Efficient Packet Error Rate Estimation in Wireless Networks

Bo Han
Department of Computer Science
University of Maryland
College Park, MD 20742, USA
bohan@cs.umd.edu

Seungjoon Lee
Department of Computer Science
University of Maryland
College Park, MD 20742, USA
slee@cs.umd.edu

***Abstract*—Wireless links exhibit diverse link quality (e.g., in terms of error rate, latency, and required transmit power), and inferring the link cost correctly is crucial to finding an efficient path in multihop wireless networks. In this paper, we consider the problem of estimating packet error rate (PER) in wireless networks. Most existing schemes use observed error rates of probe messages as representative packet error rates for wireless links. However, since typical data packets are larger than probe messages, the actual data packet error rate is often higher than the observed error rate of probe messages. Instead of directly using measurement statistics, our approach is to analyze them to determine the link characteristics and estimate the link quality (e.g., PER) based on the understanding. We describe a simple estimation scheme based on a well-known two-state Markov bit error model for wireless channels. We perform various experiments on two wireless testbeds and show the proposed scheme can successfully estimate PERs for packets of arbitrary size in diverse environments. When compared to other alternatives, the proposed estimation scheme is accurate and incurs small control overhead. We also integrate the scheme into an existing routing scheme. Our experiment results show that the combined scheme can reduce the transmission overhead by 60% when compared to a naive estimation strategy based on one type of probe messages.**

I. Introduction

In multihop wireless networks, end nodes typically relay non-local messages to achieve end-to-end connectivity between arbitrary source and destination. In earlier *routing* schemes, the goal was to find a path with minimum hop count [1]–[3]. In practice, however, different wireless links show wide diversity in quality [4]–[6], and more recent schemes attempt to find a path with low cost, for example, based on energy consumption, latency, and link error rate. In particular, wireless transmissions are typically more prone to bit errors than wired ones, and several routing schemes consider packet error rates over wireless links to improve message delivery performance [5], [7]–[9].

To benefit from those proposed schemes and achieve actual performance improvement, we need to be able to assess link quality (e.g., data packet error rate) accurately and preferably with low overhead. The most popular assessment strategy is to use probe messages (e.g., periodic HELLO messages) and use the observed error rate for all packets sent over the wireless link [5], [6], [10]. One problem with this approach is that probe messages are often shorter and less prone to bit-error-induced losses than typical data packets. Therefore, the estimated error rate based on probe messages can be significantly lower than the actual error rate for data packets, especially in the so-called communication *gray zone* [4]. For example, in our experiments, average packet error rates for a particular wireless link were around 14% for 16-byte messages and more than 37% for 1024-byte messages. Clearly, path selection based on the inaccurate estimation will lead to suboptimal performance. An alternative is to use probe messages with different sizes (e.g., 128-, 256-, 512-, and 1024-byte probe messages) and use the observed PER for a similar size as an approximation. However, this approach (called *multi-probe* approach in this paper) will incur higher overhead due to increased message length and count.

We consider the accurate estimation of packet error rates (PERs) over wireless links and its application to efficient routing in multihop wireless networks. Our general approach is that instead of directly using measurement values, we analyze them to determine the channel characteristics and estimate packet error rates based on the understanding. In this paper, we present a simple estimation strategy that employs a well-known two-state Markov bit error model for wireless channels [11], [12]. This scheme extrapolates data packet error rates based on the statistics of a few different types of probe messages. We use two wireless testbeds to measure packet error rates for various links and demonstrate that the strategy can successfully estimate error rates for packets of arbitrary size in diverse wireless environments. We also illustrate how to incorporate this strategy into an existing routing scheme in practice. In one of our experiments, when used with an existing routing scheme, this estimation technique leads to 60% lower data transmission overhead when compared to the naive estimation strategy using probe messages of a single type. We also discuss possible approaches to further improve estimation performance with smaller overhead.

The rest of this paper is organized as follows. In Section II, we review some of related work. We present our estimation scheme and discuss issues for practical deployment in Section III. In Section IV, we report experiment results on two wireless testbeds and demonstrate the estimation performance of the proposed scheme. We describe how to integrate the estimation scheme into an existing routing protocol and present simple experiment results in Section V. We conclude and present our future directions in Section VI.

II. Related Work

Packet transmissions in wireless networks are more prone to bit errors than in wired networks, and significant research efforts have been made to understand and model wireless errors. Although some protocols and coding schemes assume bit errors are independent and identically distributed, Markov models with finite states have been popular. Among such models, the two-state Gilbert/Elliot model [11], [12] is best known due to its simplicity yet reasonable accuracy, which we describe in more detail in Section III-A. Kopke et al. [13] propose to use a chaotic map as a model for bit errors over wireless channels and describe how to determine the model parameters based on measurement data. To use this chaotic map model, however, we need the information of bit error processes, which is difficult to obtain from practical packet-level communications. Also, we need to determine more parameters to employ the model, which can be more complex and error-prone in practice. Although Kopke et al. present results that the Gilbert/Elliot model sometimes does not accurately model the bit error processes, our experiment results in Section IV indicate that the Gilbert/Elliot model is a good approximation that can be efficiently implemented.

A number of researchers have performed experiments to validate Markov models and utilize them for packet error simulation. Arauz and Krishnamurthy [14] validate the traditional two-state Markov models using experimental traces in various IEEE 802.11-based network envirnoments. Hartwell and Fapojuwo [15] train Markov models with different number of states using 802.11a measurement traces. Then, they use the trained Markov models to generate simulated traces and compare the generated traces with the original traces. Willig et al. [16] present results of bit error measurements using an IEEE 802.11-compliant radio modem in an industrial environment. They show that the popular Gilbert/Elliot model and its slight modification are useful for simulating bit errors on a wireless link. The main focus of their work is to simulate wireless errors more accurately. In contrast, our goal is to propose an estimation scheme for packet error rates based on the wireless error models.

Other researchers report measurement results in various environments. Using AT&T WaveLAN wireless interfaces, Eckhardt and Steenkiste [17] characterize packet errors and evaluate the effects of interference and attenuation due to distance and obstacles on the packet loss rate and bit error rate. So et al. [18] report results from a series of experiments designed to investigate loss behavior of broadcast messages in a wireless sensor network. Zhao and Govindan [19] report on a systematic medium-scale measurement of packet delivery performance in three different environments: an indoor office building, a habitat with moderate foliage, and an open parking lot. Aguayo et al. [6] analyze the causes of packet loss in a 38-node urban multihop 802.11b network. They find that link error rates stay relatively uniform for the majority of links. These efforts attempt to determine wireless link characteristics through measurements, while we propose an estimation strategy built on top of these previous findings.

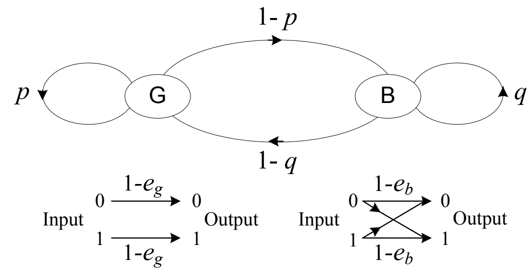

Fig. 1. The two-state Markov channel model by Gilbert and Elliot. G denotes *good* state, and B denotes *bad* state.

Recently, attention has been paid to finding high-quality paths in the face of error-prone wireless channels. Specifically, some new routing metrics for multihop wireless networks take link quality into consideration. For example, De Couto et al. [5] propose a link metric called *ETX (Expected Transmission Count)*, which corresponds to the expected number of transmissions required to successfully deliver a packet over the wireless link. This metric requires per-link measurements of packet loss rates in both directions of a wireless link. Draves et al. [7] propose *WCETT (Weighted Cumulative Expected Transmission Time)*, which assigns weights to individual links based on *ETT (Expected Transmission Time)* of a packet over the link. ETT is a function of the loss rate and the bandwidth of the link. For minimum-energy routing in wireless networks, Banerjee and Misra [20] define a link cost as a function of both the link error rate and the energy required for a single transmission attempt across the link. Lee et al. [8] propose a new link metric for geographic routing, which we describe in more detail in Section V. Compared to these works, our work presents a PER estimation strategy and is complementary to the practical implementation of such routing schemes.

III. Packet Error Rate Estimation

Our PER estimation technique is based on the Gilbert/Elliot bit error model. In this section, we describe the model first and the estimation technique next.

A. Gilbert/Elliot Model

In the Gilbert/Elliot (GE) model [12], a wireless channel is in one of the following two states: *good* and *bad* (Figure 1). If the channel is in good state, then a bit transmission error occurs with the probability of e_g. On the other hand, if the channel is in bad state, the probability of bit transmission error is e_b. Prior to the transmission of each new bit, the channel may change states or remain in the current state. Figure 1 shows the GE model representation with state-transition probabilities. We can easily obtain the following steady-state probabilities:

$$P_G = \frac{1-q}{2-(p+q)}, \quad P_B = \frac{1-p}{2-(p+q)}, \tag{1}$$

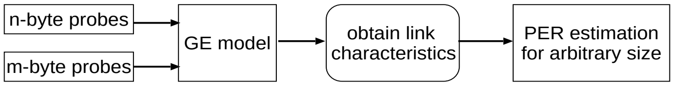

Fig. 2. Proposed estimation strategy. We use two observed error rates of two different types of probe messages to determine the link characteristic parameters of the GE model.

where P_G and P_B are steady-state probabilities for *good* and *bad* states. In this paper, we use $e_g = 0$ and $e_b = 1$ for simplicity.

In this model, the PER of n-byte message is:

$$PER(n) = 1 - (P_G\ p^{8n} + P_B(1-q)p^{8n-1}). \qquad (2)$$

Note that there are two cases where no bit error occurs in a packet. First, the channel is initially in good state and remains there for all bit transmissions, and this probability is $P_G\, p^{8n}$ for an n-byte packet. In the other case, the channel is initially in bad state, but the channel changes into good state for the first bit transmission and remains in good state. This probability is $P_B\ (1-q)p^{8n-1}$. A packet error occurs if neither of them happens, and hence Eq. 2.

B. Estimation Strategy

In this subsection, we describe how we determine the link characteristics based on the GE model and estimate packet error rates using the determined link characteristics. Figure 2 illustrates the overall approach. Note that the simplified GE model has only two characteristic parameters, and we use statistics from two distinct types of probe messages to solve two instances of Eq. 2 and obtain p and q. Then, when we send a L-byte data packet, we can use the obtained parameters (p and q) and Eq. 2 to estimate the expected error rate. Specifically, given $PER(m)$ and $PER(n)$ from m-byte and n-byte probe messages, we can calculate p as follows:

$$p = \left(\frac{1 - PER(n)}{1 - PER(m)}\right)^{\frac{1}{8(n-m)}}$$

Then, we can estimate the PER of L-byte data messages using the following formula:

$$PER(L) = 1 - (1 - PER(m))\left(\frac{1 - PER(n)}{1 - PER(m)}\right)^{\frac{L-m}{n-m}} \qquad (3)$$

As illustrated in Section IV, if we obtain good statistics from probe messages, we can estimate expected error rates for packets of arbitrary size with reasonable accuracy.

Overhead Comparison with the Multi-probe Approach: Since this extrapolation-based technique uses only two probe types, when compared to the *multi-probe* strategy, it typically incurs less estimation overhead due to fewer and smaller probe messages. Our approach also can be beneficial when a node needs to inform its neighbors of reverse link quality due to asymmetry [5], [8]. For example, although node A knows the observed link quality for $B \to A$, it needs to know the link quality for $A \to B$, which A should learn from B. In the *multi-probe* approach, B needs to inform A of multiple error rates for various packet sizes (e.g., 4 different error rates for 128-, 256-, 512-, and 1024-byte packets). Since a periodic broadcast should typically include such information for all neighbors, the control messages can be considerably large when the node has many neighbors. With our estimation scheme, the node needs to send appropriate p and q only, not observed error rates for multiple packet sizes; as a result, the size of control message can be a fraction when compared to the *multi-probe* case.

C. Using Existing Periodic Messages

Many existing protocols use periodic message exchange [3], [7], [21], [22], and our estimation technique can take advantage of it. Depending on the protocol, the periodic message can include various information about sender, neighbors, and links between them [7], [21], [22], which can be quite long. Another major objective of such periodic messages is to inform whether a node is within the communication range of other nodes (e.g., when nodes are mobile) [21], [22]. Since our estimation technique needs two measurement values, we propose that two different types of periodic messages be used and sent in an alternate manner. The shorter message type includes only the ID of transmitting node, which is mainly to inform reachability with lower overhead. The longer message type will include all the protocol-specific information. From the two types of messages, we will be able to apply our technique to understand the channel characteristics.

When there are no such periodic messages used in other protocols, we need to send our own probe messages. This incurs control overhead, but will be typically less than that of the *multi-probe* approach because our estimation technique typically uses fewer types of probe messages. To minimize the control overhead due to periodic messages, Zhang et al. [23] propose to infer link quality by using data packet transmissions only. However, their scheme probabilistically sends data packets to a set of neighbors to obtain up-to-date link quality information, which can lead to inefficient use of wireless resource. The relative performance of such data-driven and beacon-based link estimation schemes will depend on various aspects such as specific applications and network environments.

We next discuss a few issues with the efficient implementation of the scheme in practice.

D. Discussions

Estimation Accuracy: Since our estimation technique is based on extrapolation, small measurement errors for shorter probe messages will amplify the estimation error for longer data packets. To avoid this, we often need to collect statistics from a significant number of probe message to obtain reliable measurements. In Section IV, we empirically show how many probe message are needed to estimate PER with reasonable accuracy. Another issue arises when we use existing protocol-specific periodic messages as probes. Unless a network is static, the probe message length may change over time (e.g., due to change in the number of neighbors). Then, we will have probe messages of various sizes, and using probe messages of any two fixed sizes may not lead to sufficiently accurate estimation. In this scenario, applying a regression analysis technique will be an interesting approach to investigate.

Unicast vs. Broadcast: In our PER estimation, we use broadcast probe messages to infer the PER of unicast packets. The difference in the two transmission mechanisms can potentially affect the estimation accuracy [23]. Wireless packet errors may be caused by collisions. In IEEE 802.11 networks, RTS and CTS control messages exchange can reduce the collision-induced errors of actual unicast data messages [24]. However, since broadcast messages do not have such a virtual carrier-sense mechanism, they are more prone to collision-based errors than unicast messages. One possible approach will be to leverage existing schemes that infer active stations [25] and take the collision probability into account [24]. Moreover, multiple data transmission rates are supported in IEEE 802.11-based networks. Probe messages are often broadcast typically at a lower rate such as 1 Mbps. In contrast, unicast messages can be sent at a higher transmission rate (up to 54 Mbps). We will discuss the relationship between packet error rates and data transmission rates in more detail in Section IV-B.

Other Wireless Channel Models: Our estimation technique uses packet-level loss information. As in many bit error models, however, if we use the knowledge of bit-error processes, then we can potentially understand the channel characteristics better and estimate packet error rates more accurately. For example, applying the chaotic map-based model [13] or the bipartite models [26] might lead to better estimation accuracy. However, while it is relatively easy to detect *whether* any bit error has occurred (i.e., packet corruption detected by checksum), it is more difficult and expensive in practice to learn *which* bits are corrupted. To model wireless channel more accurately, there are also some other Markov models which consider wireless channel fading [27], [28]. But these models are more complicated and use more states than the GE model. Another possible approach for more accurate packet error rate estimation is to use above mentioned Markov models. In general, however, we need to determine more parameters in such models, which potentially can lead to more types of probe messages and higher control overhead.

IV. TESTBED EXPERIMENTS

In this section, we present results from our experiments performed on real testbeds and demonstrate that our estimation strategy performs well in various wireless environments.

A. Experiment Setup

We have performed our experiments in two open access wireless testbeds: Emulab (http://www.emulab.net) and ORBIT (http://www.orbit-lab.org). Although Emulab is often used to provide emulated network environments for wired networks experiments, the Emulab wireless testbed uses *real air communication* through IEEE 802.11 wireless interfaces between stationary PC nodes scattered around a typical office building. We only use the nodes on the third floor (Figure 3). Each PC has two Netgear WAG311 wireless interface cards based on the Atheros 5212 chipset. It uses Redhat 9.0 with 2.4 kernel and the MadWifi open-source device driver[1]. The ORBIT testbed currently consists of 400 wireless nodes, each equipped with two IEEE 802.11 wireless cards laid out in a 20-by-20 grid with approximately one meter spacing between nearby nodes. Due to the relatively small deployment area, observed packet error rates in ORBIT show less diversity. Thus, although we present results from ORBIT, we focus on results from Emulab to illustrate that the estimation technique performs well for both low-error and high-error links.

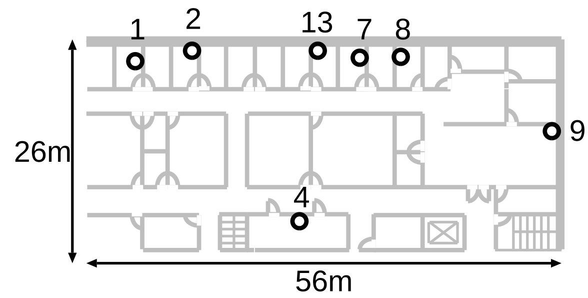

Fig. 3. Partial floorplan for the Emulab wireless testbed. Nodes 1 and 9 cannot directly communicate, and all the other node pairs can talk to each other.

In our experiments, a sender broadcasts 16, 32, 64, 128, 256, 512 and 1024-byte UDP packets every 0.05 seconds in an intermixed fashion to minimize the effect of link condition variation over time on the error rates of different message types. In our experiments, we use only one sender at any instant to minimize the interference and collisions. Each sender broadcasts 10000 packets for each size (70000 packets total). All nodes receiving the packets record the packet size and sequence number to calculate the observed PERs for each message type. In this paper, without otherwise mentioned, we use the fixed transmission rate of 1 Mbps, the default broadcast transmission rate in the MadWifi device driver, for all messages. But we also evaluate the estimation performance for other different data transmission rates in Section IV-B. The transmit power is fixed at 31 mW, which is the default value in the device driver.

We compare the estimation performance of the following strategies:

- BASIC(m): This scheme uses the average error rate of m-byte probe messages for data packets of all sizes.
- INDEP(m): This scheme assumes the independent bit error model and extrapolates the expected packet error rate based on the statistics of m-byte probe messages.
- GE(m, n): This is the proposed scheme based on the GE

[1] http://www.madwifi.org

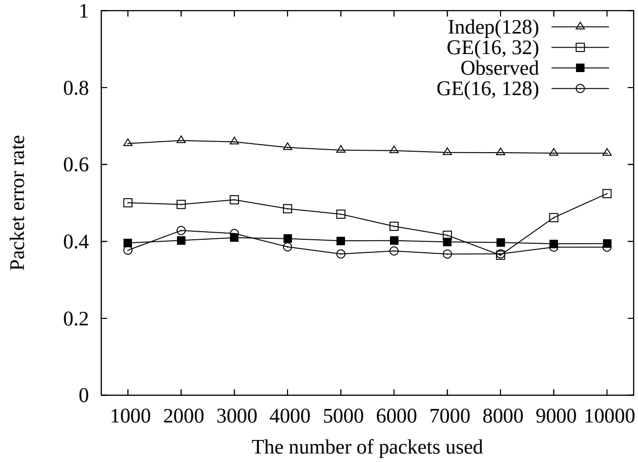

Fig. 4. Estimated and observed PERs for 1024-byte packets over the link from node 1 to node 4 in Emulab (Figure 3).

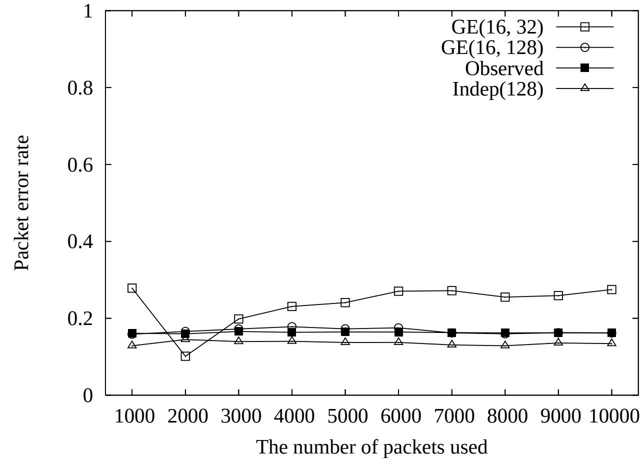

Fig. 5. Estimated and observed PERs for 1024-byte packets over a link in ORBIT.

model, which uses the statistics of m-byte and n-byte probe messages.

- OBSERVED: This is the actual observed packet error rate.

For the INDEP scheme, we can estimate PERs as follows. Suppose we know the observed value of $PER(m)$ from m-byte probe messages. To infer the PER for L-byte data messages, we can use $PER(L) = 1 - (1 - PER(m))^{L/m}$. Only one measurement value is required for INDEP; GE uses two parameters, and there can be more possible combinations of the two. For both schemes, proper parameter choice can be crucial to correct PER estimation. We consider three different combinations of parameters for GE and two different cases for INDEP and compare the estimation performance.

B. Experiment Results

We first consider how well the above estimation strategies perform. In Figure 4, we plot the observed error rate for 1024-byte packets and estimated error rates by different schemes[2]. We use a representative experiment sending 10000 packets for each probe type, and each point in the figure is based on cumulative packet error rates after every 1000 packets. In Figure 4, the estimation by GE(16,128) closely matches the actual average packet error rate. In general, we observe that the estimation error for GE(16,128) becomes smaller as we use more probe messages; we discuss this issue later in more detail.

In our experiments, GE(16,32) does not perform as well as GE(16,128). In Figure 4 there is considerable difference in the estimated value over time, and the measurement error is often relatively large. One possible explanation is that the estimation by GE(16,32) is less robust because we use extrapolation based on two relatively nearby sample points; a small measurement error can amplify the estimation error. Also, Kopke et al. [13] find that there is difference in bit error probability depending on the bit position, and bit errors occur more frequently at the beginning of a packet. As a result, estimation using short probe messages alone can potentially lead to higher estimation errors. In Figure 4, INDEP does not estimate PER correctly, and although not shown in the figure, the estimation error by INDEP(16) is larger than that of INDEP(128). Although we do not show all the results here, we have experimented with other links and performed multiple experiments for each link, and the results are similar. We later present some of them in Table I.

[2]We include additional 84 bytes of lower layer headers in the calculation.

	Emulab Links						
	8→9	1→13	1→7	1→4	1→8	11→16	16→5
OBSERVED	0.018	0.135	0.145	0.334	0.375	0.548	0.738
GE(16,128)	**0.021**	**0.131**	**0.145**	**0.385**	**0.393**	**0.526**	**0.754**
GE(16,64)	0.025	0.222	0.247	0.465	0.332	0.415	0.791
GE(16,32)	0.046	0.154	0.043	0.524	0.243	0.594	0.677
INDEP(128)	0.052	0.222	0.255	0.629	0.645	0.907	0.996
INDEP(16)	0.092	0.332	0.383	0.816	0.831	0.993	1.000
BASIC(128)	0.010	0.047	0.055	0.173	0.180	0.385	0.646

TABLE I

COMPARISON OF DIFFERENT ESTIMATION TECHNIQUES AGAINST ACTUAL PACKET ERROR RATES. WE USE 10000 PACKETS FOR EACH OF PROBE AND DATA MESSAGE TYPES. VALUES IN BOLD REPRESENT THE CASES WITH MINIMUM ESTIMATION ERROR.

We have performed experiments on the ORBIT testbed. In Figure 5, we plot the observed and estimated error rates for 1024-byte obtained from ORBIT. In ORBIT, all the nodes are relatively close to each other, and in the figure, we use the link with the highest link error rate. We observe the trend shown in Figure 5 (from ORBIT) is similar to the one in Figure 4 (from Emulab); GE(16,128) performs better than GE(16,32). The performance of INDEP(128) in this particular experiment is better than in Figure 4. In the rest of this section, we use results from Emulab only.

Experiments with Various Links: In the previous results, we considered results only from a few links. We now present results from various wireless links with diverse link quality. In Table I, we report estimated PERs by different schemes

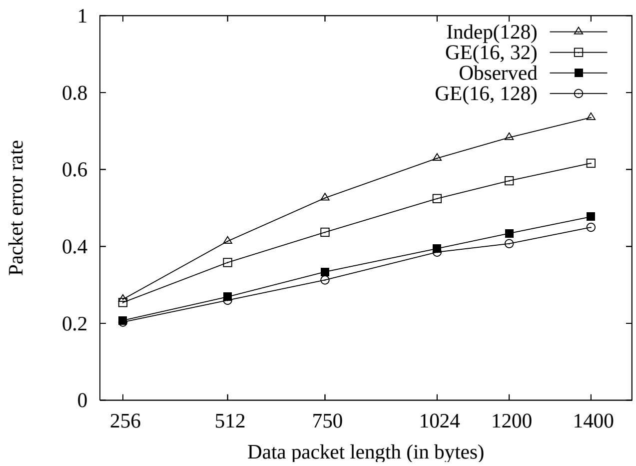

Fig. 6. PER estimation for different packet sizes. Data transmission rate is 1M. We use the link from node 1 to node 4. Again, GE(16,128) performs best for all packet sizes.

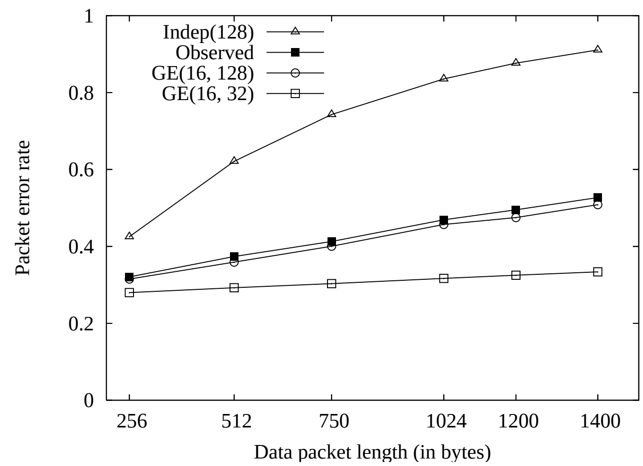

Fig. 7. PER estimation for different packet sizes. Data transmission rate is 2M. Again, we use the link from node 1 to node 4. GE(16,128) performs best for all packet sizes.

as well as observed error rates for 1024-byte packets[3]. We observe that GE(16,128) estimation is the most accurate in all cases (highlighted in bold), and the estimation error is small regardless of link quality. GE(16,64) often performs better than GE(16,32), but both of them result in larger estimation errors than GE(16,128). As in Figure 4, INDEP leads to large estimation errors, while INDEP(128) performs better than INDEP(16). Although the independent bit error model has served as a reasonable model in [29], it does not seem to reflect the channel characteristics correctly in our indoor experiments. BASIC(128) uses the error rate of 128-byte probe messages as the estimation for 1024-byte packets, which results in significant underestimation. In Section V, we illustrate that this underestimation by BASIC can lead to significant inefficiency when used with existing routing schemes.

Varying Data Packet Sizes: In the previous experiments, we fixed the data packet length to 1024 bytes. In this set of experiments, we vary the data packet size and compare the estimated and observed error rates. In this experiment, we use additional packet sizes (750, 1200, and 1400 bytes). In Figure 6, we plot the estimated and actual packet error rates with varying packet sizes. We use the statistics of 10000 messages for each probe type. Not surprisingly, average packet error rates increase as data packets become larger. We observe that GE(16,128) again performs best in estimating error rates for all packet sizes. Other schemes show similar trends to Figure 4; GE(16,64) performs worse than GE(16,128), while INDEP performs worst. This result illustrates that our proposed technique estimates error rates for various packet sizes.

Varying Data Transmission Rate: IEEE 802.11-based networks provide multiple bit-rates for data transmission. Rate adaptation algorithms such as Auto Rate Fallback (ARF) [30] are needed to achieve higher performance under varying conditions. Among these rates, 1M, 2M, 5.5M and 11M use direct sequence spread spectrum (DSSS), while the other transmission rates such as 6M, 12M and 54M employ orthogonal frequency division multiplexing (OFDM). Generally, high quality links can use higher bit-rates to transmit more data and lower bit-rates usually have a lower packet error rate on low quality links. In previous research work, it was pointed out that, due to different modulation techniques used, there are many links that can operate at lower bit rate but not at higher bit rate and vice-versa [31].

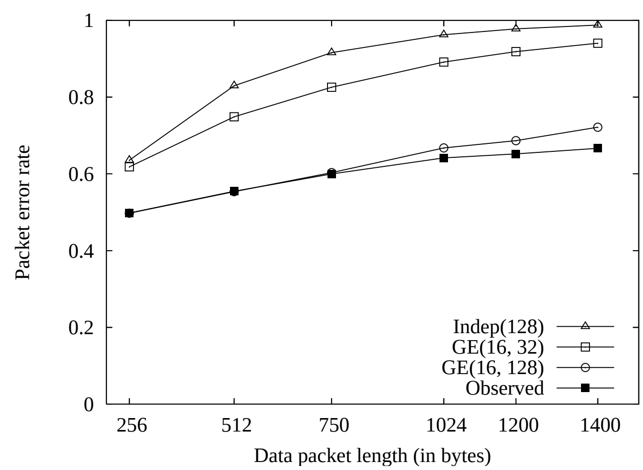

Fig. 8. PER estimation for different packet sizes. Data transmission rate is 11M. We still use the link from node 1 to node 4. GE(16,128) performs best for all packet sizes.

In the previous experiments, we fixed the data transmission rate to 1M. In this set of experiments, we vary the data transmission rate and compare the estimated and observed error rates. All the other configurations are the same as those in the experiment for Figure 6. In Figure 7 and Figure 8, we plot the estimated and actual packet error rates for different

[3]Nodes 5, 11, and 16 are not shown in Figure 3. The full floorplan is available at https://www.emulab.net/floormap.php3.

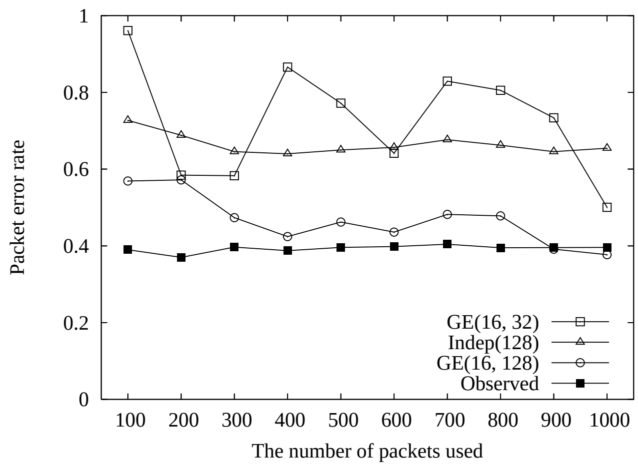

Fig. 9. PER estimation based on 1000 packets. We use the link from node 1 to node 4 again, but the values on the X-axis are smaller than those in Figure 4.

packet sizes at 2M and 11M bit-rates, respectively. We find that GE(16,128) still performs best in estimating error rates for those two bit-rates. We also evaluate the estimation performance for 5.5M bit-rate and get similar result. However, because of the limitation of Emulab testbed, for other data transmission rates such as 6M, 12M and 54M, only a few links work. As a whole, Figure 6, Figure 7 and Figure 8 show that our proposed technique can estimate error rates for various DSSS-based data transmission rates. In general, it will be useful to understand the relationship among error rates for packets sent at different transmit rates over the same wireless link [6], which we plan to investigate further in the future.

Convergence Time for Accurate Estimation: One of our goals is to estimate error rates quickly with small overhead. In this set of experiments, we look into the number of probe messages needed to achieve reasonable estimation accuracy. As mentioned in Section III, small measurement errors in probe messages can cause large extrapolation errors. However, sending 1000 probe messages (to obtain the first point in Figure 4) takes tens of minutes if we send a probe message every second. In this experiment, we calculate the estimation using the cumulative statistics after every 100 probe messages. One issue with inferring error rates based on a small number of packets is that the observed error rate of 16-byte packets is sometimes higher than that of 128-byte packets, which is contrary to the trend shown in Figure 6. In that case, when we apply Eq. 2, the estimation is often negative for longer data packets. Clearly, it is due to limited number of samples, and we are unlikely to have a good estimation by blindly applying Eq. 2 with only a small number of probe messages. In such a case, we use the maximum of the following three values as the estimated error rate: PER(16), PER(128), and the estimated PER from Eq. 2.

In Figure 9, we consider the estimation performance when we use a smaller number of probe messages. We observe that GE(16,128) converges after 300 probe messages, while GE(16,32) shows a substantial amount of fluctuation. Still, GE(16,128) takes several minutes before achieving reasonable convergence if we send a probe message every second. This amount of time is acceptable for more static wireless mesh networks [6], while more dynamic wireless networks such as ad hoc networks may require faster convergence. We plan to investigate how to reduce the number of required probe messages further in the future. As mentioned in Section III, one possibility is to experiment with regression analysis techniques combined with observed error rates for actually transmitted data packets.

V. Application to Existing Routing Schemes

This paper is part of our work to implement a geographic routing scheme that can find an efficient path. In this section, we briefly describe the proposed framework for efficient geographic routing [8] and present results when we integrate the estimation strategy into the framework.

A. Background: Efficient Geographic Routing

In geographic routing, nodes use location information of neighbors and destination to choose next hop nodes [3], [32]. The most popular strategy in geographic routing is the greedy forwarding—the current node S greedily selects the neighbor that is closest to destination T whenever possible. Let us consider the amount of decrease in distance by a neighbor n, which is called the *advance* (*ADV*) of n: $ADV(n) = D(S) - D(n)$, where $D(x)$ denotes the distance from node x to T. Then, the above greedy strategy tries to *maximize* the ADV of next hop without considering link quality. Lee et al. [8] propose to use a new metric called NADV (*normalized advance*), defined as ADV divided by link cost, i.e., $NADV = ADV/Cost$. In particular, if we consider packet error rate and use *Cost*=ETX=1/(1-PER), we have the following link metric[4]:

$$NADV = \frac{ADV}{Cost} = ADV(1 - PER). \qquad (4)$$

Then, among the available neighbors, we choose the neighbor with largest NADV. Intuitively, NADV denotes the amount of advance achieved per unit cost. We can show that NADV can find optimal paths under certain idealized conditions and achieve significant performance improvement in more realistic scenarios. (See [8] for details.)

B. Experiment Results

We have modified the geographic routing implementation from USC[5] to account for link cost when choosing next hops. We installed the modified code at the nodes shown in Figure 3. In our experiments, node 9 is the destination, and node 1 is the source sending 1000 UDP packets (1024 bytes each) at the rate

[4]Although ETX considers the reverse error rate for ACK [5], we ignore it here for simplicity.

[5]Available at http://enl.usc.edu/software.html

Next Hop	ADV	NADV		Number of Retransmissions		
		BASIC(128)	GE(16,128)	at source	at relay	total
4	19.16	15.85	11.78	536	188	724
7	25.96	24.54	**22.19**	185	119	**304**
8	**30.00**	**24.61**	18.20	768	4	772
13	20.55	19.59	17.85	250	171	421

TABLE II
ROUTING METRICS BASED ON DIFFERENT ESTIMATION SCHEMES AND ACTUAL ROUTING PERFORMANCE. VALUES IN BOLD CORRESPOND TO THE BEST CHOICES UNDER DIFFERENT CRITERIA. THE DELIVERY RATIOS FOR ALL CASES ARE OVER 99.9% DUE TO MAC-LEVEL RETRANSMISSIONS FOR UNICAST MESSAGES.

of 20 packets per second. As in Section IV, we use the IEEE 802.11 MAC protocol, and the MAC-level transmit data rate is fixed at 1 Mbps. Depending on the estimation strategies we combine with the routing metric, we can potentially choose different next hops. For each case, we measure the average delivery ratio and number of total retransmissions (overhead). In some of our experiments, we force the routing code to choose a particular next hop to compare the performance. We use an internal variable (*LongRetryCount*) in the MadWifi device driver to retrieve the total number of MAC-level retransmissions. To maintain consistency with the results in Table I, we use the estimated values in the table as fixed link cost when choosing the next hop. We compare the performance when we use BASIC(128), INDEP(128), and GE(16,128)[6].

In Table II, we present (1) routing metrics for each neighbor from the source node 1 when using different estimation schemes and (2) the number of MAC-level data retransmissions when choosing different nodes as the next hop. Node 1 sends 1000 UDP packets total. For the ADV metric, since we do not consider link quality, we choose node 8, which is closest to the destination node 9. When we use NADV based on BASIC(128), error rates for 1024-byte packets are underestimated, and node 8 is chosen as the best next hop. However, the actual data packet error rate for the link to node 8 is significantly higher (37.5% for 1024-byte packets vs. 18.0% for 128-byte packets), and using node 8 as relay node leads to multiple packet retransmissions due to losses. In contrast, when we use NADV and GE(16,128), we can estimate the actual error rate more accurately. Consequently, we can transfer data messages with minimum data overhead; when using node 7 as relay, we experience 304 retransmissions, which is only around 40% of the case of using node 8 as relay (304 vs. 772). Although not shown here, when using INDEP(128) and NADV, node 7 is still chosen as next hop. However, this selection is based on incorrect PER estimation (25.5% for INDEP(128) estimation vs. 14.5% for observed PER). Such incorrect estimation by INDEP can potentially eliminate the use of links with reasonable quality, which will be often suboptimal. Thus, we expect that INDEP will not work well in other scenarios, and we plan to perform more experiments in various settings.

[6]In the implementation we use, periodic messages use 16-byte UDP packets. If periodic messages include neighbor information such as reverse link quality and location information [32], the size of periodic message will be easily over 128 bytes even with a few neighbors. Therefore, our scheme can be implemented without introducing additional overhead.

Although not comprehensive, we believe the results in this section indicate that our proposed scheme can achieve significant performance improvement in practice.

VI. CONCLUSIONS AND FUTURE WORK

Longer data packets are more prone to bit errors than shorter probe messages, and directly using the statistics from short probe messages as estimated data error rates can lead to significant inaccuracy and inefficiency. In this paper, we have described a simple PER estimation scheme based on the two-state Markov bit error model by Gilbert and Elliot. We perform various experiments on two wireless testbeds and compare the estimation performance of the proposed scheme with that of other schemes. We demonstrate that our proposed scheme can estimate error rates for packets of arbitrary size in various environments, while the independent bit-error model does not lead to accurate estimation. We also have integrated our scheme into an existing geographic routing framework and shown that the combination can improve the network efficiency significantly.

There are many interesting issues we can look into for more accurate PER estimation with lower overhead and delay. Using other available information (e.g., observed loss rates of data packets, SNR values, overheard nearby communications, etc.) will be a promising approach to reduce the number of probe messages while achieving reasonable estimation accuracy. In practice, there can be collision-induced packet errors, and identifying those errors to improve the estimation accuracy will be an interesting issue for future work.

ACKNOWLEDGMENT

The authors thank Emulab and ORBIT staff members for their help during the experiments and the USC Embedded Networks Laboratory for the geographic routing implementation. The authors are grateful to Bobby Bhattacharjee and Arunchandar Vasan for useful discussions.

REFERENCES

[1] D. Johnson and D. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic Publishers, 2001.

[2] C.E. Perkins and E.M. Belding-Royer. Ad hoc on-demand distance vector (AODV) routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

[3] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th ACM/IEEE MobiCom*, pages 243–254. ACM Press, 2000.

[4] Henrik Lundgren, Erik Nordstr, and Christian Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc networks. In *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, pages 49–55, 2002.

[5] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of ACM Mobicom*, pages 134–146. ACM Press, 2003.

[6] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *ACM SIGCOMM*, 2004.

[7] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of ACM MobiCom*, pages 114–128. ACM Press, 2004.

[8] Seungjoon Lee, Bobby Bhattacharjee, and Suman Banerjee. Efficient geographic routing in multihop wireless networks. In *Proceedings of ACM Mobihoc 2005*, May 2005.

[9] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *Proceedings of ACM SenSys*, pages 108–121. ACM Press, 2004.

[10] Abtin Keshavarzin, Elif Uysal-Biyikoglu, Falk Herrmann, and Arati Manjeshwar. Energy-efficient link assessment in wireless sensor networks. In *Proc. of IEEE Infocom*, March 2004.

[11] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, 1960.

[12] E. O. Elliot. Estimates of error rates for codes on burst-noise channels. *Bell Systems Technical Journal*, 42:1977–1997, 1963.

[13] Andreas Kopke, Andreas Willig, and Holger Karl. Chaotic maps as parsimonious bit error models of wireless channels. In *Proceedings of Infocom*, April 2003.

[14] J. Arauz and P. Krishnamurthy. Markov modeling of 802.11 channels. In *Proc. of VTC-Fall*, pages 771–775, October 2003.

[15] J. A. Hartwell and A. O. Fapojuwo. Modeling and characterization of frame loss process in ieee 802.11 wireless local area networks. In *Proc. of VTC-Fall*, pages 4481–4485, September 2004.

[16] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an ieee 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, December 2002.

[17] D. Eckhard and P. Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *Proc. of ACM Sigcomm*, August 1996.

[18] H.-S. So, K. Fall, and J. Walrand. Packet loss behavior in a wireless broadcast sensor networks. Technical report, University of California, Berkeley, 2003.

[19] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of ACM SenSys*, pages 1–13. ACM Press, 2003.

[20] Suman Banerjee and Archan Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *Proceedings of the 3rd ACM MobiHoc*, pages 146–156, 2002.

[21] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR), October 2003. IETF RFC 3626.

[22] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5), 2002.

[23] Hongwei Zhang, Anish Arora, and Prasun Sinha. Learn on the fly: Data-driven link estimation and routing in sensor network backbones. In *25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[24] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 18(3):535–547, August 2000.

[25] Martin Heusse, Franck Rousseau, Romaric Guillier, and Andrzej Duda. Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless lans. In *Proc. of ACM Sigcomm*, August 2005.

[26] A. Willig. A new class of packet- and bit-level models for wireless channels. In *Proc. of IEEE PIMRC*, pages 2434–2440, September 2002.

[27] Christopher C. Tan and Norman C. Beaulieu. On first-order markov modeling for the rayleigh fading channel. *IEEE Transactions on Communications*, 48(12):2032–2040, December 2000.

[28] William Turin and Michele Zorzi. Performance analysis of delay-constrained communications over slow rayleigh fading channels. *IEEE Transactions on Wireless Communications*, 1(10):801–807, October 2002.

[29] Marco Zuniga and Bhaskar Krishnamachari. Analyzing the transitional region in low power wireless links. In *Proceedings of IEEE SECON*, October 2004.

[30] A. Kamerman and L. Monteban. Wavelan-II: A high-performance wireless lan for the unlicensed band. *Bell Lab Technical Journal*, 2(3):118–133, 1997.

[31] John Bicket. Bit-rate selection in wireless networks. Master's thesis, Massachusetts Institute of Technology, feb 2005.

[32] I. Stojmenovic and X. Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, October 2001.