two areas: (a) packet delivery function estimation approaches and (b) Markovian approaches.

2.1. Packet Delivery Function Estimation Approaches

Reis et al. [2006] noted the fact that in wireless networks, measurements of average behavior over even relatively short time periods tend to be stable, even for widely separated intervals. They exploit this to develop models for wireless delivery with interference using RSSI measurements. They model delivery probability as a function of interference, which the authors posit is the prime cause of variation in packet delivery. The receiver model is used with the RF profile to compute that the probability that a packet is correctly received in the presence of competing transmissions. In experimental evaluations, the approach has an RMS error of 0.5 and 0.3 for 802.11a and 802.11b networks, respectively, compared to a model that ignores interference. Similarly, Reddy and Riley [2007] utilized RSSI and packet retry measurement values to create a radio propagation and packet error rate model, respectively.

Kashyap et al. [2008] proposed a measurement-based approach to model the physical layer behavior, mainly, radio propagation, carrier sense, and packet reception models. Radio propagation is modeled using a log-normal shadowing model. A carrier sense model is created using a function fitted to measurements of received signal strength between a pair of nodes at different locations. Packet reception is modeled by curve-fitting of packet reception probability and signal to noise ratio data. In experiments conducted on a 12-node test bed, using the proposed approach, there was a 10% difference between measured and predicted throughput capacity in contrast to a 50% difference in more traditional simulation models.

Lenders and Martonosi [2009] combine a physical receiver model with a MAC deferral and interference model. Here, the physical receiver model models the effects of radio propagation, environmental noise, and node mobility, whereas the MAC model predicts packet delivery in the presence of carrier sense and interference from concurrent transmissions. The physical receiver model utilizes the windowed average at each time instance to compute the packet reception probability. The MAC model uses the packet reception probabilities from the physical receiver model to simulate carrier sense (channel deferral). This approach was evaluated using five mobile 802.11 nodes competing for the same channel under different environment and mobility conditions. Across various settings, the root mean square error (RMSE) of the estimated versus measured benchmark packet probability was below 12 percent. The error of the estimated throughput versus the effective observed throughput was below ten percent, in contrast to up to >50 percent for a naive model that ignores carrier sense and interference effects.

The common theme in these approaches is that a function relating physical layer characteristics, such as RSSI or SINR, to packet reception is computed from measurement data. Using this empirical pdf of packet reception, the packet delivery probability is computed. The problem with such approaches is that packet errors are assumed to be independent which has been proven incorrect in empirical studies [Cerpa et al. 2005; Srinivasan et al. 2008] showing temporal correlations between successful and failed packet receptions. Burstiness behavior in packet delivery where packet receptions and losses shows high correlation is not captured in these models. Also, all of these models do not account for the fact that packet reception shows long periods of stability.

2.2. Markovian Approaches

The Gilbert model [Gilbert 1960] is a probabilistic model for simulating burst noise in data transmission channels. In this model, a hidden Markov model with two states is used to generate noise bursts the first state has zero probability of encountering an error, whereas the other state has a certain fixed nonzero probability for transmission errors. The transition probabilities control the amount of time spent in each state, thus controlling the error patterns from a given set of parameters. Analysis of traces [Nguyen et al. 1996] for the AT&T Wavelan system concluded that loss behavior could not be accounted by the two-state Markov model. They proposed a methodology to model the error-free and error traces using exponential and Pareto distributions to model segments of the trace. Traces modeled [Towsley et al. 1999] from measurements of Internet packet loss compared between a Bernoulli model, two-state Markov chain model, and *k*th order Markov chains to check for the accuracy of the loss estimation over 38 stationary trace segments. They concluded that all these models are inadequate as they could not accurately model losses in their dataset.

Markov-based Trace Analysis (MTA) [Konrad et al. 2001] and Multiple MTA [Konrad et al. 2006] approaches propose modeling channel errors by decomposing a trace with nonstationary properties into a set of piecewise stationary traces consisting of *lossy* and *error-free* states. Lossy states exhibit stationarity, where a sequence of lossy states can be modeled by a traditional discrete time Markov chain (DTMC). In Salamatian and Vaton [2001], HMMs were proposed for modeling packet reception traces and choosing a model based on the likelihood criterion. Markov-based stochastic chains were proposed [Khayam and Radha 2003] to model 802.11b channel behavior for bit errors and packet losses. The study compared the performance of high-order Markov chains, two-state Hidden Markov Models, and hierarchical Markov models and concluded that

Markov chains of order nine (i.e., 2^9 states) are required for accurate models for the bit error process. These studies helped reinforce the notion that for any modeling approach to simulate behavior of wireless links, the model needs to account for the long- and short-term variations in the link quality. Also, the model should be easy to train and show close correlation between the input and the simulated data traces. Further discussion of the aforementioned approaches and our proposed approach is provided in Section 7.

2.3. TOSSIM

TOSSIM [Levis et al. 2003] is a discrete event simulator for sensor networks running on the TinyOS operating system. It allows users to write TinyOS code in a simulation environment that is scalable and bridges the gap between algorithm testing and application development. TOSSIM simulates behavior of the CPUs, radios, and sensors in different sensor nodes, networking stacks, and other OS primitives.

TOSSIM supports several radio models, namely the Simple Propagation model, the Link Layer model [Zuniga and Krishnamachari 2004], and the CPM model [Lee et al. 2007]. In the Simple Propagation model, every node can receive packets transmitted by any other node. The Link Layer model specifies the behavior of the wireless link depending on the radio and the channel characteristics for static and low-dynamic environments. CPM is based on a statistical model created from noise reading traces collected from the deployment environment. It computes the probability distribution of n_t given the noise readings $(n_{t-k}, n_{t-k+1}, \ldots n_{t-1})$, where k is the duration of noise history considered by the model. A $k = 0$ would make each noise value independent, while k equal to the length of the trace would provide an exact replay of the noise trace. The CPM models helps capture the short-term link quality variation caused by factors, such as 802.11 traffic. In a recent paper [Rusak and Levis 2008], two approaches (Expected Value PMF and Average Signal Power Value) were proposed to estimate the signal power of missing packets in a packet reception trace, and using this data, the CPM algorithm models the variations in packet signal strength. These existing models require the modeling of two separate physical-layer measurements, namely, RSSI and noise/interference values to create a representative model of a real environment.

In contrast, in this article, we propose the modeling of correlations between successive packet receptions and failures from a given packet reception trace as the packet reception traces are a direct indicator of the link quality.

3. WIRELESS LINK MODELING

3.1. Collection of Packet Reception Traces

In order to create an accurate packet loss model, we required a comprehensive database of packet reception traces of links having different reception rates. For this task, we collected data from a 75-node MoteIV Tmote Sky test bed deployed along the ceiling of the Science and Engineering Building (SE test bed). Each mote is comprised of an ultra low-power Texas Instruments MSP430 F1611 micro-controller featuring 10 KB of RAM, 48 KB of flash, and an 802.15.4-compliant Chipcon CC2420 radio (channel 26) for wireless communication. The node locations are fixed for the duration of our experiments (refer to Figure 1 for details). All the motes in a group are connected to a Linksys NSLU2 network storage device via an USB hub. The Linksys NSLU2 device is used to bridge serial communication between the motes and a central server over the local network.

W-e performed a number of experiments to collect packet reception traces from a diverse set of links (see Table I). In each experiment, we have one fixed sender and multiple receivers. The sender sends 64 packets per second with an interpacket interval

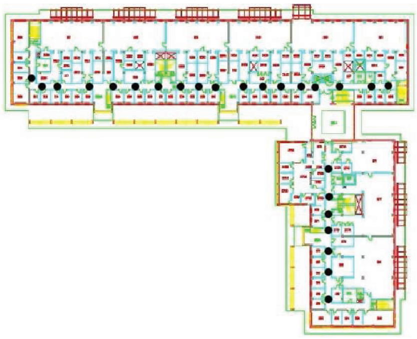


Fig. 1. SE test bed: 25 groups of three nodes each separated by a distance of 40 cm. Nodes are placed at fixed locations along the corridor ceiling of the building.

Table I. Summary of Experiments Conducted on the MoteLab and SE Test Beds

Testbed	Program	Num. Expts.	Duration	Num. Packets/Expt.	CC2420 Tx power levels
SE	RssiDemo	9	1 hour	230,400	1–31
SE	RssiDemo	1	6 hours	1,382,400	7
SE	RssiDemo	3	12 hours	2,764,800	8,9,11
SE	RssiSample	3	30 minutes	196,608	—
MoteLab	RssiDemo	18	30 minutes	115,200	31
MoteLab	RssiSample	3	30 minutes	196,608	—

Note: 802.15.4 channel 26 is used in all experiments.

(IPI) of 16 ms on channel 26 for durations of 1, 6, and 12 hours. The receivers record the sequence number, received signal strength (RSSI), and link quality indicator (LQI) values of each received packet. We also collected the same data from the MoteLab testbed [Werner-Allen et al. 2005], but the duration of each experiment was limited to 30 minutes due to storage concerns regarding the large amount of data generated in every run. After each experiment, we created records or traces of packet reception for each of the receiver nodes. In addition, we also gathered noise data (channel 26) for all nodes using the *RssiSample* program on both the test beds. The length of the noise traces is equivalent to the *meyer-heavy* trace collected in Lee et al. [2007]. The noise traces are meant to be utilized for a faithful comparison between the TOSSIM simulation model and our proposed approach.

3.2. Exploratory Data Analysis

In this section, we highlight issues that need to be addressed when modeling 802.15.4 wireless links. We term links having packet reception rate (PRR) $< 10\%$ as bad or poor links, links having PRR between 10% and 90% as intermediate links, and links having PRR greater than 90% as good links. Links having PRR $= 0\%$ are termed as inactive links.

Prior studies have shown that 802.15.4 links can vary significantly over time [Cerpa et al. 2003, 2005; Lin et al. 2006; Srinivasan et al. 2006, 2008, 2010; Rusak and Levis 2009]. In Figure 2, the average network throughput per hour averaged over all the links having $PRR > 10\%$ in the network is shown as a function of time of day for the 12 hour experiments. The figure clearly shows that the average network throughput is not constant, but fluctuates with time. This is a clear indication of variation of PRR across nodes in the network. The radio transmission power levels in experiments 24, 26, and 28 correspond to values 11, 10, and 8 in the CC2420 registers. This would lead one to think that the throughput should be highest for experiment 24, followed by level 26 and

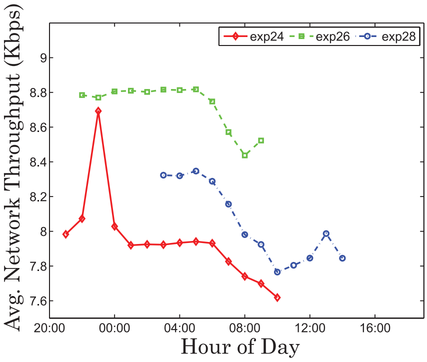


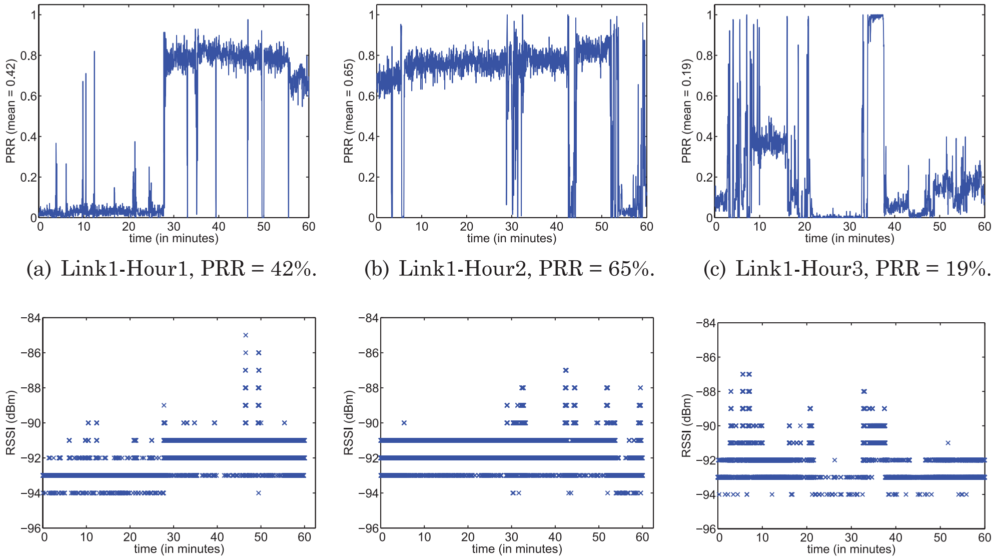
Fig. 2. Variation of the average data throughput per hour for all good and intermediate links in the network.

Table II. Summary of Variation of Link Quality in a Network as a Function of Sender Radio Transmission Power

Exp. #	CC2420 Tx. Lvl.	Good	Bad	Interm	Inactive
24	11	20 (48%)	8 (19%)	7 (17%)	7 (16%)
26	10	19 (45%)	7 (17%)	3 (7%)	13 (31%)
28	8	19 (45%)	10 (24%)	3 (7%)	10 (23%)

28, respectively. However, from the data, we see that the throughput for experiment 24 is less than that of the others. This can be explained by the higher total number of intermediate links (see Table II) compared to the other experiments. An interesting artifact of the environment can be seen in Figure 2, which shows a fairly consistent decrease in throughput from midnight to midday in all three experiments. From our experimental data, we observed that good and bad links are relatively stable over time, whereas intermediate links show significant variation in link quality over time. This is consistent with previous findings reported [Cerpa et al. 2003, 2005; Rusak and Levis 2009; Srinivasan et al. 2010]. In general, in simulation, it is easy to model good links, as they do not show significant variation with time [Metcalf 2007; Rusak and Levis 2008]. On the other hand, there is a significant difference between the models of intermediate links in simulation and the real world. If the accuracy of simulation models of these intermediate links were improved, then it is possible that WSN application simulations could show the potential benefits of using these intermediate links when their quality is high enough for transporting data instead of permanently ignoring or blacklisting them. In addition, it would help application designers to test performance of algorithms for the common case and the corner cases that are one of the causes of protocol failure.

To emphasize this point, we plot the variation in PRR and RSSI of a representative intermediate link. For this link (see Figure 3), the PRR is plotted as a function of time, where each PRR value is calculated for a two second interval (i.e., for 128 consecutive packets at a time). Figure 3 also shows the corresponding variation in RSSI values of the received packets. From Figure 3, we see that the average PRR of link 1 is 42%, 65%, and 19% for hours 1, 2, and 3, respectively, and the corresponding average RSSI values are -91.85 dBm, -91.8 dBm, and -91.67 dBm, respectively. In each hour, we see that the PRR and RSSI values fluctuates widely, cycling between good, intermediate, and bad states. In each state, the link is relatively stable for a given period of time before a significant change in link quality. A closer look at the sequence of received packets within a few tens of seconds reveals that packet receptions and losses are not independent, that is, intermediate links show significant bursty behavior. This behavior



(a) Link1-Hour1, PRR = 42%. (b) Link1-Hour2, PRR = 65%. (c) Link1-Hour3, PRR = 19%.

(d) Link1-Hour1, PRR = 42%. (e) Link1-Hour2, PRR = 65%. (f) Link1-Hour3, PRR = 19%.

Fig. 3. Variation in PRR and RSSI by hour for a typical intermediate quality link. Figure shows that PRR and RSSI values are stable for short periods of time.

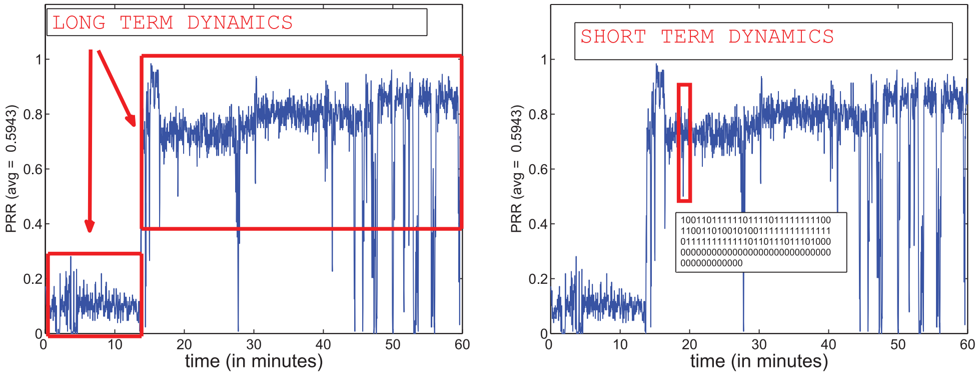
is similar to that previously reported [Rusak and Levis 2009; Srinivasan et al. 2010]. It shows that links of intermediate quality manifest highly dynamic behavior over time at different time scales, thus highlighting the nontrivial nature of the modeling problem for such links.

3.3. Our Modeling Approach

We consider our observed data as binary sequences, where 1 indicates successful packet reception and 0 indicates lost or corrupted packets. (We will also consider a sequence of continuous values, namely the reception rates in [0, 1] indicating the average over a binary window.) The fundamental motivation for our modeling approach is that observed traces display structure at different temporal scales. In Figure 4, for example, one can see that over a period of minutes, the link seems to switch between two states: one with PRR ≈ 0.1 and the other with PRR ≈ 0.7. We call this *long-term dynamics*. In a period of seconds, however, while the PRR may stay roughly constant, it is more likely to observe a bursty sequence `0000111111` than a wildly oscillating sequence `1010101101`. We call this *short-term dynamics*. In order to simulate realistically the behavior of links, we want a model that is flexible enough to replicate this multiscale structure, and we want to estimate its parameters (which determine its typical PRRs or its local burstiness) from observed traces. In the next section, we describe the details of our model, the *multilevel Markov (M&M) model* (Appendices A and B give an overview of hidden Markov models and mixtures of multivariate Bernoulli distributions).

4. THE MULTILEVEL MARKOV (M&M) MODEL

We model a possibly infinite binary sequence (the data trace) as a *sequence of binary strings* (windows) $\mathbf{x}_t$ of length W, as shown in Figure 5. A hidden Markov model (HMM) with S different states $q = 1, \ldots, S$ models transitions between long-term states, and has S^2 tunable parameters (the transition probabilities $p(q_t = j|q_{t-1} = i)$). Each long-term state q has its own distribution $p(\mathbf{x}|q)$ of emitting binary W-windows, which



(a) (b)

Fig. 4. Illustration of long- and short-term dynamics in an empirical trace (avg. PRR = 59.43%). (a) Long-term dynamics are periods of nearly constant PRR (10% and 70%, respectively) which persist for periods in the order of minutes. (b) Short-term dynamics are the burstiness observed in packet delivery over a period of seconds, indicating that packet receptions and losses are highly correlated.

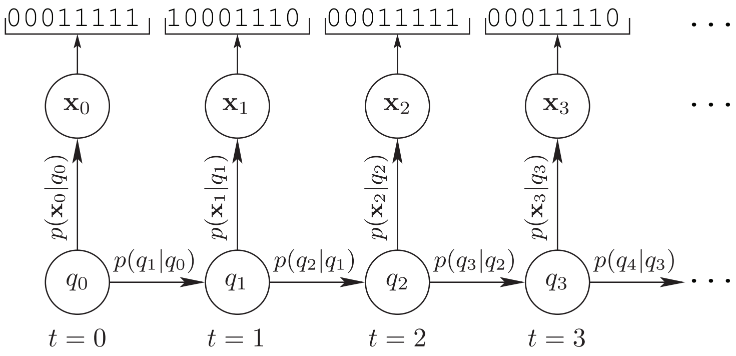


Fig. 5. Graphical model of a HMM which emits binary strings $\mathbf{x}_t$ of length 8. In the M&M model, $p(q_t|q_{t-1})$ is modeled by the transition matrix of the HMM, and $p(\mathbf{x}_t|q_t)$ is modeled using a MMB emission distribution for the HMM.

captures the short-term behavior of the link in that state, that is, the dynamics of the variations in consecutive packet reception successes or failures that has its own parameters (described next). Thus, W controls the trade-off of short versus long term. The emission distribution of the HMM $p(\mathbf{x}|q)$ is a *mixture of multivariate Bernoulli (MMB)*. For each state q, this mixture has M components, and each component has $W + 1$ parameters: a mixture proportion and a vector $\mathbf{p} = (p_1, \ldots, p_W)$ of Bernoulli parameters.

Next, we explain how to simulate a binary trace from our model (sampling) and how to estimate good model parameters from measured data (learning).

4.1. Sampling

In order to generate a trace of length L bits from the model, we sample as follows.

—Generate a long-term state sequence of length L/W using the transition probabilities of the HMM.

—For each long-term state q of this sequence, we sample a W-window $\mathbf{x}$ from its $p(\mathbf{x}|q)$ (i.e., the corresponding MMB emission distribution).

The trace is the concatenation of the L/W windows.

4.2. Learning: Hidden Markov Model Parameter Estimation

An HMM with MMB emission distribution (HMM–MMB) models the wireless data trace in terms of a sequence $q_0, q_1, \ldots, q_T$ of hidden (unobserved) random variables called states and a sequence $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T)$ of observed random variables (see Figure 5). The parameters of the HMM are the transition probability matrix $p(q_t = j|q_{t-1} = i)$, the MMB emission distribution $p(\mathbf{x}|q = i)$ for each state, and the initial state probability $p(q_0 = i)$. We jointly estimate these parameters using the expectation-maximization (EM) algorithm we derived for HMM–MMBs. We optimize the following Q function given the HMM–MMB model with its $S^2 + SM(W+1) + S$ free parameters (denoted Θ).[1]

$$Q(\Theta, \Theta') = \sum_{\mathbf{q} \in \mathcal{Q}} \sum_{\mathbf{m} \in \mathcal{M}} \log(P(\mathbf{X}, \mathbf{q}, \mathbf{m}; \Theta))(\mathbf{X}, \mathbf{q}, \mathbf{m}; \Theta'), \tag{1}$$

where $\mathcal{Q}$ is the set of all state sequences, $\mathbf{q}$ is a vector of length T representing a specific state sequence, $\mathcal{M}$ is the set of all component sequences compatible with $\mathbf{q}$, and $\mathbf{m}$ is vector of length T containing a component sequence corresponding with each state inside $\mathbf{q}$. Θ' are our estimates of the parameters in the previous iteration. $P(\mathbf{X}, q|\Theta)$ is the complete-data likelihood function given by

$$P(\mathbf{X}, \mathbf{q}, \mathbf{m}; \Theta) = p(q_0) \prod_{t=1}^{T} p(q_t|q_{t-1}) p(\mathbf{x}_t, m_t|q_t), \tag{2}$$

and, $p(\mathbf{x}_t, m_t|q_t)$ is the MMB emission distribution (Appendix B) for state q_t and component m_t.

$$p(\mathbf{x}_t, m_t|q_t) = c_{q_t m_t} \prod_{w=1}^{W} p_{q_t m_t w}^{x_{tw}} (1 - p_{q_t m_t w})^{1 - x_{tw}}.$$

The Q function then becomes

$$\begin{aligned} Q(\Theta, \Theta') = & \sum_{\mathbf{q} \in \mathcal{Q}} \sum_{\mathbf{m} \in \mathcal{M}} \sum_{t=1}^{T} \log(p(q_0)) P(\mathbf{X}, \mathbf{q}, \mathbf{m}; \Theta') \\ & + \sum_{\mathbf{q} \in \mathcal{Q}} \sum_{\mathbf{m} \in \mathcal{M}} \sum_{t=1}^{T} \log(p(q_t|q_{t-1})) P(\mathbf{X}, \mathbf{q}, \mathbf{m}; \Theta') \\ & + \sum_{\mathbf{q} \in \mathcal{Q}} \sum_{\mathbf{m} \in \mathcal{M}} \sum_{t=1}^{T} \log(p(\mathbf{x}_t, m_t|q_t)) P(\mathbf{X}, \mathbf{q}, \mathbf{m}; \Theta'). \end{aligned} \tag{3}$$

Setting the derivative of the Q function with respect to each parameter to zero, we get the following expressions.

$$\pi_i = p(q_0 = i) = \frac{P(\mathbf{X}, q_0 = i; \Theta')}{P(\mathbf{X}; \Theta')}, i = 1, \ldots, S, \tag{4}$$

[1] Notations followed are similar to the procedure described in Bilmes [1997].

where $\boldsymbol{\pi}$ is the vector of initial state probabilities.

$$a_{ij} = p(q_t = j|q_{t-1} = i) = \frac{\sum_{t=1}^{T} P(\mathbf{X}, q_{t-1} = i, q_t = j; \Theta')}{\sum_{t=1}^{T} P(\mathbf{X}, q_{t-1} = i; \Theta')}, i, j = 1, \dots, S, \tag{5}$$

where $A = (a_{ij})$ is the transition matrix of the HMM containing S^2 parameters.

$$c_{il} = \frac{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = l|\mathbf{X}; \Theta')}{\sum_{t=1}^{T} \sum_{l=1}^{M_i} P(q_t = i, m_{q_t t} = l|\mathbf{X}; \Theta')}, i = 1, \dots, S \text{ and } l = 1, \dots, M_i, \tag{6}$$

$$\mathbf{p}_{il} = \frac{\sum_{t=1}^{T} \mathbf{x}_t P(q_t = i, m_{q_t t} = l|\mathbf{X}; \Theta')}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = l|\mathbf{X}; \Theta')}, i = 1, \dots, S \text{ and } l = 1, \dots, M_i, \tag{7}$$

where $\{\mathbf{c}_{il}, \mathbf{p}_{il}\}_{l=1}^{M_i}$ are the component proportions and Bernoulli parameter vectors for the MMB for state i.

In each step of the EM algorithm, we first compute the complete data likelihood using the parameter estimates in the current step. Using this value, we compute the updated parameter estimates by maximizing the Q function. This continues till we reach a local optima. In the end, the most likely sequence of state values corresponding to an observed sequence can be obtained using the Viterbi algorithm [Forney Jr. 1973].

4.3. Model Initialization

Finding good initialization values for the model parameters (transition probabilities and emission distribution) is done as follows.

—*Transition probabilities*. The binary input trace is transformed into a sequence of PRRs (in $[0, 1]$) computed over a window size W. We define a continuous HMM with S states and a continuous emission distribution (beta, normal) and use the EM algorithm to estimate by maximum likelihood its parameters (which we then discard) and its transition probabilities, given the sequence of PRRs.

—*MMB parameters*. Use the Viterbi algorithm [Forney Jr. 1973] to obtain the most likely state sequence for each input trace and grouped it into the same cluster all windows assigned to the same state. Practically speaking, this tends to group windows associated with similar PRR values. For each long-term state, we trained an MMB model only on its corresponding cluster. The training vectors in each cluster are split in training (70%) and testing sets (30%). For each training-testing set combination, we try ten different initializations for the MMB parameters. We used an EM algorithm for MMB, as described in Carreira-Perpiñán and Renals [2000], resulting in the initial estimates for proportion and Bernoulli W-dimensional vector for each of the M mixture components. The parameters with the highest log-likelihood on the test set are selected as initialization for the MMB emission distribution of the HMM in the M&M model.

For models with hidden variables, such as HMMs or mixtures (of widespread use in machine learning and statistics), local optima are a known problem. In practice, one usually tries multiple restarts, and while one can't generally expect to find the global optimum, typically one finds a good enough local optimum. Using the aforementioned initialization methods, the EM algorithm converges to one of several local optima for the problem.

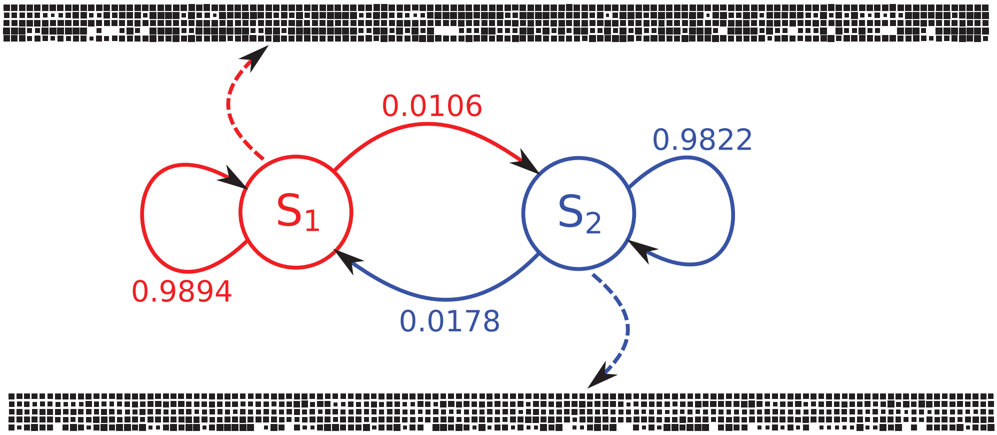


Fig. 6. Graphical Representation of the M&M Model. The transition probabilities are overlaid on the arrows showing the state transitions. The MMB emission distribution for each state is represented by a matrix of shaded squares, wherein the degree of shading indicates if the output is 1 (black square) or 0 (white square).

4.4. Graphical Representation of the M&M Model

Figure 6 shows a graphical representation of the M&M model for $S = 2$, $M = 5$, and $W = 128$. The transition probability matrix is shown using a transition diagram for the HMM. The MMB emission distribution of each state is represented using a format similar to Hinton diagrams. Each state output is represented by an $M \times W$ matrix of shaded squares, wherein the shaded area inside each square is proportional to the Bernoulli parameters for the MMB distribution. If $p = 1$, then the square appears black and if $p = 0$ then the square appears white. The diagram is meant to provide a visual representation for the 1/0 output of the M&M model. Longer runs of 1's would be outputted by a model if we see sequences of consecutive black square, and longer runs of 0's would be outputted for sequences of consecutive white squares.

5. EVALUATION OF THE M&M MODEL

To evaluate the performance of our approach, we trained models for links with different reception rates from the experimental data traces (training set, length = 230,400). As the problem is unsupervised (i.e., there is no ground truth to compare with) and the generated sequences can have any length, we do not compare the likelihood value that the models produce for a trace. Instead, we compare on the basis of statistics computed on the traces versus a different set of unseen data traces (testing set) having similar PRR characteristics. For each link, we proceeded as follows: (1) we learned the model parameters given the (training set) data traces and different combinations of model sizes, that is, S ($\in \{2, 4, 6\}$), M ($\in \{2, 4, 5, 10, 15\}$), and W ($\in \{8, 16, 24, 32, 48, 64, 96, 128, 160, 192\}$). (2) For each model, we sampled a sequence as long as computationally possible (to reduce the variability in our statistics). (3) From this sequence, we computed the following statistics and compared them with the same statistics computed for the testing set (different from the training set).

(1) PRR, to assess the long-term behavior of a link.
(2) Distributions of run lengths of 1's, $r_1(n)$, and 0's, $r_0(n)$, for $n = 1, 2, \ldots$ This assesses both the global and local behavior. The run-length (RL) distribution estimate is defined on a range independent of the data, namely, $[1, \infty)$. Different RL distributions can easily be compared (e.g., with the Lp distance) and have statistics defined on them (e.g., variance). Each new bit changes the RL distribution in a localized way:

it adds 1 to the appropriate run length. The information about long bursts is easily seen by looking at the tail of the RL distribution and can be enhanced by having each run of length L count as L, instead of 1. We term this *weighted run length (WRL)* distribution. It is similar to the RL distribution except that it enhances the longer runs. Later, we plot the WRL distributions to emphasize the occurrence of long runs of 1's and 0's.

(3) The conditional packet delivery function (CPDF) [Lee et al. 2007] $C(n)$, defined as the conditional probability of observing a 1 after n consecutive 1's or 0's. This assesses the global and local behavior. The CPDF estimate is defined only on a range $[0, R]$, where R is the length of the longest run, which depends on the data sequence. It is not defined beyond R, because no such run is observed. In fact, even in that range, $C(n)$ is highly sensitive to the sequence, particularly for the larger n. CPDFs are sensitive to the appearance of a single burst which adds an area of probability approximately equal to 1 around $n = L/2$, where L is the burst length. This happens no matter how long the trace is and no matter how often such bursts occur, as long as they occur at least once. Each new bit (1/0) in the sequence changes a possibly large part of the CPDF (up to the whole of it). Thus, CPDFs are good for detecting a burst of 1/0's but not suitable for determining the frequency of 1/0's. It is difficult to compare CPDFs from different datasets as the length of the largest burst will vary from sequence to sequence. While one can eliminate all d values having less than a minimum number of runs, this loses information by truncating the tail.

(4) Allan deviation[2] [Aguayo et al. 2004] (AD) is computed as the square root of one-half of the average differences between squares of successive samples over a given sampling period. The formula for AD of a sequence of samples x_i is

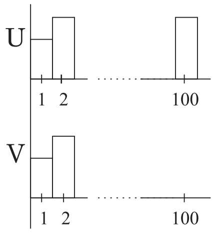
$$AD = \sqrt{\frac{1}{2(n-1)}\sum_{i=1}^{n-1}(x_{i+1} - x_i)^2}.$$

The AD plot summarizes the difference between successive samples of a quantity at different time scales. In our case, the samples are PRRs computed over different window sizes. According to Aguayo et al. [2004], the AD will be high for window sizes near the "characteristic burst length". In our study, we utilized the AD plots to determine the combination of S, M, and W that shows the most similarity between the simulated traces and the testing traces.

5.1. Comparing RL and CPDF Distributions

To compare differences in the distributions of the run lengths and CPDFs of the testing and simulated traces, we can compute the average L_1-norm between them. However, when computing the average L_1-norm, the difference in the two distributions is weighted equally for the common cases, that is, short runs/bursts of 1/0's, and for the rare cases, that is, very long runs of 1/0's. The absence of rare cases in the simulated traces does not significantly affect the L_1-norm between the two distributions, thereby potentially misrepresenting the performance of a modeling approach. The inability of a modeling approach to account for the rare cases is a serious shortcoming for simulation users, as they will not be able to adjust the behavior of algorithms/protocols for such cases which will eventually result in failure under real-world conditions. On the other hand, the L_1-norm would exaggerate the difference between traces from the same model when the length of the long runs/bursts varies slightly. To highlight the effect of the absence of rare cases and that of minor differences between rare cases

[2] http://www.allanstime.com/.



$$D(U, V) = |U(1) - V(1)| + |U(2) - V(2)| + |U(100) - V(2)| + |100 - 2|/1000$$

$$D(V, U) = |V(1) - U(1)| + |V(2) - U(2)|$$

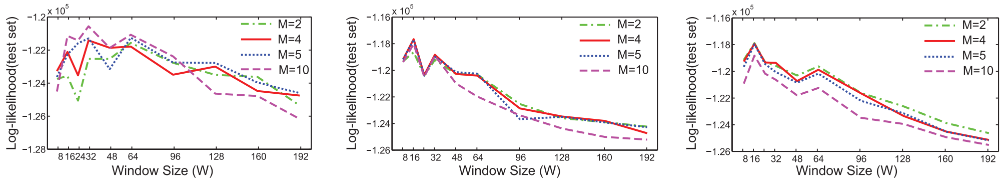
Fig. 7. Computing the distance between two distributions U and V. In this illustration, U is defined at 1, 2 and 100, and V is defined at 1 and 2 only.

from the same model, we designed a new metric called the Nearest Neighbor Distance. Although this is not the only way to emphasize the importance of rare events [Kullback and Leibler 1951; Rubner et al. 2000], it worked well in our case.

Nearest Neighbor Distance (NND). Let U and V be two functions, each defined on a (possibly different) subset of the natural numbers. In our case, U and V are the RL or CPDF distributions from the empirical and simulated traces, and we consider the RL distribution to be defined only where its value is positive. We define a nonsymmetric distance $D(U, V)$, as the sum over all the existing entries i of U of the following: $|U(i) - V(i)|$ if $V(i)$ is defined, and $|U(i) - V(j)| + \alpha|j - i|$ if $V(i)$ is not defined, where j is the closest entry to i for which $V(j)$ is defined. That is, $D(U, V)$ behaves like the L_1 distance, where both U and V are defined, and like a penalized L_1 distance to the closest entry, where V is defined, otherwise. We chose $\alpha = 1/1000$ empirically. We tried several values that could highlight differences in traces with regards to missing runs of 1's and 0's. With $\alpha = 1/1000$, we observed that we did not severely penalize cases where the difference in run lengths was not significant <50. On the other hand, if there was an absence of long runs of 1's or 0's, then our choice of α was sufficient to capture it. Figure 7 shows a sample calculation of $D(U, V)$ and $D(V, U)$. *NND* is then computed as $(D(U, V) + D(V, U))/2$, which is now symmetric.

5.2. Model Selection—Trade-off between *S*, *W*, and *M*

Before settling on a model size, we would like to analyze the trade-offs associated with it. Using many parameters (higher S, W, and M) yields a powerful model but is more prone to over-fitting and local optima. In addition, such models are computationally costly. On the other hand, using few parameters may not yield a powerful enough model. Generally, a model with fewer parameters is preferred because (i) a model should make as few assumptions as possible (Occam's razor) and (ii) the data required for training the model to a given degree of accuracy grows exponentially with the number of variables. From the simulation user's point of view, the model should be able to generalize to traces having similar long- and short-term dynamics, that is, the simulated traces should have small *NND* values and the AD plots of the simulated and testing traces should be qualitatively similar. Also, the simulated traces should converge to the model PRR quickly (i.e., shorter trace length). Let us suppose that for training a model, we are using a trace of length L with a certain S, W, and M. Here, the ratio $\frac{L}{SMW}$ determines approximately the number of observed sequences that can be attributed to each of the MMB components in each state of the HMM. Due to the high dimensionality of the problem, it is necessary for each MMB component to have adequate number of training sequences. Therefore, as a rule of thumb, $\frac{L}{SMW} \geq 100$ would ensure that condition to be satisfied, that is, each MMB component in each state of the HMM will have $\geq$100 training sequences (of window size W).



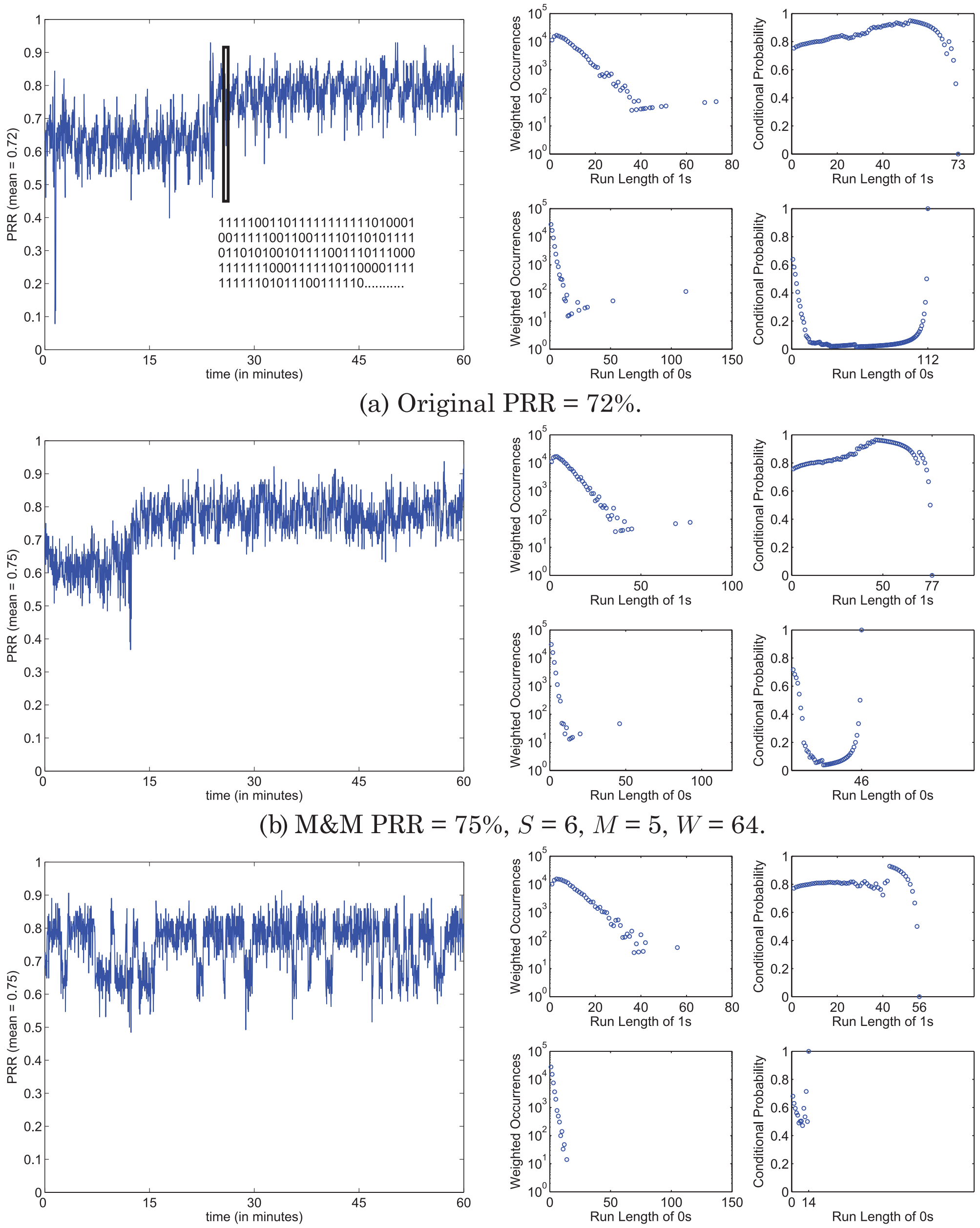
(a) $S = 2$. (b) $S = 4$. (c) $S = 6$.

Fig. 8. Variation in log-likelihood of the testing set as a function of S, M, and W using the M&M model for an empirical trace. In (a), (b), and (c), the different lines correspond to different values of $M (= 2, 4, 5, 10)$.

Having more parameters leads to higher log-likelihood on the training set but makes the model more prone to over-fitting problems. In machine learning, model selection is done using a validation set. Models with different number of parameters are trained, and their performance is measured on an independent dataset. The selected model is the one with the best predictive performance on this independent dataset. In the M&M model, we measure the performance in terms of log-likelihood on an independent dataset that we term the test dataset, which was not used for training the model. Therefore, we select a model based on its performance on some unseen data which was not used for training the model. When the EM algorithm for HMM with MMB emission distribution converges to a solution, the model having the highest log-likelihood on the testing trace is selected. In general, the log-likelihood on the test set increases as we increase the number of model parameters and then starts decreasing after a certain point. Figure 8 shows the variation in the log-likelihood on the testing trace as a function of S, W, and M. The figure indicates that model selection on just the basis of log-likelihood will lead to models with small $W (= 8, 16, 24, 32)$. Also, this demonstrates that our metric (log-likelihood on the test set) is not biased towards models with larger number of parameters. Regarding use of criteria, such as AIC, such criteria do not take account of the uncertainty in the model parameters, and in practice, they tend to favour overly simplistic models [Bishop 2006].

The role of W in the M&M model is to split the responsibilities between the transition probability matrix and the emission distribution. In principle, moving the modeling responsibility entirely to the transition matrix (by making $W = 1$ and having high S) or to the emission distribution (by making W very large) could work by having a very large number of parameters. In practice, it would not, because it would require a far larger training set, and the model would be plagued with local optima of bad quality. Essentially, the short- and long-term description is a divide-and-conquer strategy, and could be applied in general with a hierarchy of levels. Besides, long-term transitions can happen no faster than every W bits, which puts an upper limit (although very large in our traces) on W. Figure 9 plots the effect of variation in window size W on the quality of the packet loss model for a given training trace ($S = 6$). From Figure 9(c), we see that at very small values of $W = 8$, the transitions between the long-term dynamics of the link are not captured accurately in the transition matrix of the underlying HMM-based model. As window size increases, models with higher values of W ($W = 64$) show similar variation in long-term dynamics as the original link. Also, from Figure 9(c), we see that for small $W (= 8)$, the model is unable to account for the longer runs of 1's and 0's, as seen in the original link. In contrast, the model for $W = 64$ has longest run of 77 1's (original link has 73 1's) and 46 0's (original link has 120 0's).

Table III ranks models based on the NND computed between the simulated and the testing trace. A lower NND is an indicator of how well the M&M model is able to simulate bursts of 1/0s. We observe that models with longer W ($>= 64$) perform better in terms of modeling long-term dynamics and run-length distributions of 1's



(a) Original PRR = 72%.

(b) M&M PRR = 75%, $S = 6$, $M = 5$, $W = 64$.

(c) M&M PRR = 75%, $S = 6$, $M = 20$, $W = 8$.

Fig. 9. Average PRR over time from (a) experimental one-hour data trace, (b) M&M simulated trace ($W = 128$), and (c) M&M simulated trace ($W = 8$), respectively. On the right side, we see the statistics for each link for the weighted run lengths (WRLs) and CPDFs. Note: (i) WRL values do exist for each integer and are zero if not shown; (ii) CPDF values do not exist beyond a maximum run length, so CPDF plots are truncated at maximum run length.

Table III. *NND* and Log-Likelihood (LL) Values for the Testing Trace Using the M&M Model for Trace in Figure 9

			NND		
S	M	W	RL	CPDF	LL
4	4	128	0.861	19.63	−123,589
6	10	24	0.864	23.12	−120,252
6	5	24	0.887	23.60	−119,537
6	4	24	0.906	24.51	−119,417
6	4	48	0.939	24.96	−120,806
6	2	24	0.939	24.32	−119,398
4	5	128	0.943	19.92	−123,610
6	5	48	0.943	23.93	−120,929
6	4	128	0.989	20.52	−123,422
6	**5**	**64**	**0.993**	**22.17**	**−120,129**

Note: The values in bold indicates the model with balanced performance in terms of *NND* and LL. The first model performs best in terms of *NND* but worse in terms of LL.

and 0’s, compared to ones with smaller W (< 64) (from Figure 9). Thus, we need to select a model that strikes a balance between log-likelihood, *NND* values, and long-term dynamics performance. The values in bold indicate the model size (S, M, W) that provides the best balance in terms of the log-likelihood, *NND* computed with respect to the testing trace, and long-term dynamics. We also plot the Allan deviation (see Figure 10) of three links: (i) a trace which was generated assuming independent packet loss pattern, (ii) the simulated trace, and (iii) the testing trace. The point in the plot where the Allan deviation of the simulated and testing traces deviates from that of the independent trace indicates the characteristic burst length of the simulated and testing trace. From Figure 10, the Allan deviation plots of the testing trace indicates that packet reception is independent for time intervals less than one second but show bursty packet reception for intervals greater than one second.

In the measurement study of link burstiness, Srinivasan et al. [2008] have observed that waiting for 500 ms breaks the packet-loss correlation. While studying the self-similarity property of links, Rusak and Levis [2009] have observed that links start displaying self-similarity (correlations) after 640 ms. In the design of STLE, Becher et al. [2008] and Alizai et al. [2009] set the threshold for identifying an intermediate link to three packets when the IPI was set to 250 ms. This design choice has the underlying assumption that intermediate links show stable short-term behavior over a period of >750 ms. During the evaluation of Roofnet, Aguayo et al. [2004] observed that bursty links show correlation out to at least one second. These empirical studies support the case for longer window sizes. Having longer window sizes for the M&M models would help capture the correlations that exist in real-world wireless links. While this would negatively affect model performance in terms of log-likelihood (see Figure 8 and Table III), from a simulation users point of view, M&M models with longer window sizes would capture the correlations that exist in real-world wireless links. Also, we saw in Figure 9(b) that M&M model with $W = 64$ was able to capture long- and short-term behavior better than models with smaller W ($= 8$). In addition, the one second interval from the Allan deviation plot (Figure 10) coincides with window size $W = 64$.

These results show that for the M&M model to be of practical use, we need longer window sizes. Hence, we settle on $W = 64$ as the choice for window size. For the number

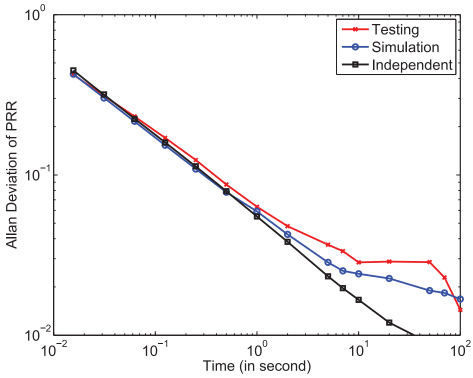


Fig. 10. Variation in Allan deviation of PRR for (i) a trace with independent packet-loss pattern, (ii) the simulated trace with $S = 6$, $M = 5$, and $W = 64$, and (iii) the testing trace.

of states, we select $S = 6$ and $M = 5$ for mixture components for each state. Our M&M models with the chosen size do not simultaneously perform best in terms of NND and LL but balance the trade-off associated with it. Note that for this particular model size, we used traces that contain at a minimum of 230,400 packets with an IPI = 16 ms. There is no formula for selecting the model size, as links may display a different burst pattern and dynamics that might require a different choice of model size. As a reference for simulation users, we offer the following guidelines for choosing a desired model size.

(1) For a given training wireless link, choose W by plotting the Allan deviation plot. Select a W such that the packets inside the W-length window cover a time interval greater than the burst length.
(2) Select S greater than or equal to the number of long term states, one observes in the wireless trace.
(3) For M, choose a value of five or less. Having more than five components increases drastically the number of parameters in the model and may lead to overfitting.
(4) In addition to points (1–3), make sure that the $\frac{L}{SMW} \geq 100$ rule is satisfied, as it ensures that each Bernoulli prototype gets enough training data (i.e., ≥ 100). As a caveat, note that the rule assumes a uniform distribution of training data for each prototype which might not be true always.

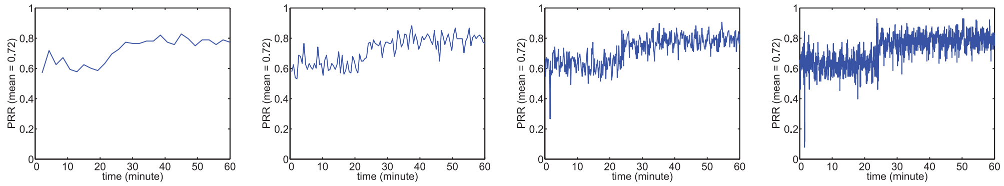
5.3. Model Convergence

Another point of interest is the rate of convergence of traces generated from the model parameters, or simply put, the variance between the original and generated traces. Ideally, we want the average packet reception rate of the simulated traces to be equal to that of the original trace. However, in reality, the simulated traces show some variance due to the stochastic nature of the model. The rate of convergence of the generated traces can be inferred from the model parameters, making certain models more suitable than others.

Using the Perron-Frobenius theorem [Meyer 2001], for the transition matrix of an irreducible finite Markov chain, the eigenvector corresponding to the leading eigenvalue is the unique stationary distribution for that Markov chain. Hence, in the M&M model for a packet reception trace, using singular value decomposition, we can calculate the leading eigenvalue and the corresponding eigenvector that gives the stationary distribution of the HMM. The ratio, λ, between the first two eigenvalues gives the rate of convergence of traces generated from the model. The closer λ is to 1, the slower is the

Table IV. Variation in λ as We Vary S and W for an Empirical Trace

S	W=32	W=64	W=96	W=128	W=160	W=192
2	1.0101	1.0105	1.0078	1.0064	1.0077	1.0059
4	1.0051	1.0016	1.0086	1.0118	***1.0123***	1.0095
6	1.0048	1.0115	1.0102	1.0122	1.0061	1.0037



(a) *Freq.* = 1 Hz. (b) *Freq.* = 4 Hz. (c) *Freq.* = 16 Hz. (d) *Freq.* = 64 Hz.

Fig. 11. Dependence on frequency of sending packets during data collection.

convergence. Table IV shows the values of λ as a function of S and W. The numbers in bold italics indicate the best value of λ (faster convergence) ($S = 4$, $W = 160$).

In addition, given the stationary distribution, we can compute the average PRR of the HMM–MMB model as follows.

$$PRR = \sum_{q=1}^{S} \nu_q \mu_q,$$

where, ν_q is the value in the eigenvector giving the stationary distribution of the HMM corresponding to the qth state and μ_q is the average PRR. From the MMB emission distribution,

$$\mu_q = \sum_{m=1}^{M} c_{qm} \frac{\sum_{w=1}^{W} p_{qmw}}{W},$$

where, c_{qm} is the mixture proportion for the mth mixture component in the qth state and p_{qmw} is Bernoulli prototype in the wth position.

5.4. Sensitivity Analysis: Dependence on IPI during Data Collection

During data collection for building the M&M model, we sent fixed-size packets at a frequency of 64 Hz or 64 packets per second (pps) in our experiments. In contrast, earlier studies [Rusak and Levis 2008] have collected the same data at a lower frequency (4 Hz). To analyze the dependence of frequency of sending packets during the data collection phase on the quality of our model, we reduced the amount of data used for creating the model from the original 64 Hz down to 1 Hz. Figure 11 shows the variation in reception rates for the same link modeled using different amounts of data. From Figures 11(a), 11(b), 11(c), and 11(d), we see that as the frequency increases, the greater amount of data used for creating the model helps the simulated trace follow the behavior of the original trace (see Figure 9(a)) very accurately at long and short time scales. As data collection frequency decreases, the simulation traces from the model get smoothed out, resulting in loss of detail in the short-term correlations. Hence, we advocate collecting data at high frequency when possible.

5.5. M&M Simulation Results

Table V shows comparisons between the test traces and the simulated traces from the M&M model ($S = 6$, $M = 5$, and $W = 64$). The average difference between the PRR of the simulated and the test link PRR is less than 1.9%, whereas the average

Table V. Comparison between Empirical Traces (Testing Set) and Simulation Traces Using the M&M Model and TOSSIM

Test	M&M					TOSSIM				
Trace	PRR	Avg. L_1-norm		NND			Avg. L_1-norm		NND	
PRR	Mean±StdDev	CPDFs	RL	CPDFs	RL	PRR	CPDFs	RL	CPDFs	RL
0.469	0.475±0.020	0.12	0.007	40.1	2.1	0.417	0.698	0.029	40.6	2.8
0.520	0.512±0.002	0.44	0.052	10.2	2.2	0.002	0.990	1.026	4,238	207
0.614	0.618±0.031	0.31	0.004	3.1	0.4	0.115	0.680	0.201	12.5	1.9
0.621	0.632±0.010	0.30	0.004	6.8	0.6	0.146	0.746	0.193	12.5	1.9
0.675	0.741±0.008	0.39	0.041	10.5	1.9	0.001	0.902	0.523	6,965	181
0.706	0.740±0.020	0.24	0.008	201	2.1	0.225	0.859	0.180	261	5.2
0.723	0.748±0.024	0.37	0.057	101	3.2	0.116	0.854	0.204	111	4.1
0.728	0.750±0.029	0.21	0.021	27.3	1.7	0.270	0.844	0.135	159	5.2
0.886	0.890±0.017	0.11	0.030	4.7	1.9	0.001	0.979	1.001	80	227
0.906	0.912±0.011	0.26	0.049	30.7	1.6	0.065	0.941	0.210	40.6	8.3

standard deviation in the PRR of the simulated M&M links is 0.017. The worst-case difference in PRR is 6.6%. Table V also shows the difference between the run-length and CPDF distributions in terms of the average L_1-norm and the NND. The minimum difference between the CPDFs in terms of the average L_1-norm and the NND is 0.11 and 3.1, respectively, and the maximum difference is 0.44 and 201, respectively. For the run lengths, the minimum difference in terms of the average L_1-norm and the NND is 0.004 and 0.4, and the maximum difference is 0.44 and 3.2, respectively. The maximum difference between the distributions occurred when the M&M model is not able to simulate the longer runs/bursts as seen in the testing trace and is captured by the NND computations.

When sampling state sequences from the M&M model, there is a possibility that the overall proportion of time the link stays is a particular state is different from the testing trace. Our goal was to be able to simulate traces which matched performance of the testing trace with similar long- and short-term dynamics as observed in the training data. While simulating traces, we sampled state sequences from the model 20 times to observe the long-term behavior. For computing other statistics, we sampled traces 100 times. However, we restricted our choice to traces that had a PRR within 5% of the training trace PRR. Therefore, in Table V, the standard deviation in PRR of simulated traces in all cases is less than 5%. While constraining the variability of the traces, it allows for a comparison to the testing trace on more even terms (i.e., state sequences (long-term behavior) are similar but not same).

5.6. M&M Simulator

In order to make the M&M model accessible to WSN simulation users, we have incorporated it in the TOSSIM simulator for TinyOS 2.0. The M&M simulator provides the end user the capability to simulate a network with links having different PRRs. Using the approaches described in Kamthe et al. [2009], we have created a library of M&M models with intermediate PRRs ranging from 0% to 100%. The simulator generates PRR traces using these pre-computed models and utilizes the values (1/0) in the trace to make a decision regarding the link quality. In addition, the simulator can re-execute PRR traces generated in prior experiments or user supplied traces to allow for deeper analysis of link quality on program execution. The files required for the M&M simulator are available at Kamthe [2009].

6. PERFORMANCE COMPARISONS WITH TOSSIM COMMUNICATION MODEL

We conducted a statistical comparison between empirical data traces (testing set), simulation traces from the M&M model, and traces from TOSSIM, the TinyOS simulation environment. TOSSIM requires a link-gain model wherein a unidirectional link between a source and destination is associated with a gain value, that is, the received signal strength between the source and destination. For simulating traces in TOSSIM, for each of the empirical traces (testing set), we computed the median RSSI value of the received packets in the traces. We used this as the gain for the link gain model for the TOSSIM links.

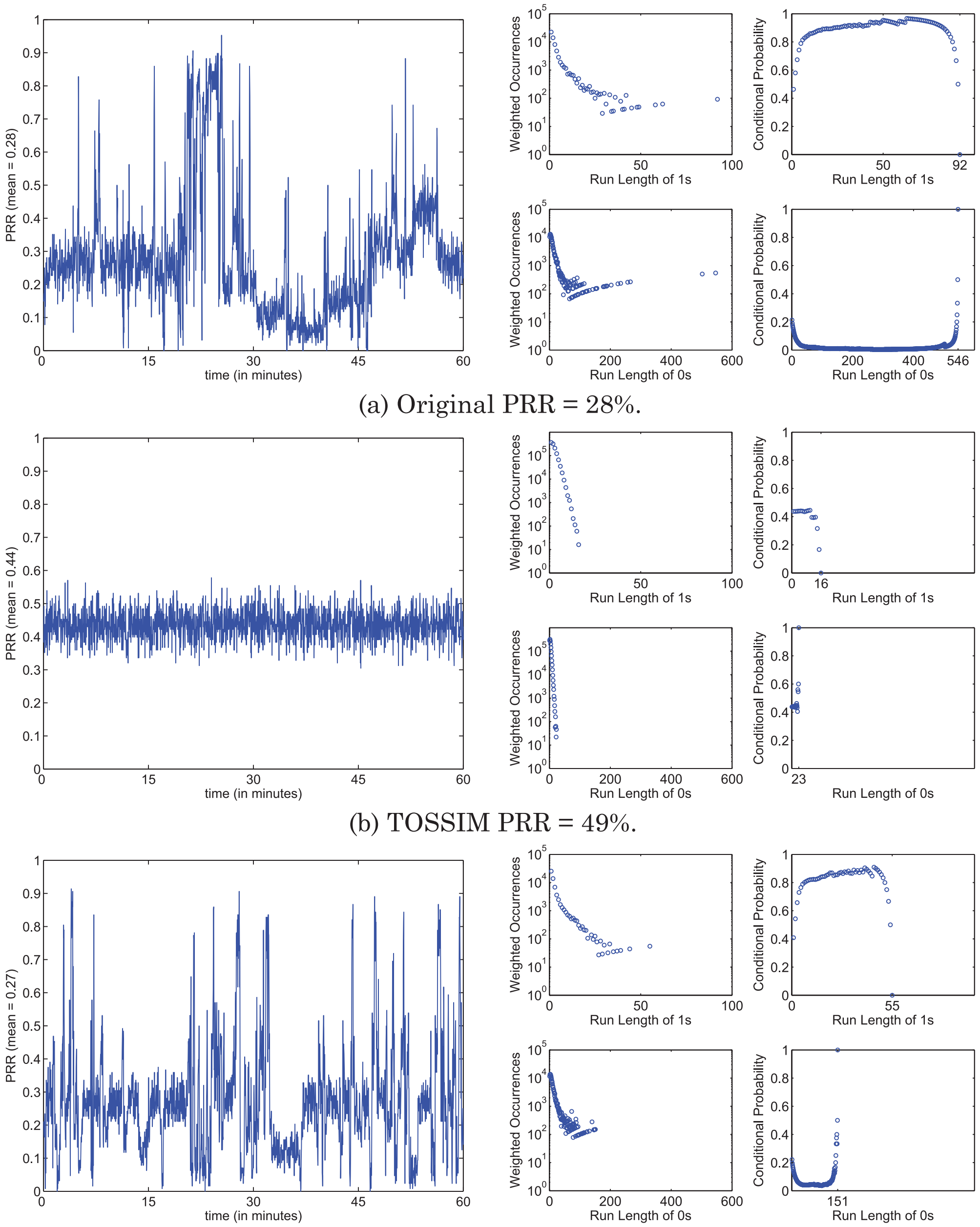
TOSSIM utilizes a communication model called Closest Pattern Matching (CPM) [Lee et al. 2007]. In order to utilize CPM, users must first collect a high-frequency noise trace from a deployed WSN that will be used to bootstrap the noise model. As mentioned in Section 3.1, we used the *RssiSample* application to collect these traces from the same environment where we collected our packet reception traces. To compare the performance of TOSSIM with the proposed M&M model, we bootstrapped the TOSSIM noise model using traces collected from the SE and MoteLab testbeds. The Signal-to-Noise Ratio (SNR) is computed using noise values generated by the CPM model. Using this SNR value, the corresponding PRR value is determined using a SNR-PRR curve (implemented in TOSSIM for TinyOS 2.x)[3] [Zuniga and Krishnamachari 2004]. The packet reception status (success/fail) for a packet is decided by sampling once from a Bernoulli distribution with $p = PRR$. In essence, the CPM model captures short-term correlations as the noise dynamics are modeled using data collected over interval of ≈3–4 minutes (1ms sampling interval). Note that during data collection, the noise samples were collected immediately after or very close to the time we collected the trace data.

Figure 12 shows the variation in PRR of a particular link and the simulated traces generated using TOSSIM and the M&M model trained on the same link. The goal of Figure 12 is to qualitatively contrast the link-quality variation in simulation traces from TOSSIM and the M&M models with respect to an original link manifesting complex link dynamics. It is clear from Figure 12(b) that TOSSIM is unable to capture the long-term variations in PRR that are better modeled by the M&M model (see Figure 12(c)). Furthermore, the average PRR of the M&M link (27%) is closer to the original link PRR (28.47%) than the TOSSIM link PRR (49.49%).

Figures 12(a), 12(b), and 12(c) show the weighted RL and CPDF distribution of 1's and 0's for the original link, TOSSIM simulation trace, and M&M simulation trace, respectively. We observe that the M&M model has longest run of 55 1's (original link has 92 1's) and 151 0's (original link has 546 0's). On the other hand, TOSSIM generated trace with 16 1's and 23 0's. It is clear that TOSSIM is not able to simulate the longer runs of 1's and 0's. This is also reflected in the NND computed for the TOSSIM and M&M traces. The NND for the RL distribution of the TOSSIM and M&M traces is 4.07 and 1.8, respectively. The NND for the CPDF distribution of the TOSSIM and M&M traces is 82.2 and 40.16, respectively. These values indicate that, quantitatively, the M&M traces are closer to the original traces than the TOSSIM traces.

Table V shows the summary of the comparison between the empirical traces (testing set) and traces generated using TOSSIM and the M&M model. The first point to notice is that there are significant differences in PRR between the actual link and TOSSIM model with a minimum difference of 5% and a maximum of 88%). In contrast, the M&M model has a maximum and minimum difference in PRR of 6.6% and 0.4%, respectively. The maximum NND for the run-length distribution of the TOSSIM and M&M traces is 226.9 and 3.2, respectively. The maximum NND for the CPDF distribution of the TOSSIM and M&M traces is 6,965 and 201, respectively. These values indicate that, quantitatively, the M&M traces are closer to the (unseen) testing traces than the

[3] `http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x/tos.lib/tossim/CpmModelC.nc?view=log`.



(a) Original PRR = 28%.

(b) TOSSIM PRR = 49%.

(c) M&M PRR = 27%, $S = 6$, $M = 5$, $W = 64$.

Fig. 12. Average PRR over time from (a) experimental one-hour data trace, (b) TOSSIM simulated trace, and (c) simulated trace using the M&M model, respectively. On the right side we see the statistics for each link for the weighted run lengths and CPDFs of packet reception (top) and losses (bottom).

TOSSIM traces. The combined knowledge of the difference in PRRs and the average L_1-norm and NND values for the distributions of run lengths and CPDFs indicate that TOSSIM does not do an adequate job of modeling the link variations.

We believe the poor performance of TOSSIM can be explained by the inadequate characterization by the path-loss model and the noise model being able to account only for short-term correlations. Currently, TOSSIM uses the gain of the link and the noise value computed by CPM to decide whether the packet is received or dropped. However, the generic constants of the path-loss model are not the same for all environments. This may lead TOSSIM to make significant errors while computing the PRR of a packet at the receiver. In addition, it has been shown that the RSSI values recorded by nodes have errors due to calibration issues [Chen and Terzis 2010]. These may introduce inaccuracies in computing the SNR which further propagate to PRR calculations. While it is possible to eliminate this error by calibrating all receiver nodes, it introduces an additional nontrivial overhead. Raman et al. [2009] have shown that intermediate links occur due to the unpredictable behavior at the RSSI threshold. This indicates that any model, such as the path-loss model, which uses SNR values that are dependent upon RSSI would inadequately characterize the link behavior. This would not affect packet-loss models such as ours that only use knowledge of packet receptions and losses to model the link behavior. In Table V, there are cases where the TOSSIM traces differs significantly in PRR from the test trace. Also, in such cases, the values of NND are extremely large. This can be attributed to the unpredictable link behavior at the RSSI threshold, resulting in the SNR–PRR curve in TOSSIM outputting a PRR that is very different from the one observed in the wireless trace. In other cases, lower NND for M&M model traces indicates better performance in terms of run-length distributions. Here, our choice of α does not exaggerate the slightly worse performance of the simulation traces when there minor differences in the run-length distributions.

7. DISCUSSION

7.1. Relevance to Other Analytical Models

The Gilbert-Elliott model [Gilbert 1960; Elliot 1965] is a particular case of the M&M model, where we have a single-bit window ($W = 1$) and $S = 2$ states, and each state has a single-component MMB ($M = 1$). Its only tunable parameters are the transition probabilities and the Bernoulli parameters (total four parameters). The generality of our model allows us to model and learn from data, not just bursts, but far more complex behaviors. The Markov-Based Trace Analysis (MTA) [Konrad et al. 2001] is an extension of the Gilbert-Elliott model wherein one state corresponds to the error-free state of the channel and the other state is comprised of a discrete time Markov chain of order six to model the lossy state of the channel. This was further extended to account for variability in wireless links by using a hierarchical model with multiple states [Konrad et al. 2006], where each state is comprised of a two-state MTA-based model. Salamatian and Vaton [2001] used hidden Markov models with Bernoulli emission distributions for modeling packet reception traces from Internet communication channels. Their model is a particular case of the M&M model with single-bit window ($W = 1$) and $S \leq 4$ states, and each state has a single-component MMB ($M = 1$).

7.2. Model Adaptation

As our model is trained using packet reception data, this methodology presents several caveats for users of our simulation model. (1) Although, the model is highly accurate for data collected from a given environment, a simulation user would be limited to simulating their network based on conditions during the data collected at the SE and MoteLab test beds. (2) If the user wants to simulate network conditions in a particular environment, (s)he should collect at least some packet reception traces in the target environment. In many settings, the benefits from a pure data-driven approach are not that large, because the generalizability of simulating from traces is a big limitation. For example, one would like to model the characteristics of a real environment in a simulated network without having to first deploy a network to measure its properties or by collecting significantly smaller data traces than the one used to train the model in a different environment. This problem can be solved by using model adaptation techniques similar to those proposed in Kamthe et al. [2011], wherein an algorithm was proposed that can adapt a preexisting MMB trained with extensive data to a new wireless link from which very limited data is available. The adaptation approach constrains the new MMB's parameters through a nonlinear transformation of the existing MMB's parameters. The transformation has a small number of parameters that are estimated using a generalized EM algorithm. The authors showed that their approach can learn MMB models from data traces of about one minute, which demonstrate comparable loglikelihood to MMB models using ten times more data. However, the caveat of such methods is that they are unable to account for the long-term characteristics, which is captured by the transition matrix of the M&M model, when the entire trace (approx. one hour) is available for training the model parameters.

7.3. User Control

The M&M model is a purely data-driven approach. It is possible to combine this with a non-adaptive approach so that the user may have manual control on specific characteristics (such as the amount of overall burstiness or fading rate) while still generating realistic traces. Future work could propose hybrid models containing a large number of parameters that are automatically tunable on a training set (e.g., the Bernoulli parameters), and a small number of "control" parameters that are set by the user. For example, consider an MMB having $W = 6$ and $M = 2$. c_i's indicate mixture proportions and p_i's indicate Bernoulli parameters for the mixture components. In this mixture, if the user wants the model to output increased runs of 1's of length 3 and runs of 0's of length 2, then the goal could be easily achieved by changing the mixture parameters as shown.

$$\begin{array}{cccccccc} & & \text{Before} & & & \text{After} & & \\ c_i & : & (p_1, \ldots, p_6) & & c'_i & : & (p'_1, \ldots, p'_6) \\ .6 & : & (.4, .7, .6, .7, .8, .5) & \rightarrow & .6 & : & (.4, .2, .9, .9, .9, .2) \\ .4 & : & (.4, .3, .3, .2, .2, .6) & \rightarrow & .4 & : & (.4, .3, .9, .1, .1, .9) \end{array}$$

Similarly, it is possible using a simple heuristic to find Bernoulli parameter values above/below a certain threshold (0.6 and 0.2 in the example) equal to the length of the required bursts and adjust them and their neighboring parameter values to ensure bursts of required lengths.

7.4. Model Limitations

A low-power wireless link with, say, two long-term states having the same average PRR but completely different short-term dynamics—independent behavior in one state and long bursts of 1/0's in the other—could be modeled adequately using the M&M model with proper initialization. However, modeling the transition matrix could be an issue if the transitions between the long-term states occurs just once in the entire trace. This can be attributed to the lack of data, and longer traces which could capture more transitions between long-term states could help alleviate the problem. In addition, the M&M model does not perform well when modeling dynamics, wherein the PRR changes monotonically with time, as the output distribution of the M&M model will converge to the average PRR, resulting in an inadequate model of the short-term dynamics.

Overfitting may be avoided during the modeling process by ensuring that we have long data traces (at least 230,400 packets) for parameter estimation. This combined with our $\frac{L}{SMW} \geq 100$ rule which assumes a uniform distribution of training vectors will greatly reduce the chances of overfitting.

As stated in Section 5.5, the maximum difference between the run-length distribution occurs when the M&M model is not able to simulate the longer runs/bursts, as seen in the testing trace, and is captured by the NND computations. This is related to the training data issue. Extremely long runs of 1's/0's are split into training vectors of length W. Since these runs are rare in the case of intermediate links, then such training sequences get combined with other sequences resulting in the prototype values in the MMB component to not be close to 1. This in turn reduces the probability of replicating extremely long runs of 1's/0's, culminating in higher NND values for the simulation traces.

7.4.1. Extension to Longer Time Scales. Rusak and Levis [2009] observed that link variation is present at different time scales (seconds, minutes, hours, and days). It is attributed to the presence of structure at different time scales, which they term *scaling*. When training the M&M model, the duration of our longest data traces was 12 hours. The level of long-term dynamics for traces up to 12 hours can be accounted for using the M&M model. For longer duration traces, of the order of days, months, years, the level of long-term dynamics might be greater than that of the 12 hour traces. For modeling such traces, the M&M model may not suffice. However, the modeling approach from Kamthe et al. [2009] can be used for longer time scales using an N-level hierarchy. For these longer duration traces, the strategy would be to model the dynamics of the longest time scales (e.g., years) at level 1 of the multilevel model. All subsequent level of dynamics, for example, month, weeks, days, hours, would have to be accounted at levels 2, 3, 4, and 5, respectively. At level 5, there would be an M&M model emitting binary sequences, which is ultimately the object of interest to the simulation user.

8. CONCLUSIONS AND FUTURE WORK

We presented a new multilevel Markov model (M&M) to replicate more realistic short- and long-term dynamics in wireless simulations. Our M&M model generalizes many existing wireless link models, can model complex correlations if sufficient parameters are used, and is straightforward to learn from data and to sample from. New M&M models could be created by mixing preexisting M&M models from a library. Based on extensive evaluation using long experimental data traces collected in multiple test-bed environments, we showed that the model significantly outperforms other simulation tools available in the WSN community.

There are multiple areas for future work. Regarding modeling, one could use the emission distribution restricted Boltzmann machines, which are another powerful way of representing high-dimensional binary data. Transforming existing model parameters to simulate new environments using orders of magnitude less training samples by applying model adaptation techniques is part of our research agenda. Moreover, the model could be extended to emit signal strength values, thus, modeling physical layer characteristics, such as RSSI values of wireless traces.

APPENDIXES

A. HIDDEN MARKOV MODELS

A good description of HMMs and MMBs can be found in Bishop [2006] and Carreira-Perpiñán and Renals [2000]. AWHMM models an observed sequence of (continuous or discrete) vectors in terms of a sequence $q_0,\ q_1, \ldots$ of hidden (unobserved) random variables called *states* and a sequence $\mathbf{x}_0,\ \mathbf{x}_1, \ldots$ of observed random variables (see Figure 5). The HMM represents the probability of the observed sequence in terms of the *state transition probability* $p(q = j|q = i)$ (which assumes the Markov property and is independent of time) between every pair of state values, and the *emission probability* $p(\mathbf{x}|q = i)$ of outputting a vector $\mathbf{x}$ when in state i. The latter can be, for example, a Gaussian or beta (or mixture thereof) for continuous $\mathbf{x}$ and a Bernoulli, multinomial (or mixture thereof) or a simple probability table for discrete $\mathbf{x}$. Thus, the probability of observing $\mathbf{x}_0,\ \mathbf{x}_1, \ldots, \mathbf{x}_T$ is

$$p(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T) = \sum_{q_0,\ldots,q_T} p(q_0) \prod_{t=1}^{T} p(\mathbf{x}_t|q_t) p(q_t|q_{t-1}),$$

where the sum is over all possible state sequences. A HMM is then described by the dimension W of the observed vector $\mathbf{x}$, the number of state values S, the $S \times S$ matrix of transition probabilities $a_{ij} = p(q = j|q = i)$, and the parameters of the emission distribution for each state value.

For simple emission distributions, the HMM parameters (transition probabilities and emission parameters) can be estimated given only a sequence of observed vectors $\{\mathbf{x}_t\}$ by maximum likelihood using an expectation maximization (EM) algorithm [Baum et al. 1970], which iterates from initial parameter values. This is the training or learning problem, and it is possible to converge to a local optimum. The most likely sequence of state values corresponding to an observed sequence can be obtained using the Viterbi algorithm. This is the decoding problem. Sampling from a trained HMM given an initial state value simply requires sampling states from the transition probabilities and sampling an $\mathbf{x}$ for each state from its emission distribution.

B. MIXTURES OF MULTIVARIATE BERNOULLI DISTRIBUTIONS

A Bernoulli distribution for a binary random variable x assigns probability p to $x = 1$ and $1-p$ to $x = 0$. A Bernoulli distribution in W binary variables is the product of W independent univariate Bernoulli distributions with parameter vector $\mathbf{p} = (p_1, \ldots, p_W)^T$.

$$p(\mathbf{x}) = \prod_{i=1}^{W} p_i^{x_i} (1 - p_i)^{1-x_i}.$$

A mixture distribution is constructed given M component distributions $p_1(\mathbf{x}), \ldots, p_M(\mathbf{x})$ and M component proportions $c_1, \ldots, c_M$ (with each $c_m \in (0, 1)$ and $\sum_{m=1}^{M} c_m = 1$).

$$p(\mathbf{x}) = \sum_{m=1}^{M} c_m p_m(\mathbf{x}),$$

and, if $M > 1$, then the components of $\mathbf{x}$ are not, in general, independent from each other; in fact, we can model complex correlations this way. The parameters $\{c_m, \mathbf{p}_m\}_{m=1}^{M}$ of a mixture of multivariate Bernoulli distributions (MMB) can be estimated given a collection of N W-dimensional binary vectors using an EM algorithm [Carreira-Perpiñán and Renals 2000], which iterates from initial parameter values and can converge to a local optimum. Sampling from a MMB simply requires picking a component with probability proportional to its proportion, and then sampling the binary vector from its Bernoulli.

REFERENCES

AGUAYO, D., BICKET, J., BISWAS, S., JUDD, G., AND MORRIS, R. 2004. Link-level measurements from an 802.11b mesh network. *SIGCOMM Comput. Commun. Rev. 34,* 4, 121–132.

ALIZAI, M. H., LANDSIEDEL, O., ÁGILA BITSCH LINK, J., GÖTZ, S., AND WEHRLE, K. 2009. Bursty traffic over bursty links. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*.

BAUM, L. E., PETRIE, T., SOULES, G., AND WEISS, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Stat. 41,* 1, 164–171.

BECHER, A., LANDSIEDEL, O., KUNZ, G., AND WEHRLE, K. 2008. Towards short-term wireless link quality estimation. In *Proceedings of the 5th ACM Workshop on Embedded Networked Sensors (HotEmNetS'08)*.

BILMES, J. 1997. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Tech. rep. TR-97-021, International Computer Science Institute.

BISHOP, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin.

CARREIRA-PERPIÑÁN, M. Á. AND RENALS, S. 2000. Practical identifiability of finite mixtures of multivariate Bernoulli distributions. *Neural Comput. 12,* 1, 141–152.

CERPA, A., BUSEK, N., AND ESTRIN, D. 2003. SCALE: A tool for simple connectivity assessment in lossy environments. Tech. rep. 0021, University of California, Los Angeles.

CERPA, A., WONG, J., KUANG, L., POTKONJAK, M., AND ESTRIN, D. 2005a. Statistical model of lossy links in wireless sensor networks. In *Proceedings of IEEE IPSN'05*. 81–88.

CERPA, A., WONG, J., POTKONJAK, M., AND ESTRIN, D. 2005. Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. In *Proceedings of ACM MobiHoc'05*. 414–425.

CHEN, Y. AND TERZIS, A. 2010. On the mechanisms and effects of calibrating RSSI measurements for 802.15.4 radios. In *Wireless Sensor Networks*. Lecture Notes in Computer Science, vol. 5970, 256–271.

ELLIOT, E. O. 1965. A model of the switched telephone network for data communications. *Bell Syst. Tech. J. 4,* 1, 89–109.

FORNEY JR., G. D. 1973. The viterbi algorithm. *Proc. IEEE 61,* 3, 268–278.

GILBERT, E. N. 1960. Capacity of a burst-noise channel. *Bell Syst. Tech. J. 39*, 1253–1266.

GIROD, L., ELSON, J., CERPA, A., STATHOPOULOS, T., RAMANATHAN, N., AND ESTRIN, D. 2004. Emstar: A software environment for developing and deploying wireless sensor networks. In *Proceedings of the Annual USENIX Technical Conference (USENIX'04)*. 283–296.

KAMTHE, A. 2009. M&M simulator for TOSSIM. `http://www.andes.ucmerced.edu/software`.

KAMTHE, A., CARREIRA-PERPIÑÁN, M. Á., AND CERPA, A. 2011. Adaptation of a mixture of multivariate bernoulli distributions. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*.

KAMTHE, A., CARREIRA-PERPINÁN, M. A., AND CERPA, A. E. 2009. M&M: Multi-level Markov model for wireless link simulations. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*. ACM, New York, NY, 57–70.

KASHYAP, A., GANGULY, S., AND DAS, S. R. 2008. Measurement-based approaches for accurate simulation of 802.11-based wireless networks. In *Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'08)*. ACM, New York, NY, 54–59.

KHAYAM, S. A. AND RADHA, H. 2003. Markov-based modeling of wireless local area networks. In *Proceedings of the 6th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM'03)*. 100–107.

KONRAD, A., ZHAO, B., JOSEPH, A., AND LUDWIG, R. 2001. A Markov-based channel model algorithm for wireless networks. In *Proceedings of the 4th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM'01)*.

KONRAD, A., ZHAO, B. Y., AND JOSEPH, A. D. 2006. Determining model accuracy of network traces. *J. Comput. Syst. Sci. 72,* 7, 1156–1171.

KOTZ, D., NEWPORT, C., GRAY, R. S., LIU, J., YUAN, Y., AND ELLIOTT, C. 2004. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*. 78–82.

KULLBACK, S. AND LEIBLER, R. A. 1951. On information and sufficiency. *Ann. Math. Stat. 22,* 1, 79–86.

LEE, H., CERPA, A., AND LEVIS, P. 2007. Improving wireless simulation through noise modeling. In *Proceedings of ACM IPSN'07*. 368–373.

LENDERS, V. AND MARTONOSI, M. 2009. Repeatable and realistic experimentation in mobile wireless networks. *IEEE Trans. Mobile Comput. 8*, 1718–1728.

Levis, P. and Lee, N. 2003. TOSSIM: A simulator for tinyOS networks - user manual. http://www.cs.berkeley.edu/~pal.pubs/nido.pdf.

Levis, P., Lee, N., Welsh, M., and Culler, D. 2003. TOSSIM: Accurate and scalable simulation of entire tinyOS applications. In *Proceedings of SenSys'03*. 126–137.

Lin, S., Zhang, J., Zhou, G., Gu, L., Stankovic, J. A., and He, T. 2006. ATPC: Adaptive transmission power control for wireless sensor networks. In *Proceedings of SenSys'06*. 223–236.

Metcalf, C. 2007. TOSSIM live: Towards a testbed in a thread Master's Thesis, Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO.

Meyer, C. 2001. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, PA.

Nguyen, G. T., Katz, R. H., Noble, B., and Satyanarayanan, M. 1996. A trace-based approach for modeling wireless channel behavior. In *Proceedings of the Winter Simulation Conference*. 597–604.

Pawlikowski, K., Jeong, H.-D. J., and Lee, J.-S. R. 2002. On credibility of simulation studies of telecommunication networks. *IEEE Commun. Mag. 40,* 1, 132–139.

Raman, B., Chebrolu, K., Gokhale, D., and Sen, S. 2009. On the feasibility of the link abstraction in wireless mesh networks. *ACM Trans. Netw. 17,* 2, 528–541.

Rappaport, T. 2001. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ.

Reddy, D. and Riley, G. 2007. Measurement based physical layer modeling for wireless network simulations. In *Proceedings of the International Symposium on Modeling, Analysis, and Simulation of Computer Systems*. 46–53.

Reis, C., Mahajan, R., Rodrig, M., Wetherall, D., and Zahorjan, J. 2006. Measurement-based models of delivery and interference in static wireless networks. *SIGCOMM Comput. Commun. Rev. 36,* 4, 51–62.

Rubner, Y., Tomasi, C., and Guibas, L. J. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision 40,* 2, 99–121.

Rusak, T. and Levis, P. 2009. Burstiness and scaling in the structure of low-power wireless links. *SIGMOBILE Mob. Comput. Commun. Rev. 13,* 1, 60–64.

Rusak, T. and Levis, P. A. 2008. Investigating a physically-based signal power model for robust low power wireless link simulation. In *Proceedings of the International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM'08)*. 37–46.

Salamatian, K. and Vaton, S. 2001. Hidden Markov modeling for network communication channels. *SIGMETRICS Perform. Eval. Rev. 29,* 1, 92–101.

Srinivasan, K., Dutta, P., Tavakoli, A., and Levis, P. 2006. Some implications of low power wireless to IP networking. In *Proceedings of the ACM HotNets Conference*. 31–37.

Srinivasan, K., Dutta, P., Tavakoli, A., and Levis, P. 2010. An empirical study of low-power wireless. *ACM Trans. Sen. Netw. 6*.

Srinivasan, K., Kazandjieva, M. A., Agarwal, S., and Levis, P. 2008. The β-factor: Measuring wireless link burstiness. In *Proceedings of SenSys'08*. 29–42.

Towsley, D., Yajnik, M., Moon, S. B., and Kurose, J. 1999. Measurement and modeling of the temporal dependence in packet loss. In *Proceedings of IEEE INFOCOM'99*.

Werner-Allen, G., Swieskowski, P., and Welsh, M. 2005. Motelab: A wireless sensor network testbed. In *Proceedings of IPSN'05*. 68.

Zhao, J. and Govindan, R. 2003. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of SenSys'03*. 1–13.

Zuniga, M. and Krishnamachari, B. 2004. Analyzing the transitional region in low power wireless links. In *Proceedings of SECON'04*. 517–526.

Received May 2011; revised March, September 2012; accepted September 2012