On the Impact of Ignoring Markovian Channel Memory on the Analysis of Wireless Systems

Syed Ali Khayam
NUST Institute of Information Technology
National University of Sciences & Technology (NUST)
Rawalpindi, Pakistan
khayam@niit.edu.pk

Hayder Radha
Department of Electrical & Computer Engineering
Michigan State University
East Lansing, MI 48824 USA
radha@egr.msu.edu

Abstract – **Recent wireless measurement studies have revealed the presence of high-order memory in wireless bit-error channels. However, most wireless studies continue to employ the memory-less or 1st order Gilbert bit-error channels to design, analyze and verify wireless protocols and systems. The inaccuracies incurred by ignoring high-order channel memory are largely unexplored. This paper quantifies inaccuracies incurred by ignoring bit-level wireless channel memory in the context of two simple and commonly-used protocol metrics: (i) packet goodput of an abstract unreliable protocol and (ii) number of retransmissions per packet for an abstract reliable protocol. We analytically derive expected values of these metrics in terms of the parameters of four models with varying levels of memory. We then train the models using actual 802.11b bit-error traces and use the analytical expressions to estimate the metrics. Comparison of the model-based estimates with actual values of the metrics derived from the traces shows that the memory-less and 1st order models incur significant inaccuracies. The remaining two models that capture high-order memory are very accurate in their estimates of the crucial goodput and retransmission metrics.**

I. INTRODUCTION

Wireless channels experience significantly higher bit-error rates than wired channels. Many recent wireless channel measurement studies have analyzed and modeled wireless residual channels at packet- and bit-levels [1]–[9]; the term "residual" referring to the channel observed at the medium access control (MAC) layer after physical layer processing [1]. These studies have shown that residual wireless bit-error and packet-loss channels have memory. Most of the measurement-based studies focus on packet-level channel modeling [1]–[7] and unanimously agree that a two-state Gilbert Markov model [10] that captures 1st order memory can characterize the packet-error random process. Memory of the bit-error random process, however, is of higher orders than the packet-level process [7]–[9].

The burstiness and the consequent memory of wireless channels are well-accepted concepts in the wireless research community. However, much of the contemporary research continues to use memory-less binary-symmetric and 1st order Gilbert channels for bit-level theoretical analysis and experimental evaluation of wireless protocols and applications [11]–[25]. The impacts of these simplistic bit-error channel models on the design and evaluation of wireless systems are largely unexplored.

In this paper, we quantify the impact of channel memory on the performance of two commonly-used and very meaningful wireless performance metrics: the expected goodput of an unreliable protocol and the expected number of per-packet retransmissions for a reliable wireless protocol operating on a single-hop wireless network. We derive these protocol performance metrics in terms of the parameters of four channel models of varying memory-lengths, namely a memory-less binary-symmetric channel (BSC) model, a two-state Gilbert channel (GC) model [10], an order-10 (1024 state) Markov chain [2], [7], and an order-20 constant-complexity model (CCM) [9]. These models are trained using actual 802.11b MAC layer bit-error traces and subsequently the trained models are used to estimate the goodput and retransmissions.

We show that extremely misleading estimates of goodput and retransmissions are obtained when using a BSC or a GC. In particular, for the retransmission metric the results obtained under the memory-less assumption can be orders of magnitude more pessimistic than what is observed on the actual channel. On the other hand, the estimates provided by channel models with high-order memory (i.e., 1024 state Markov and constant-complexity models) are extremely accurate.

The rest of this paper is organized as follows. Section II provides the requisite background and notation that is needed for this work. Section III describes collection of the wireless traces that are used in this paper. Sections 0 and V derive expected values of the goodput and retransmission metrics as functions of the channel models' parameters. The analytical results are substantiated using trace-driven protocol emulations. Section VI summarizes key conclusions of this paper.

II. BACKGROUND AND NOTATION

A. Bit-Error Channel Modeling using High-Order Full-State Markov Chains

Most wireless bit-error random processes are bursty with a memory-length greater than one bit. To make such a process comply with the Markov property [26], wireless studies define a high-order full-state Markov (FSM) chain such that at each time instance the process is characterized by as many bits as its memory-length [2], [7]–[9]. Specifically, at each time instance, a new bit is added to the memory-window and the oldest bit is dropped from the memory-window. Thus the states of a k-th order (memory-length=k) FSM chain comprise 2^k possible combinations of k consecutive bits; that is, each FSM state corresponds to the decimal equivalent of a unique k bit sequence. Transition probabilities between states are computed by sliding a k bit memory-window over the train-

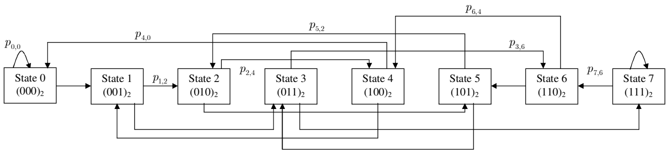

Fig. 1. A 3-rd order (memory-length=3) full-state Markov chain.

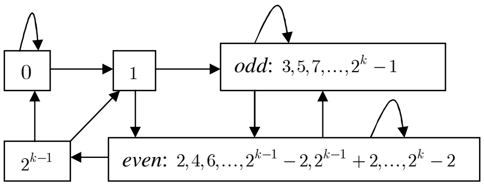

Fig. 2. State aggregation and transitions for the CCM. Each box represents an aggregate CCM state. The number(s) inside a CCM state are the aggregated FSM states [9].

ing data and counting the number of times a bit-pattern $[x_1,x_2,\dots,x_k]$ is followed by another bit-pattern $[y_1,y_2,\dots,y_k]$.

In [8], it was shown that at 2 and 5.5 Mbps, residual bit-errors on an 802.11b LAN have Markovian memory and therefore these errors can be modeled using Markov chains. Furthermore, it was shown in [8] that bit-errors at 11 Mbps have long-range dependence and Markov models cannot characterize these bit-errors accurately. Since the 11 Mbps process is not Markovian, in this paper we focus solely on the bit-errors at the 2 and 5.5 Mbps data rates.

B. FSM Chain Notation

Throughout this paper, $p_{i,j}$ denotes the probability of transiting from current FSM state i to FSM state j. We use π_i to represent the steady-state probability of being in FSM state i. Wireless error traces are treated as a binary time series with elements $x(i)\in\{0,1\}$, where $x(i)=0 \Rightarrow \text{error-free bit}$ and $x(i)=1 \Rightarrow \text{corrupted bit}$. In this paper, FSM states corresponding to even (odd) decimal numbers are referred as even (odd) states. Using the above notation, an example 3-rd order full-state Markov chain is shown in Fig. 1; only transitions to even states are shown. If the FSM chain is in an even state, the last received bit (i.e., the least significant bit position in the memory window) must be error-free. Similarly, an FSM chain in the odd state implies that the last bit was corrupted.

Note in Fig. 1 that due to the binary nature of the underlying wireless bit-error process, each FSM state can transit to only two other states. This is due to the FSM definition in which the memory-window at each time instance is left-shifted by one bit and a one or a zero bit is added to the least-significant bit position. Thus from state i, an FSM chain can transit either to even state $(2i)\bmod 2^k$ or to odd state $(2i+1)\bmod 2^k$. Since the sum of all transitions from a Markov state must sum to one, for any state i we have $p_{(2i+1)\bmod 2^k}=1-p_{(2i)\bmod 2^k}$. It should also be emphasized that once a corrupted bit is received, a k-th order Markov chain will return to state 0 (i.e., the no error state) only from state 2^{k-1} after k transitions; see in Fig. 1 that at state $2^{3-1}=4$, the Markov chain wraps around to state 0.

C. The Constant-Complexity Model

The complexity of FSM chains increases exponentially with respect to memory-length – 2^k state for a k-th order process. To mitigate the exponential FSM complexity, in [9] the authors derived probability distributions for bursts of good and bad bits in terms of FSM chain parameters. Based on these distributions, a constant-complexity model (CCM) was proposed. Fig. 2 shows the CCM's construction.

The CCM aggregates states of an FSM chain of arbitrary order to a five state model. Specifically, FSM states 0, 1 and 2^{k-1} are kept in three isolated states of the CCM. The remaining even FSM states are aggregated into one CCM state, while the remaining odd FSM states are aggregated into another CCM state. It was shown that the CCM captures the good- and bad-burst behavior of wireless channels very accurately [9]. Since the CCM always comprises five states irrespective of the underlying FSM chain's memory-length, behavior of FSM chains with high memory-lengths can be approximated using the CCM.

Throughout the following text, we refer to the five CCM states as c_0, c_1, $c_{2^{k-1}}$, c_{even} and c_{odd}. At any time instance, if the process is in states c_0, $c_{2^{k-1}}$ or c_{even} then the last received bit was error-free. Similarly, the CCM being in state c_1 or state c_{odd} implies that the last received bit was corrupted. The probability of transiting from current CCM state c_i to CCM state c_j is denoted by p_{c_i,c_j}, and π_{c_i} represents the steady-state probability of being in CCM state c_i.

III. DATA COLLECTION

For this study, five wireless receivers were used to simultaneously collect error traces on an infrastructure 802.11b LAN. Wireless receivers were placed at different locations in a room across the hallway from the access point (AP) to simulate a realistic home/office/business setting. The receivers' MAC layer device drivers were modified to pass corrupted packets to higher layers. To capture packets at high transmission rates, packet dissectors were implemented inside the device drivers. These packet dissectors ensured that only packets pertinent to our wireless experiment are processed, while all other packets are simply dropped. Trace collection experiments were repeated multiple times for each configuration. Each experiment comprised of one million packets with a fixed-length payload.

A wired sender was used to send multicast packets with a predetermined payload on the wireless LAN; multicasting disabled MAC layer retransmissions. The sender used different transmission rates ranging from 500 Kbps to 1 Mbps for each experiment. At the physical layer, the auto rate selection feature of the AP was disabled and for each experiment the AP was forced to transmit at a fixed data rate. Each trace collection experiment was repeated multiple times at 2, 5.5 and 11 Mbps physical layer data rates and at different times of day.

For empirical evaluations in this paper, we use five traces each at 2 and 5.5 data rates. These traces are selected to provide maximum diversity in receiver location, packet transmis-

TABLE I. STATISTICS OF 802.11B BIT-ERROR TRACES USED IN THIS PAPER

Physical layer data rate	**Receiver ID**	**Packet Tx rate**	**Bit-error rate**
2 Mbps	1	500 Kbps	0.002127
	2	750 Kbps	0.000245
	3	900 Kbps	0.002127
	4	1024 Kbps	0.002266
	5	1024 Kbps	0.002335
5.5 Mbps	1	500 Kbps	0.003742
	2	750 Kbps	0.002129
	3	900 Kbps	0.002106
	4	1024 Kbps	0.002207
	5	1024 Kbps	0.000066

sion rates and bit-error rates. Some statistics of the traces that are used in subsequent sections are tabulated in Table I.

IV. GOODPUT OF AN UNRELIABLE PROTOCOL

In this section, we quantify the goodput of an abstract unreliable protocol – such as the user datagram protocol [27] – operating over wireless links. Here goodput refers to the ratio between the number of received error-free packets and the total number of transmitted packets. We compare how accurately the following bit-error wireless channel models estimate the goodput of a wireless channel: (i) a memory-less binary-symmetric channel (BSC) model, (ii) a 2-state Gilbert channel (GC) model [10], (iii) a k-th order full-state Markov (FSM) channel model, and (iv) a constant-complexity channel model (CCM) [9]. We focus solely on analyzing the impact of ignoring channel memory at the bit-level because the packet-level models are known to have low-order memory [1]–[7], and therefore can be modeled using memory-less or 1st order channel models. Also, a somewhat similar investigation for packet-level models has already been done in [28]. We first analytically derive packet goodput in terms of the channel models' parameters. We train these models using actual traces and then estimate the traces' goodputs using the trained models. If a model accurately characterizes the bit-error channel then it should provide a goodput estimate that is very close to the trace-based goodput.

A. Goodput of a Wireless Channel

Contemporary wireless stacks perform a checksum on each packet to detect and drop corrupted packets. Thus in this section we assume an abstract protocol that drops all packets with one or more bit-errors. To cater for end-to-end sessions with multiple hops that include a wired (Internet) segment followed by a wireless access segment, we assume that only the last transmission hop is a wireless link. We assume an uncogested path between the sender and the receiver. Also, the wireless hop employs a CSMA/CA mechanism to resolve channel contentions, and therefore the number of collisions is negligible. These assumptions ensure that all packet drops are due to channel noise and interference; i.e., for simplicity of analysis, we ignore packet drops due to congestion or collisions.

Since we define goodput as the ratio between the number of received error-free packets and the total number of transmitted packets, goodput is simply the probability γ of receiving an error-free packet on the wireless channel. Goodput is constrained by $0 \le \gamma \le 1$, where $\gamma = 0$ represents the limiting case when all the received packets have errors and are therefore dropped, and $\gamma = 1$ represents the limiting case when all the received packets are error-free.

We first derive expressions of goodput estimates $\hat{\gamma}$ in terms of the parameters of the trained channel models. Second, we compute the actual goodput γ of the bit-error traces used in this study. Then for each wireless trace, we train all four channel models considered in this paper. Finally, the actual and estimated goodputs (γ and $\hat{\gamma}$'s) are compared.

B. Goodput of a Binary-Symmetric Channel Model

A binary symmetric channel (BSC) is a stateless channel that corrupts every transmitted bit with a probability ε. Consequently, goodput or the probability of receiving an error-free packet of length L over a BSC is simply given by:

$$\hat{\gamma}_{BSC} = \Pr\left\{\text{error-free pkt}\middle|\,\text{BSC}\right\} = (1-\varepsilon)^{L}. \qquad (1)$$

Given training bit-error data, the parameter ε is computed by taking the ratio between the number of bad bits and the total number of bits in the training data. Column 4 of Table I lists the values of ε for the traces used in this paper.

C. Goodput of a Gilbert Channel Model

The Gilbert channel (GC) [10] is a 1st order Markov chain with a good and a bad state. In the present bit-error modeling context, the two Gilbert states jointly capture a process with a memory-length of one bit. The probability of the next (good or bad) bit is dependent on the whether the last received bit was good or bad. Transitions to the good state result in error-free bits, while transitions to the bad state yield corrupted bits. Due to the present notation, we represent the good and bad states as state 0 and state 1, respectively. The GC is completely characterized using two parameters, $p_{0,0}$ and $p_{1,1}$. Although both BSC and GC are special cases of FSM chains, we treat them separately because of their widespread use in wireless studies [11]–[25].

Goodput or the probability of receiving an error-free packet of length L over a GC is given by:

$$\begin{aligned}\hat{\gamma}_{GC} &= \Pr\left\{\text{error-free pkt}\middle|\,\text{GC}\right\}\\ &= \pi_0\left(p_{0,0}\right)^{L} + \pi_1 p_{1,0}\left(p_{0,0}\right)^{L-1} = \left(p_{0,0}\right)^{L-1}\left[\pi_0 p_{0,0} + \pi_1 p_{1,0}\right]\\ &= \pi_0\left(p_{0,0}\right)^{L-1}. \end{aligned} \qquad (2)$$

The above expression shows that the probability of getting a good packet over a Gilbert channel model is simply the probability of starting in the error-free state and then staying in that state for the length of the packet.

D. Goodput of a k-th Order Full-state Markov Channel Model

The probability of receiving an error-free packet of L bits on a k-th order full-state Markov (FSM) channel model is dependent on the present state of the model. If the last received bit was error-free then the least-significant bit in the memory-window will be zero, implying that the FSM chain is in an even state. On the other hand, if the last received bit was corrupted then the FSM chain would be in an odd state.

Let us first focus on the scenario of currently being in an even state and then receiving L consecutive good bits. Throughout this paper, we follow a realistic assumption that $L > k$, where k is the memory-length of the process. Let FSM state $2i$, $0 \le i \le 2^{k-1} - 1$, be the current even state of the FSM channel model. Since all FSM states have the $\bmod 2^k$ operation, unless otherwise stated, we drop the $\bmod 2^k$ operation throughout this paper. Recall that every FSM state i can transit to only two other states. Thus the current state $2i$ can transit to either state $2(2i)$ or state $2(2i)+1$. Since we are only concerned with bursts of error-free bits, the probability of getting an error-free bit starting in state $2i$ is $p_{2i,2(2i)}$. Now for the length of the memory-window, the next $k-1$ transitions will be between even states giving the following states sequence:

$$2i = 2^0(2i) \to 2(2i) = 2^1(2i) \cdots \to \left(2^{k-1}(2i)\right) \bmod 2^k = 0 .$$

Thus after these $k-1$ transitions the process will be in FSM state 0. From that state, to get the remaining error-free bits, the next $L-(k-1)$ transitions will be from state 0 to state 0. To generalize the above discussion in terms of FSM chain parameters, the probability of getting a burst of L good bits starting in FSM state $2i$ is given by $\pi_{2i} \prod_{j=0}^{k-2} p_{2^j(2i),2^{j+1}(2i)} \left(p_{0,0}\right)^{L-(k-1)}$. This probability has to be summed over all possible even FSM states yielding $\sum_{i=0}^{2^{k-1}-1} \pi_{2i} \prod_{j=0}^{k-2} p_{2^j(2i),2^{j+1}(2i)} \left(p_{0,0}\right)^{L-(k-1)}$.

An expression for the probability of getting an error-free packet starting in an odd FSM state can be derived similarly. Adding these expressions gives the goodput of an FSM channel model as follows:

$$\begin{aligned}
\hat{\gamma}_{FSM} &= \Pr\left\{\text{error-free pkt} \middle| \text{FSM}\right\} \\
&= \sum_{i=0}^{2^{k-1}-1} \left[\begin{array}{l} \pi_{2i} \prod_{j=0}^{k-2} p_{(2i)2^j,(2i)2^{j+1}} \left(p_{0,0}\right)^{L-(k-1)} + \\ \pi_{2i+1} \prod_{j=0}^{k-2} p_{(2i+1)2^j,(2i+1)2^{j+1}} \left(p_{0,0}\right)^{L-(k-1)} \end{array} \right] \\
&= \left(p_{0,0}\right)^{L-(k-1)} \sum_{i=0}^{2^{k-1}-1} \left[\begin{array}{l} \pi_{2i} \prod_{j=0}^{k-2} p_{(2i)2^j,(2i)2^{j+1}} + \\ \pi_{2i+1} \prod_{j=0}^{k-2} p_{(2i+1)2^j,(2i+1)2^{j+1}} \end{array} \right].
\end{aligned} \tag{3}$$

The above expression gives the overall probability of getting L consecutive error-free bits by summing over all possible state paths starting in an even or an odd FSM state.

E. Goodput of a Constant-Complexity Channel Model

The constant-complexity model (CCM) [9] aggregates states of the FSM chain as shown in Fig. 2. Recall that we refer to the five CCM states as c_0, c_1, $c_{2^{k-1}}$, c_{even} and c_{odd}. To get a burst of L error-free bits on a CCM-based channel, we have to consider that the CCM can be in any of the five states when the burst starts.

If the CCM is in state c_0 at the start of the burst then the probability that the following L bits are error-free is simply given by $\left(p_{0,0}\right)^L$. If the process is in state c_1, for the next bit to be error-free, the CCM should transit to state c_{even}. This transition has to be followed by $k-3$ good bits, i.e., $k-3$ transitions from c_{even} to c_{even}. After that the CCM should transit to state $c_{2^{k-1}}$ and then to state c_0. Once in state c_0, the process will continue being in that state for the following $L-k$ transitions. Summarizing the above discussion gives probability of receiving an error-free packet starting in state c_1 as $\pi_{c_1} p_{c_1,even} \left(p_{c_{even},c_{even}}\right)^{k-3} p_{c_{even},c_{2^k-1}} p_{c_{2^k-1},c_0} \left(p_{c_0,c_0}\right)^{L-k}$. Similar expressions can be derived for the remaining CCM states. Now summing over all possible initial states gives the complete expression for CCM goodput as

$$\begin{aligned}
\hat{\gamma}_{CCM} &= \Pr\left\{\text{error-free pkt} \middle| \text{CCM}\right\} \\
&= \pi_{c_0} \left(p_{c_0,c_0}\right)^L + \\
&\quad \pi_{c_1} p_{c_1,c_{even}} \left(p_{c_{even},c_{even}}\right)^{k-3} p_{c_{even},c_{2^k-1}} p_{c_{2^k-1},c_0} \left(p_{c_0,c_0}\right)^{L-k} + \\
&\quad \pi_{c_{odd}} p_{c_{odd},c_{even}} \left(p_{c_{even},c_{even}}\right)^{k-3} p_{c_{even},c_{2^k-1}} p_{c_{2^k-1},c_0} \left(p_{c_0,c_0}\right)^{L-k} + \\
&\quad \pi_{c_{even}} \left(p_{c_{even},c_{even}}\right)^{k-2} p_{c_{even},c_{2^k-1}} p_{c_{2^k-1},c_0} \left(p_{c_0,c_0}\right)^{L-k} + \\
&\quad \pi_{c_{2^k-1}} p_{c_{2^k-1},c_0} \left(p_{c_0,c_0}\right)^{L-1} .
\end{aligned} \tag{4}$$

Similar to the FSM expression of (3), the above probability sums over all possible CCM state paths of receiving an error-free packet of length L bits.

F. Comparison of Estimated Goodputs

In this section, we compare the goodput estimates provided by the channel models against the goodput computed from an actual trace. For comparison with a trace, we first train all four models (BSC, GC, FSM and CCM) using that trace. We then plug in the trained parameters of these models into equations (1), (2), (3) and (4) to get throughput estimates of the trace from the models.

Actual and estimated goodputs are compared in Fig. 3. The results in Fig. 3 are reported for the traces of Table I. Results at each physical layer data rate are averaged over five traces. The CCM is trained by aggregating states of an order-20 FSM chain. A packet length of 100 bytes is used to compute actual and estimated goodputs. It can be clearly seen that for both data rates the goodput estimates provided by the binary-symmetric and Gilbert channels are highly pessimistic and inaccurate; at both data rates percentage goodputs estimated by the BSC and the GC are approximately 20% and 30% respectively, while the actual goodput is approximately 97%. Since the Gilbert channel incorporates one bit of memory, its goodput estimate is slightly better than the memory-less binary-symmetric channel. However, both these channels models are too inaccurate to be used in any realistic measurement or analytical study. The order-10 full-state Markov model pro-

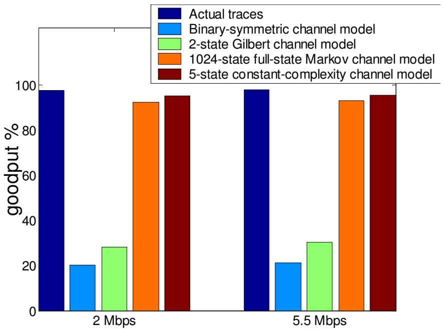

Fig. 3. Comparison of the average goodput of the actual traces with the goodput estimates provided by BSC, Gilbert, 1024-state Markov, and 5-state CCM models; each result is averaged over five traces.

vides very accurate goodput estimates because it incorporates high-order channel memory. While being significantly less-complex than the FSM model, the CCM provides estimates that are even better than the order-10 FSM models because the CCM is constructed by aggregating states of an order-20 FSM chain.

V. RETRANSMISSIONS OF A RELIABLE PROTOCOL

In this section, we show that the expected number of retransmissions per packet can be modeled as a simple function of the goodput. We then compare the retransmission estimates provided by the models under consideration.

A. Expected Retransmissions on a Wireless Channel

In this section, we quantify the expected number of retransmissions experienced by a packet being transported by an abstract reliable protocol – such as the transmission control protocol (TCP) [29] or the 802.11 MAC layer protocol [30]. We only focus on the retransmission-due-to-channel-noise aspect of reliable protocols by employing the following simple abstraction: keep retransmitting until the packet is received correctly. We acknowledge that this abstraction is somewhat unrealistic because reliable protocols generally stop retransmitting after a certain threshold. However, this abstraction allows us to quantify the worst-case performances of the channel models under consideration. Like the previous section, at the receiver the abstract reliable protocol drops all packets with one or more bit-errors. Also, we carry the assumption from the last section that only the last transmission hop is a wireless link.

Let X denote the random variable representing the total number of retransmissions required to successfully transmit a packet under the abstract retransmission protocol. Due to the present abstraction, X can be modeled as a geometric random variable with parameter γ, where γ is defined in the last section as the true probability of a successful packet on the wireless channel. More specifically, the probability that a packet will experience m retransmissions can be expressed as $\Pr\{X=m\}=(1-\gamma)^m\gamma$. Consequently, the expected number of retransmissions β is

$$\beta = \mathrm{E}\{X\} = \frac{1}{\gamma} - 1\,. \tag{5}$$

As expected intuitively, the expected number of retransmission is inversely proportional to the probability of a good packet; increase in the probability of a good packet γ will cause the $1/\gamma$ expression to decrease.

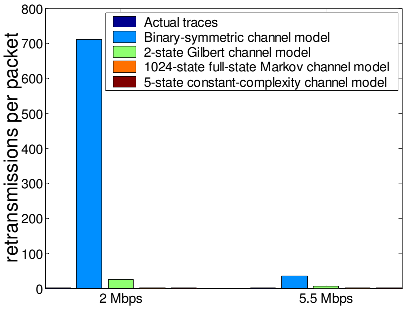

Fig. 4. Comparison of the number of retransmissions per packet estimated by BSC, Gilbert, 1024-state Markov, and 5-state CCM models; each result is averaged over five traces.

Until this point, we have assumed that we accurately know the value γ, the true probability of a successful packet on the wireless channel. In wireless simulations, an estimate of this parameter $\hat{\gamma}$ is provided by a wireless channel model. From the last section, we know that equations (1), (2), (3) and (4) provide the $\hat{\gamma}$ estimates for the BSC, GC, FSM and CCM channel models. Given the $\hat{\gamma}$ estimates, the estimated number of retransmissions per packet can be computed as:

$$\hat{\beta} = \frac{1}{\hat{\gamma}} - 1\,. \tag{6}$$

Plugging in equations (1), (2), (3) and (4) renders each channel model's estimate of per-packet retransmissions.

B. Comparison of Estimated Retransmissions

To compute the average number of retransmissions per packet from an actual trace, we divide the trace into 100 byte packets. Then to emulate transmission of packet i, we count the burst-length of corrupted packets including and following packet i. This burst-length is the number of retransmissions that packet i will experience. Burst-lengths/retransmissions of all the emulated packets are accumulated. Finally, the accumulated retransmission count is normalized by the total number of error-free packet transmissions.

As before, parameters of the channel models are derived from the traces against which they are being compared. Note here that the results of Fig. 4 are not computed by taking the reciprocal of the "averaged" goodput results of Fig. 3. The retransmission estimates are computed by applying equation (5) to a model that is trained specifically for a particular trace. Since equation (5) takes the reciprocal of $0 \le \hat{\gamma} \le 1$, a trace with model with low $\hat{\gamma}$ can render very high values of $\hat{\beta}$.

Fig. 4 plots the average number of retransmissions per packet observed in an actual trace compared against the retransmission estimates provided by the binary-symmetric, Gilbert, full-state Markov and constant-complexity channel models. It can be clearly seen in Fig. 4 that the estimates provided by the BSC model are grossly inaccurate. For instance, at 2 Mbps the BSC models estimates the expected number of retransmissions per packet to be approximately 700 whereas

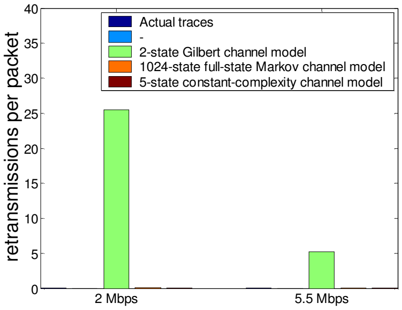

Fig. 5. Number of retransmissions per packet without the BSC model.

the average number of per-packet retransmissions observed in the actual traces is about 0.02. The highly inaccurate retransmission estimates by the BSC are mostly due to receiver-4's traces. The goodput estimate of the BSC model for this trace is approximately 0.0003 at 2 Mbps. Putting this value into equation (5) gives an extremely inaccurate estimate of more than 3000 retransmissions per packet. This simple result shows the scale of inaccuracy that is incurred if channel memory is completely ignored during theoretical or experimental verification of a wireless system.

The estimates of the BSC model are so overwhelming inaccurate that the remaining plots are not clearly visible in Fig. 4. Therefore, in Fig. 5 we plot the results without the BSC model. From Fig. 5, it can be seen that at 2 Mbps even the Gilbert channel provides very inaccurate estimates of the expected number of retransmissions. The GC estimate is closer to the actual traces at 5.5 Mbps, but is still significantly worse than the FSM and CCM models. Fig. 6 only shows the estimates by the 1024-state FSM model and the CCM. Since these channel models incorporate high-order memory, their estimates are quite close to the retransmissions observed in the actual traces.

VI. CONCLUSIONS

We showed that the use of an inaccurate memory-less or low-memory channel model can lead to highly inaccurate theoretical and experimental deductions. Packet goodput and retransmissions were used as example metrics to show that an inaccurate model can render very inaccurate estimates of these metrics. We also showed that models capturing high-order memory can provide accurate estimates of both metrics.

REFERENCES

[1] M. Zorzi and R. R. Rao, "On the statistics of block errors in bursty channels," *IEEE Trans. Com.*, (45)6, 660–667, June 1997.

[2] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A Markov-based channel model algorithm for wireless networks," *ACM Wireless Net.*, (9), 189–199, May 2003.

[3] G. T. Nguyen, R. H. Katz, B. Noble, and M. Satyanarayanan, "A trace-based approach for modeling wireless channel behavior," *Winter Sim. Conf.*, 1996.

[4] H. Balakrishnan and R. H. Katz, "Explicit loss notification and wireless web performance," *IEEE Globecom*, 1998.

[5] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using and 802.11-compliant physical layer," *IEEE Trans. Indus. Elec.*, (49)6, 1265–1282, Dec. 2002.

[6] C. Jiao, L. Schwiebert, and B. Xu, "On modeling the packet error statistics in bursty channels," *IEEE LCN*, Nov. 2002.

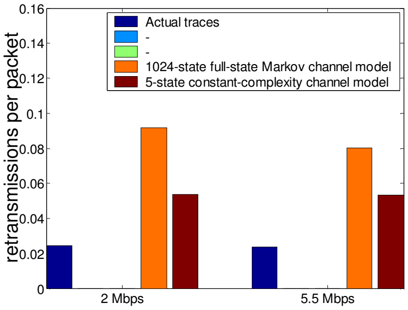

Fig. 6. Number of retransmissions per packet without the BSC and the Gilbert channel models.

[7] S. A. Khayam and H. Radha, "Markov-based modeling of wireless local area networks," *ACM MSWiM*, Oct. 2003.

[8] S. A. Khayam, H. Radha, S. Aviyente, and J. R. Deller, Jr., "Markov and multifractal wavelet models for wireless MAC-to-MAC channels," *Elsevier Performance Evaluation*, vol. 64, no. 4, pp. 298–314, May 2007.

[9] S. A. Khayam and H. Radha, "Constant-complexity models for wireless channels," *IEEE Infocom*, Apr. 2006.

[10] E. N. Gilbert, "Capacity of a Burst Noise Channel," *Bell. Sys. Tech. Journal*, (39), 1253–1265, Sep. 1960.

[11] A. Natu and D. Taubman, "Unequal protection of JPEG2000 code-streams in wireless channels," *IEEE Globecom*, Nov. 2002.

[12] M. Grangetto, E. Magli, G. Olmo, "Reliable JPEG 2000 wireless imaging by means of error-correcting coder," *IEEE ICME*, June 2004.

[13] D. Krishnaswamy and S. Kalluri, "Multi-level weighted combining of retransmitted vectors in wireless communications," *IEEE VTC*, 2006.

[14] C. E. Koksal and H. Balakrishnan, "Quality-aware routing metrics for time-varying wireless mesh networks," *IEEE JSAC*, to appear.

[15] J. Farber and K. Zeger, "Optimality of the natural binary code for quantizers with channel optimized decoders," *IEEE ISIT*, July 2003.

[16] W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Error resilience in video and multiplexing layers for very low bit-rate video coding systems," (15)9, 1764–1774, 1997.

[17] X. Luo and G. B. Giannakis, "Energy-constrained optimal quantization for wireless sensor networks," *IEEE SECON*, Oct. 2004.

[18] M. U. Ilyas and H. Radha, "End-to-end channel capacity of a wireless sensor network under reachback," *CISS*, Mar. 2006.

[19] M. Godavarti and A. O. Hero III, "Diversity and degrees of freedom in wireless communications," *IEEE ICASSP*, May 2001.

[20] H. Dong, I. D. Chakares, A. Gersho, E. Belding-Royer, and J. D. Gibson, "Selective bit-error checking at the MAC layer for voice over mobile ad hoc networks with IEEE 802.11," *IEEE WCNC*, Mar. 2004.

[21] L. Bononi, M. Conti, and E. Gregori, "Runtime Optimization of IEEE 802.11 Wireless LANs Performance," *IEEE Trans. Paral. and Distrib.*, (15) 1, 2004.

[22] C-F. Chiasserini and E. Magli, "Energy-Efficient Coding and Error Control for Wireless Video-Surveillance Networks," *Telecom. Sys.*, vol. (26) 2, 369–387, 2004.

[23] Z-H. Tan, P. Dalsgaard, and B. Lindberg, "A subvector-based error concealment algorithm for speech recognition over mobile networks," *IEEE ICASSP*, May 2004.

[24] W. S. Lee, M. R. Frater, M. R. Pickering, and J. F. Arnold, "A robust codec for transmission of very low bit-rate video over channels with bursty errors," *IEEE Trans. CSVT*, (10) 8, 1403–1412, Dec. 2000.

[25] L. Zhong, F. Alajaji, and G. Takahara, "A queue-based model for wireless Rayleigh fading channels with memory," *IEEE VTC*, Sep. 2005.

[26] J. G. Kemeny and J. L. Snell, "Finite Markov Chains," Springer-Verlag: New York, 1976.

[27] J. Postel, "User datagram protocol," RFC 768, Aug. 1980.

[28] J. McDougall and S. Miller, "Sensitivity of wireless network simulations to a two-state Markov model channel approximation," *IEEE Globecom*, Dec. 2003.

[29] J. Postel, "Transmission control protocol," RFC 793, Sep. 1981.

[30] IEEE Std. 802.11b-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz band," Sep. 1999.