All Bits Are Not Equal – A Study of IEEE 802.11 Communication Bit Errors

Bo Han*, Lusheng Ji†, Seungjoon Lee†, Bobby Bhattacharjee*, and Robert R. Miller†
*Department of Computer Science, University of Maryland, College Park, MD 20742, USA
†AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932, USA

***Abstract*—In IEEE 802.11 Wireless LAN (WLAN) systems, techniques such as acknowledgement, retransmission, and transmission rate adaptation, are frame-level mechanisms designed for combating transmission errors. Recently sub-frame level mechanisms such as frame combining have been proposed by the research community. In this paper, we present results obtained from our bit error study for identifying sub-frame error patterns because we believe that identifiable bit error patterns can potentially introduce new opportunities in channel coding, network coding, forward error correction (FEC), and frame combining mechanisms. We have constructed a number of IEEE 802.11 wireless LAN testbeds and conducted extensive experiments to study the characteristics of bit errors and their location distribution. Conventional wisdom dictates that bit error probability is the result of channel condition and ought to follow corresponding distribution. However our measurement results identify three repeatable bit error patterns that are not induced by channel conditions. We have verified that such error patterns are present in WLAN transmissions in different physical environments and across different wireless LAN hardware platforms. We also discuss our current hypotheses for the reasons behind these bit error probability patterns and how identifying these patterns may help improving WLAN transmission robustness.**

***Index Terms*—Sub-frame bit errors; bit error patterns; measurement study; calibration; IEEE 802.11.**

I. Introduction

In modern digital wireless communication, a transmitter maps data bits into states of information bearers, i.e. frequency, phase, and amplitude, of a sinusoidal electromagnetic wave called the carrier. Each block of data bits is modulated onto a segment of the carrier wave with persistent information bearer states. Such a segment of the carrier is called a *symbol*, which is often represented by a complex number.

As transmitter emitted electromagnetic energy propagates through communication medium and reaches the intended receiver, the receiver demodulates and recovers the original data bits by detecting the states of the information bearers of each symbol. Any factors during transmission, propagation, and reception processes that distort symbols may reduce signal quality, which is often described quantitatively as Signal to Interference and Noise Ratio (SINR), because they make recovery of the original data bits more difficult.

Wireless communication signal attenuates much more rapidly over distance compared to wired communication because it uses open space as its medium. Instead of being contained within the physical boundaries of the wired medium, wireless communication carrier energy is radiated along all directions. Also because of the open nature of wireless communication medium, electromagnetic energy is often reflected, diffracted, and scattered by obstacles in environment. Interference energy introduced by other nearby transmitters and noise energy present in both environment and circuitries of the transmitter and receiver also affect the received signal quality. As a result, wireless communication is often characterized by its relatively low received signal quality and high variance in quality. Consequently not only bit errors occur more often but also the variance in bit error probability in wireless communication systems is high compared to wired communications.

It is well known that wireless channel errors have a significant impact on the performance of various protocols [1], [2], [3]. There are many techniques in wireless communication systems designed to overcome the bit error characteristics. A transmitter may use a modulation scheme with sparse constellation to reduce the probability that a symbol is mistaken with another, which may lead to data bits carried by this symbol being recovered incorrectly. Wireless communication systems may also use error detection and correction coding schemes to help the receiver recovering the original data bits by including redundant information in transmissions. Modern wireless communication systems often support multiple modulation and channel coding schemes for balancing throughput and error correction capability under different channel conditions. Of course all approaches have their limits and their achievable throughputs are still bounded by laws such as the Shannon-Hartley capacity.

Compared to its wired counterparts, WLAN communication has some unique transmission error characteristics. In this paper, we present experiment results obtained from a study focusing on WLAN transmission bit errors. We believe that getting a better understanding of such bit error behaviors can potentially introduce new possibilities for improving WLAN transmission robustness.

For systems such as the IEEE 802.11 WLANs, each physical layer (PHY) frame is a self-contained communication information unit with both PHY control and data information. With a commonly known format, all PHY layer communication parameters are embedded within each frame itself. For instance, there is no side band used for synchronization between a transmitter and a receiver. Such synchronization is achieved by the receiver receiving a special SYNC field, which is a fixed number of symbols of known contents, at the beginning of each frame. Modulation and coding specification

is also embedded in PHY layer frame header. With this design, naturally error recovery techniques are performed at frame level. The acknowledgment based retransmission mechanism is an integrated part the standard. Rate adaptation and other frame recovery schemes have also been proposed to improve packet loss resilience and increase the throughput of wireless networks [2], [4], [5]. More recent proposals have begun to look at using information, i.e. data bits, at sub-frame level [3], [6], [7]. For example, with frame combining, multiple possibly erroneous receptions of a given frame are combined together to recover the original frame without further retransmission. Partly motivated by this trend, we began to study the positions of erroneous bits within frames. We believe that repeatable and predictable patterns are very helpful for designing sub-frame level mechanisms such as frame combining.

For WLAN transmissions, assuming both the transmitter and receiver are stationary, conventional wisdom dictates that the bit errors should be evenly distributed across the entire frame. This is largely due to the expectation that within frame transmission duration the channel condition likely remains unchanged. Markov models with finite states are also popular [8], [9]. In addition, Poisson-distributed bit error model has been used to measure the performance of wireless TCP protocols (e.g., the snoop protocol [10]). Recently, a chaotic map model has been proposed which determines its parameters based on measurement data [11]. There are also measurement studies of error characteristics for in-building wireless networks [12], wireless links in industrial environments [13], and urban mesh networks [14].

In this work, we study WLAN transmission errors on the "sub-frame" level. We have conducted extensive experiments on IEEE 802.11 wireless LAN testbeds. Our measurement results have identified that in addition to channel condition induced bit error distribution, other bit error probability patterns also exist across different communication environments and different hardware platforms. To our best knowledge, this is the first detailed, systematic experiment study of sub-frame bit error characteristics. The contributions of our work are as follows.

- We have performed extensive experiments on IEEE 802.11 wireless LAN testbeds to study sub-frame error characteristics and their location distribution.
- We have identified three patterns for bit error probabilities with respect to bit position in a frame that are not caused by channel fading, namely the slope-line pattern, the saw-line pattern, and the finger pattern.
- We have verified that such characteristics exist in different physical environments and across different wireless LAN hardware platforms.

The rest of this paper is organized as follows. We first give a brief introduction of the IEEE 802.11 modulation and channel coding schemes in Section II. In Section III, we describe our testbed construction and experiment configurations. We report our measurement results in Section IV and discuss hypotheses for the reasons behind the bit error probability patterns in Section V. After reviewing some of related work in Section VI, we conclude with a discussion of future work in Section VII.

Rate (Mbps)	802.11 amendment	Modulation	Coding rate	Data bits / symbol
1	-/DSSS	DBPSK	1	1/11 chips
2	-/DSSS	DQPSK	1	2/11 chips
5.5	b/DSSS	CCK	1	4/8 chips
11	b/DSSS	CCK	1	8/8 chips
6	ag/OFDM	BPSK	1/2	24/OFDM Symbol
9	ag/OFDM	BPSK	3/4	36/OFDM Symbol
12	ag/OFDM	QPSK	1/2	48/OFDM Symbol
18	ag/OFDM	QPSK	3/4	72/OFDM Symbol
24	ag/OFDM	16-QAM	1/2	96/OFDM Symbol
36	ag/OFDM	16-QAM	3/4	144/OFDM Symbol
48	ag/OFDM	64-QAM	2/3	192/OFDM Symbol
54	ag/OFDM	64-QAM	3/4	216/OFDM Symbol

TABLE I
IEEE 802.11 PHY PARAMETERS.

II. IEEE 802.11 WIRELESS LAN BACKGROUND

The IEEE 802.11 standard covers both the Medium Access Control (MAC) and Physical (PHY) layers [15]. For our study, the most important parts of the PHY layer are modulation and channel coding schemes.

The original 802.11 standard defines a Direct Sequence Spread Spectrum (DSSS) system operating in the 2.4 GHz frequency band. A number of amendments have greatly expanded WLAN capability by specifying more modulation and coding schemes and more frequency bands. IEEE 802.11b still uses DSSS but with two additional modulation schemes. Both IEEE 802.11a and 11g are Orthogonal Frequency-Division Multiplexing (OFDM) systems. We summarize the various PHY layer parameters for different variations of the IEEE 802.11 standard in Table I.

The brief description below is specific to OFDM PHYs. More detailed and complete information can be found in [15]. As mentioned before, each 802.11 frame is a self-contained communication unit. It begins with a PHY header of a format that is known by all WLAN receivers. The PHY header consists of a PLCP Preamble and a PLCP Header. The PLCP Preamble contains a number of training symbols, which help receivers detect signal, configure gain control, align frequency, and synchronize timing. Time synchronization enables the receiver to determine the boundaries of each symbol. The PLCP Header specifies the modulation and coding scheme and the length of the frame. Once the PLCP Header is received, the receivers are configured accordingly to demodulate and decode the rest of the frame, which contains PHY layer data.

The data portion of each frame is the result of the PHY encoding process. Data bits received from the MAC layer are first scrambled by XOR-ing the data bits with a scrambling sequence. The scrambler is used to randomize the data bits which may contain long sequence of binary 1s or 0s. The scrambled data bits are then encoded by convolutional code with rate of $1/2$. Higher coding rates are achieved by discarding (puncturing) coded bits at certain positions. The scrambled and coded data bits are subsequently interleaved by a two-step permutation. The interleaver works over blocks

of size corresponding to the number of coded bit per symbol for the specified data rate. The first permutation is used to map adjacent coded bits onto nonadjacent sub-carriers. The second is used to avoid long runs of low reliability bits by mapping adjacent coded bits onto less and more significant bits of the constellation. Finally the scrambled, encoded, and interleaved data bits are divided into groups with each group converted into a complex number according to the specified modulation scheme for each sub-carrier of the OFDM system. Every 48 complex numbers are transformed into one clip of time domain wave form, called an OFDM symbol, by an Inversed Fast Fourier Transformation (IFFT).

III. Experimental Platform

A. Hardware Configuration

We use the same kind of hardware platform for both transmitter nodes and receiver nodes in our experiments. Each of the nodes is a Soekris Engineering net4826 embedded computer with 2 mini-PCI type III sockets for options such as WLAN cards. We primarily use the EMP-8602 and DCMA-82 mini PCI cards in our experiments. Both use Atheros AR5006 802.11 a/b/g chipset.[1] On each node the WLAN card is connected to an omni-directional antenna with 5 dBi (4.8 dBi after cable/connector loss) gain. We use a USB port on each node to dump received frames to an external storage. Each node runs a Debian Linux distribution with kernel version 2.6.15 and its WLAN operation is supported by the MadWifi v0.9.3 device driver.

B. RSSI Calibration

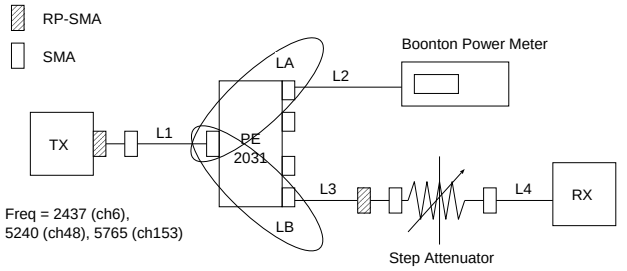

Fig. 1. Calibration setup.

Most WLAN chips report received signal quality using a numerical value called the Received Signal Strength Indicator (RSSI) [16], [17]. Since there is no standard definition for RSSI, device manufacturers may have their own interpretations and thus implement it differently. For the Atheros chipset and MadWifi driver, the RSSI is believed to be a linear scale representation of the actual received signal power in dBm. Because bit errors are highly related to signal quality, we consider signal strength information extremely important. Hence we first calibrated the RSSI values of the WLAN cards used in our experiments with the setup shown in Figure 1. In this setup, a step attenuator is placed between the receiver and the B port of the PE2031 RF signal splitter to produce different received signal power levels.

[1]http://www.atheros.com

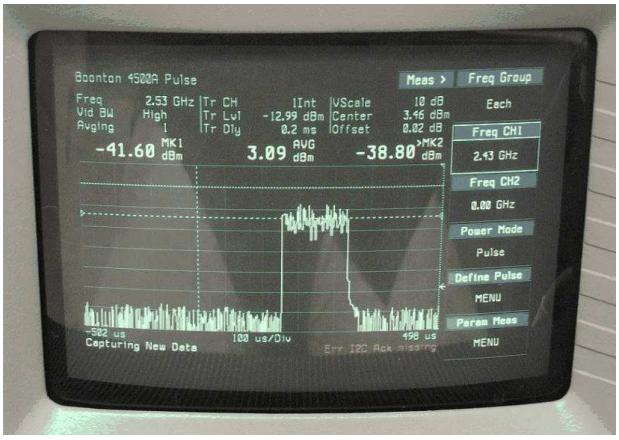

Fig. 2. Boonton 4400 Power Meter Display

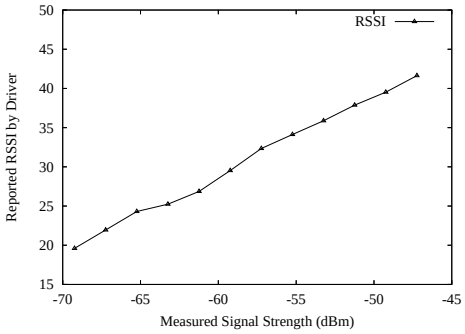

Fig. 3. RSSI to received signal power mapping.

With this setup, after all individual component attenuation is measured, the signal strength at the receiver S_{RX} can be calculated as:

$$S_{RX} = S_{PM} + L2 + LA - LB - L3 - LS - L4$$

where Li is cable i's attenuation, LA and LB are the attenuations of splitter port A and B respectively, LS is the attenuation of the step attenuator, and S_{PM} is the power meter reading. During calibration process, the WLAN transmitter periodically transmits data frames of the same length and contents on channel 6 (2.437 GHz). The transmissions are received by both the power meter and the WLAN receiver. The Boonton[2] 4400 RF Peak Power Meter screen as shown in Figure 2 displays a captured WLAN frame at 54 Mbps transmission rate. The received signal power at the WLAN receiver can then be calculated and compared with the RSSI value reported by the same WLAN card. The step attenuator is used to add series of different attenuations before the signal reaches the splitter, as a way of controlling different received signal power. Figure 3 plots a typical calibration results. Our results indicate that for our WLAN cards, RSSI has linear relationship with the received signal power in dBm.

C. Experiment Procedure

During the experiments, we configure one node to be the transmitter and a number of nodes as receivers. We disable

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

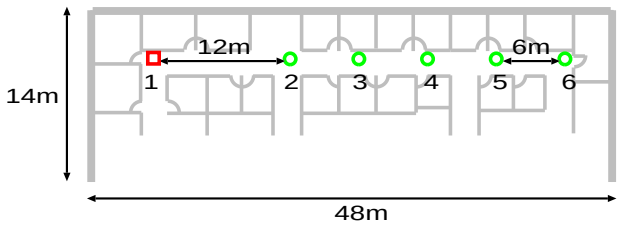

Fig. 4. Primary testbed topology.

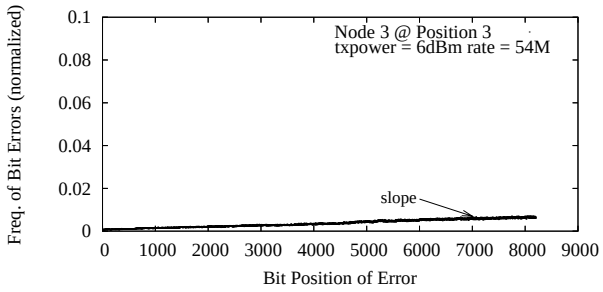

Fig. 5. Normalized bit error frequency for node 3.

antenna diversity on both transmitter and receiver to avoid signal quality variation caused by either end switching to a different antenna port. The transmitter continuously sends 1024-byte long UDP packets every 10 ms. Within each data packet, we reserve the first 4 data bytes as sequence number to match received frames with originally transmitted frames. We put the receivers under "monitor" mode and configure them to pass all data frames received from the transmitter, regardless of error status, to user space. The received frames are compared with the original frames to locate at what bit positions they differ.

It is worth noting that the MAC header and our data sequence number field are not immune to transmission errors, which may cause miss-matching between a transmitted frame and a received frame, or discard/accept frames mistakenly. Such errors are identified in our experiments if possible or otherwise ignored. Because of the relatively small number of bits involved for such errors, the probability of such an event is small.

We mostly use data packets with all data bytes set to 0x00. Because of the scrambling procedure at PHY layer, we do not expect the contents of data packets to have any significant impact on the experiment results. In some experiments, we also used data contents of all bytes set to 0xFF (all 1's), 0x55 (alternating 0's and 1's), or random values. We only study bit errors in UDP payload (not including the first 4-byte sequence number). In each experiment, the transmitter sends out 100,000 identical packets unless stated otherwise.

Our primary testbed consists of 6 nodes linearly deployed along a hallway of an indoor office environment, as illustrated in Figure 4. Node 1 is configured as the transmitter and the rest 5 nodes are receivers. The transmitter and the first receiver is approximately 12 meters apart, and the adjacent receivers 6 meters apart. This particular setup allows us to see how bit errors may happen as the same transmission is received by receivers at increasing distance, or decreasing signal quality, from the transmitter. Limited by physical space constrains, other testbeds often consist of fewer receiver nodes. In these situations we have to reduce transmission power or apply attenuator to achieve the same attenuation that distance can produce. All the experiments on the primary testbed were performed during daytime of weekday with some other nearby 802.11 networks operating on the same channel. The details of these secondary testbeds will be explained along as we discuss the results.

IV. Experiments and Results

As received signal quality decreases, the difficulty for a receiver to receive a frame correctly increases. Loosely speaking, frames that failed to be received correctly fall into one of three categories: frames received with bit errors, truncated frames, and completely lost frames. Frames with bit errors usually occur when the received signal quality is marginal. In this case only some bits within a frame are decoded in error. Although 802.11a/g PHY layer utilizes convolutional coding for error correction, once the number and distribution of erroneous bits exceed coding correction capability, the resultant frame after PHY decoding will contain error bits. Such errors will likely be caught by MAC layer integrity check and cause the frame to be discarded. During the reception of a frame, if the received signal quality drops so much that the receiver could no longer even detect carrier, the PHY layer will prematurely exit from frame reception, which results in a truncated frame. In some cases, a transmitted frame may be completely lost. Examples of conditions causing lost frames include: the receiver could not detect carrier at all, or it could not lock its clock with the synchronization symbols included in the beginning of the frame, or it could not receive and decode preamble and PLCP header of the frame, etc.

We must point out two compromises that we make during the experiments. The first is that since we could only intercept received bits at the top of PHY layer because in commercial WLAN products PHY processing including channel encoding/decoding is concealed within hardware/firmware and not accessible from outside, the bits under study are after-channel-decoding bits not over-the-air bits. The other is that not all experiments are conducted with the same transmission power. Examples of experiments done at different transmission power from the primary testbed include those conducted in small closed environments, or those on testbeds that node distance is irrelevant. In these cases we have to vary transmission power to produce the effects that distance would.

A. Bit Error Distribution Patterns

Our measurement results on the primary testbed have identified several interesting bit error probability patterns.

Figure 5 is a histogram of where the erroneous bits are located for receiver node 3. The x-axis is the bit position within the 1024-byte data packets and the y-axis is the error frequency for each bit position. The y-axis value is normalized over the

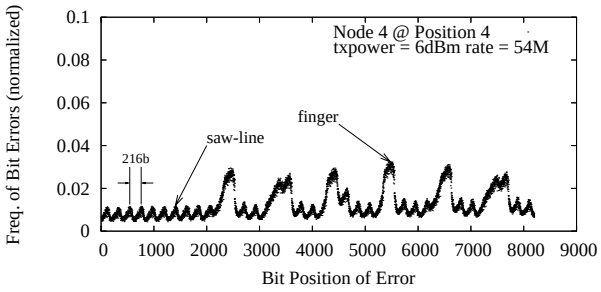

Fig. 6. Normalized bit error frequency for node 4 with data rate 54 Mbps. The average RSSIs of correct packets, truncated packets and packets with bit errors are 36, 21 and 22, respectively.

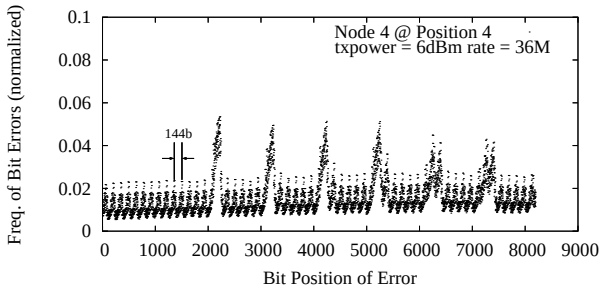

Fig. 7. Normalized bit error frequency for node 4 with data transmission rate 36 Mbps. The average RSSIs of correct packets, truncated packets and packets with bit errors are 34, 19 and 21, respectively.

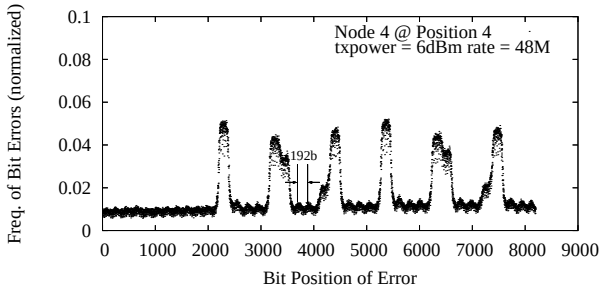

Fig. 8. Normalized bit error frequency for node 4 with data transmission rate 48 Mbps. The average RSSIs of correct packets, truncated packets and packets with bit errors are 35, 22 and 26, respectively.

total number of transmitted packets. In this experiment, we set the transmission power to 6 dBm and bit rate to 54 Mbps. The average RSSIs for correct, truncated and error packets received during this experiment are 37, 28 and 29, respectively. During the experiments, we send out 100,000 packets with all bytes set to 0x00. Among the 100,000 packets, the total number of received packets is 86,119, including 198 truncated packets and 5,238 packets with bit errors. We have only plotted erroneous bits for packets received with bit errors. Figure 5 clearly shows that there exists a linear relationship between the frequency of bit errors and the bit position in the frame. A bit near the end of a frame is more likely to be received in error than a bit near the beginning of the frame. For example, a bit at position 8,000 (0.00656) is about 3 times more likely to be received in error than a bit at position 1,000 (0.00161).

We show the same bit error frequency vs. bit position plot with data collected from receiver node 4 during the same experiment in Figure 6. Since node 4 is farther away from the transmitter than node 3, this plot also exhibits different patterns. While the slope pattern is still present, Figure 6 also displays two additional patterns: what we refer to as the *saw-line* pattern and the *finger* pattern. The saw-line pattern is the fine zig-zag line that goes across the full length of the frame. What is interesting about this pattern is that the saw-tooth peak-to-peak period is about the same as the number of bits each OFDM symbol carries at 54 Mbps transmission rate. The finger pattern refers to the larger peaks, which begins to appear after certain bit position (around 2,000th bit) and repeats at a fairly regular interval. The overall bit error frequency plot in Figure 6 is actually the combination of all three patterns.

Similar patterns can also be observed from results obtained from node 5 and 6. Node 2 is the closest to the transmitter among all receivers. It has the best received signal quality. As a result, we were not able to collect enough frames with erroneous bits to produce any meaningful bit error histogram plots for node 2. Clearly none of these three patterns observed are the result of any known kind of channel fading.

The experiment is repeated with transmission rate set to 36 and 48 Mbps, and with different data contents (all bytes set to 0xFF, 0x55, or random value). Due to space limitation, we only show the plots for 36 and 48 Mbps with all bytes set to 0x00 in Figure 7 and 8. While the same three patterns can be observed from all these plots, the peak-to-peak period of the saw-line pattern changes as transmission rate changes.

B. Quantification of Patterns

In this subsection, we further analyze the three patterns identified above by quantitatively modeling the patterns using curve fitting techniques. As we mentioned above, the error bit frequency results are apparently the combination of slope-line, saw-line and fingers.

We first use a linear function $l(x) = u * x + v$ to fit the slope pattern. Because the fingers have high peaks that would affect the fitting result, we calculate the slope parameters using a modified plot by removing all the data points in the finger regions. We then model the saw-line for the first 2,000 bits. As mentioned before, the fingers only appear after certain point and within the first 2,000 bits there is no finger. The saw-line is modeled by a periodic function

$$s(x) = a + b * cos(\omega * x) + c * sin(\omega * x) + l(x)$$

where $l(x)$ is the bit errors contributed by the slope line at position x.

We summarize the fitting results for the patterns observed at node 4 for 54 Mbps (Figure 6), 48 Mbps (Figure 8), and 36 Mbps (Figure 7) in Table II. For the saw-line fitting, after we determine the value of ω, we can calculate the saw-tooth period as $2 * \pi/\omega$, which is shown in the last column of Table II. The calculated saw-tooth period values have verified our earlier observation that the saw-line period is exactly the symbol length for the corresponding transmission bit rate (216 for 54 Mbps, 192 for 48 Mbps and 144 for 36 Mbps).

Once the bit errors contributed by the slope and saw-line patterns are determined, they can be removed and all

Bit Rate	u	v	ω at 95% confidence	Period
54M	5.1×10^{-7}	7.3×10^{-3}	(0.02906, 0.02917)	215.8
48M	4.5×10^{-7}	8.8×10^{-3}	(0.0325, 0.033)	191.9
36M	6.8×10^{-7}	1.1×10^{-2}	(0.04354, 0.04372)	144.0

TABLE II
THE SLOPE AND INTERCEPT OF THE FITTING LINE, THE FITTING FUNCTION FOR SAW-LINE AND THE CALCULATED SAW-LINE PERIOD.

Bit Rate	54M	48M	36M
Finger 1	648(3x)	775(4.036x)	436(3.028x)
Finger 2	858(3.972x)	768(4x)	436(3.028x)
Finger 3	848(4x)	768(4x)	432(3x)
Finger 4	648(3x)	768(4x)	432(3x)
Finger 5	649(3.005x)	768(4.x)	576(4x)
Finger 6	835(3.87x)	761(3.964x)	576(4x)

TABLE III
FINGER WIDTH

remaining bit errors are considered to be the result of the finger pattern. We present the width of the 6 fingers found in results for node 4 from all experiments in Table III. The numbers in the parentheses are the ratio between the finger width and the corresponding symbol length. This table shows that the widths of the fingers are multiples of corresponding number of data bits per symbol.

C. Different Physical Environments

We have repeated our experiments in two different environments to verify that the three identified patterns are not the result of the specific environment of our primary testbed.

We first tried to eliminate effects of radio interferences in our experiment environment by constructing another testbed using the same nodes as in the primary testbed in a small shielded room located in the AT&T Shannon Lab. The shielded room is a 12' x 12' room with metal floor, ceiling, and walls. It is designed to shield what is in the room from all external radio interferences. The transmitter is located in one corner of the room and the receivers are put in another corner diagonally across the room. We present the result for node 3 in Figure 9. The data transmission rate is 54 Mbps. The total number of packets transmitted is 10,000. The three aforementioned bit error patterns are still easy to observe.

Although the shielded room can separate external interferences, it cannot prevent all environmental effects on over-the-air wireless transmissions. One particular example is reflection. Hence we conducted another group of experiments in a

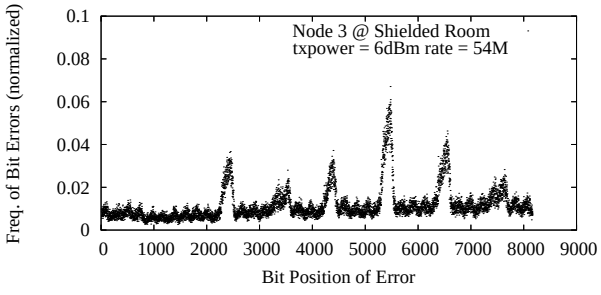

Fig. 9. Normalized bit error frequency for node 3 in shielded room.

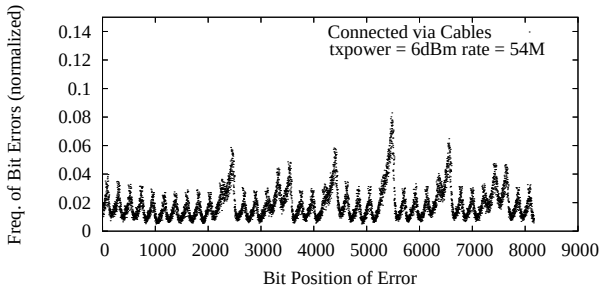

Fig. 10. Normalized bit error frequency for over-cable communication.

Transmitter	Receiver		
	EMP-8602	DCMA-82	Intel PRO 2100
Intel PRO 2915		*	
ZyXEL AG-225H		*	
Conexant 3894		*	*
Agilent E4438C	*		

TABLE IV
EXPERIMENT HARDWARE COMBINATIONS (INDICATED BY *).

laboratory where the transmitter and receiver are directly connected using the same setup as we used for RSSI calibration (Figure 1). The step attenuator is used to gradually reduce the received signal strength. In this group of experiments, the data transmission rate is 54 Mbps and a total of 10,000 packets are transmitted over the directly connected system. Because there is little fluctuation in received signal quality in this case, the transition from very good reception (almost no packets received with bit errors) to very poor (almost no packets received correctly) is very rapid. Figure 10 captures the bit error frequency when the average RSSI for packets with bit error is only 19. Still, the three patterns are identifiable.

Another interesting finding is that there is no truncated packet received when the transmissions are over the cables. This indicates that frame truncations are not likely due to transmitter and receiver hardware issues but likely wireless channel condition fluctuation and interferences.

D. Different Hardware Platforms

The experiment results presented so far were all obtained using WLAN cards made of Atheros chipset. This raises another question: do these patterns only occur on specific hardware platforms? In this subsection we present experiment results obtained using hardware made of different manufactures with different chipsets.

A problem of using non-Atheros chipset WLAN hardware is that the device drivers for those chipsets normally only support a very limited configuration interface. For transmitters, we would need to control transmission bit rate and transmission power. For receivers, we would need to configure the receivers to pass up frames received with bit errors to user space for processing. These requirements, especially the receiver requirement, limited our choices to the combinations of transmitter and receiver hardware as listed in Table IV. Transmitters are shown in the left most column and receivers are shown in the top row.

We show the measurement results when the ZyXEL AG-

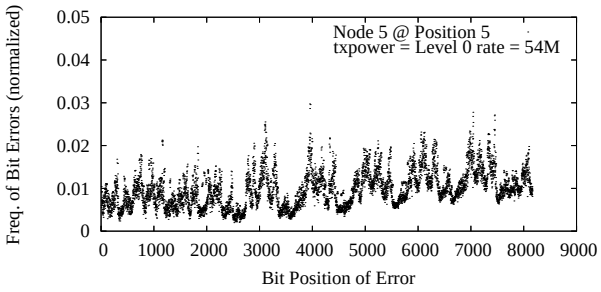

Fig. 11. Normalized bit error frequency for ZyXEL to DCMA.

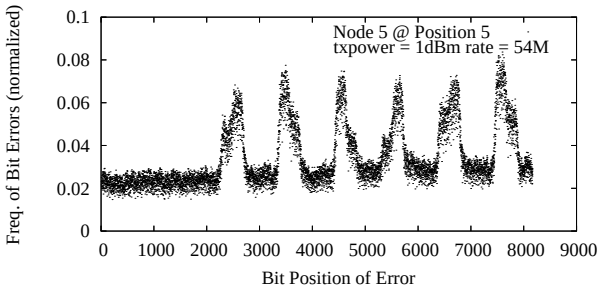

Fig. 12. Normalized bit error frequency for Conexant to DCMA.

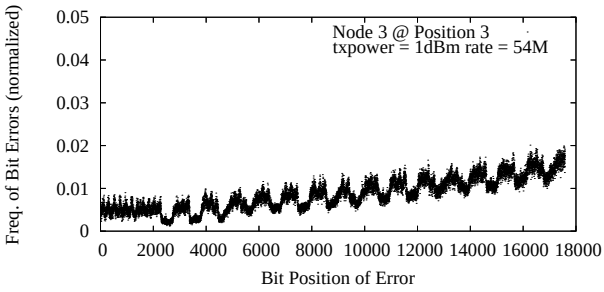

Fig. 14. Normalized bit error frequency for Intel 2915 to DCMA.

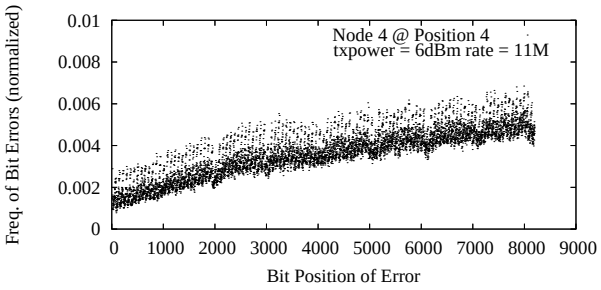

Fig. 15. Normalized bit error frequency for node 4 using IEEE 802.11b.

225H USB Adaptor and Conexant 3894 mini PCI card (also known as the WorldRadio) as transmitters while the DCMA card as receiver in Figure 11 and 12 respectively. In addition to WLAN products, we have also used an Agilent E4438C ESG Vector Signal Generator as the transmitter and connected it directly to an EMP card. This signal generator can create various WLAN waveforms using the Agilent 802.11g WLAN Signal Studio software. We show the measurement results when the transmission power is 5 dBm and bit rate is 54 Mbps in Figure 13. Once again the three patterns are present in all of these plots.

Finally we have used an Intel PRO 2915 mini PCI card as transmitter and a DCMA card as receiver. In this experiment, instead of using 1024-byte packets we used 2200-byte packets to see if the patterns continue as the packet length. The result, as plotted in Figure 14, shows that all three patterns continue all the way till the end of the frames, regardless of the frame length. Another interesting characteristic of this plot is that the fingers are "flipped". Instead of being regions with elevated bit error probability, the fingers here are actually regions with reduced bit error probability.

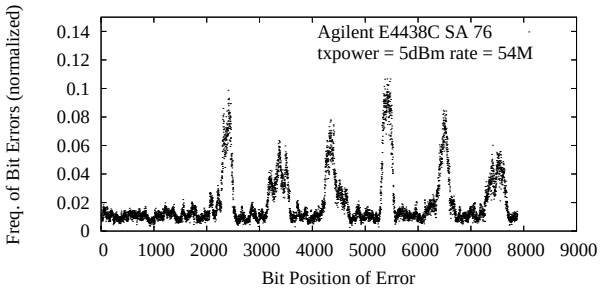

Fig. 13. Normalized bit error frequency for Agilent signal generator to EMP.

E. Different Modulation

Intrigued by the fact that the saw-tooth peak period is exactly at symbol length, we repeated the experiments with 802.11b settings (i.e. 11 Mbps transmission rate). IEEE 802.11b uses DSSS CCK modulation which is quite different from the OFDM modulation used by IEEE 802.11a/g. We show the results in Figure 15. The slope and saw-line patterns are observable in this figure. However, instead of being the number of bits each symbol carries, the saw-line peak-to-peak distance is much larger (e.g. 8 symbol lengths in Figure 15).

F. Summary

During the measurement study on IEEE 802.11 testbeds, we have identified three distinct patterns for bit error probabilities with respect to bit position: slope, saw-line, and finger. We have verified that the presence of the patterns is consistent in different environments and across different hardware platforms.

The slope pattern is the most universal among all. It is present in all experiment results obtained from different environments, with different modulation schemes, and across different hardware platforms. This pattern shows that there is apparently a linear relationship between the chance of bit error occurrence and bit position. Bits near the end of a frame are more likely to be received in error compared to bits in earlier portion of a frame.

The slope pattern may appear alone. However, as signal quality drops further, the other two patterns begin to show. The saw-line is also observable in almost all experiment results. For OFDM transmissions, the saw-tooth peak-to-peak distance is exactly the number of bits carried by each OFDM symbol. For DSSS transmissions, the peak-to-peak distance appears to be multiple of the number of bits carried by each

symbol. The finger pattern has been observed mainly in OFDM transmissions. It may be either in the form of "peaks" or "valleys". The width of the fingers is a multiple of the number of bits carried by each symbol, usually 3-4 symbols.

We have also noted that similar *finger* pattern was also observed and briefly reported in other works, such as from an 802.11b testbed in an industrial environment [13] and an in-building 802.11a testbed [2].

V. Hypotheses and Discussions

While it is extremely difficult to pinpoint the exact causes of the identified patterns without access to detailed WLAN hardware designs, we explore some possible reasons for the slope, saw-line, and finger patterns in this section.

Two apparent reasons for the *slope* pattern are clock drift and changes of channel condition. As mentioned before, the synchronization between receiver and transmitter clocks is done only through receiving special symbols prepended at the very beginning of each frame. Thus, due to synchronization error and clock drifting, as time goes on and bit reception progresses, the offset between the receiver clock and the transmitter clock increases. As a result, the alignment of boundaries of transmitted symbols and receiver samples also becomes worse. This inevitably leads to increased bit error probability.

Moreover, the transmitter only senses the wireless channel prior to transmission. Therefore, during the transmission, some hidden terminals may start their own transmission which will generate some interference. Although this is more likely to cause truncated frames, we cannot rule out this being a reason for later positions having higher bit error probability than earlier positions.

OFDM transmissions' saw-line pattern is likely caused by the frequency selectivity characteristic of wireless channel, the transmitter, and the receiver. Because of this frequency selectivity, certain OFDM sub-carriers may experience higher error rates than other sub-carriers. The interleaver of 802.11a/g is designed to map adjacent data bits to sub-carriers that are far apart from each other. However, because the interleaving permutation is identical for all symbols, frequency selectivity induced bit error pattern will also be repeated for every symbol. This is the reason that the saw-line peak-to-peak distance is exactly the symbol length.

The finger pattern is the most difficult to explain. One possibility is that this pattern is caused by the inter-play between the transmitter's power control loop and the receiver's gain control loop. Further experiments and investigations on the reasons for the finger pattern are part of our future work.

Previously the research community has mainly been focusing on characterizing channel fading, noise, and interference resulted bit errors. However none of these reasons is likely to produce patterns reported here. Most of our current hypotheses point to hardware related reasons. We believe that hardware induced bit errors patterns do exist and play an important role in causing bit errors in WLAN systems.

Despite the uncertainties in the root causes for these bit error patterns, we believe that identifying these patterns alone is beneficial for a number of sub-frame error recovery mechanisms. For instance, knowing the slope bit error pattern, instead of transmitting the same frame for the second time, retransmitting a frame with data bits reordered in reversed order from the original frame may improve loss resilience for retransmission-with-memory techniques. Moreover, in many cases the fingers are where most bit errors occur. For instance, for node 4 in our primary testbed, in some cases (e.g. 48 Mbps transmission bit rate) 17.64% of packets received with bit errors have all their erroneous bits under the fingers. A variable coding scheme that can code bits in the finger regions with rates lower than other regions may potentially reduce the number of packets received with bit errors by a healthy margin.

VI. Related Work

A. Modeling of Bit Errors over Wireless Channel

There are a large number of theoretical models proposed for describing communication bit errors over various wireless channels. Among these models, Finite-State Markov Chain [18] based models are among the most popular. For example, Zorzi et al. investigate the behavior of block errors in data transmission over fading channels [19]. Besides these Markovian models, Kopke et al. 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 [11].

B. Measurement Study of Packet/Bit Errors

Other researchers report measurement results of bit and packet errors in various environments. Using AT&T WaveLAN wireless interfaces, Eckhardt and Steenkiste 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 [12]. Willig et al. present results of bit error measurements obtained using an IEEE 802.11-compliant radio in an industrial environment [13]. The main focus of their work is to simulate wireless transmission errors more accurately. Aguayo et al. analyze the causes of packet loss in a 38-node urban multi-hop 802.11b network [14]. They find that link error rates stay relatively uniform for the majority of links. Reis et al. propose practical, measurement-based models for packet reception and interference in static wireless networks [16]. They use the measured RSS values and packet delivery probability to characterize link quality. They also find that generally packet loss at one receiver does not mean loss elsewhere.

As mentioned above, similar *finger* pattern was also observed previously as reported by [13] and [2]. Neither study was focused on bit error patterns and not as extensive as our study in terms of the diversity of experiment environments and platforms.

Studies of the same subject have also been done within the context of wireless sensor networks. So et al. report

results from a series of experiments designed to investigate loss behavior of broadcast messages in a wireless sensor network [20]. Their main finding is that regardless of indoor or outdoor environment, the loss characteristics of broadcast frames observed from different receivers are highly correlated. Zhao and Govindan report on a systematic medium-scale measurement study of packet delivery performance in dense sensor network deployments under three different environments: an indoor office building, a habitat with moderate foliage, and an open parking lot [21]. They explore the spatial-temporal characteristics of packet losses, quantify the prevalence of "gray communication zones", and indicate significant asymmetry in realistic environments.

VII. Conclusion

In this paper, we have presented experimental results from a study for understanding the sub-frame bit error characteristics of IEEE 802.11 transmissions. We have identified three distinct patterns for sub-frame bit error frequency vs. bit position: slope, saw-line and fingers. We verified that these three patterns exist in different physical environments and across different hardware platforms. We have offered hypotheses for what may cause the bit error patterns.

Lacking detailed knowledge of how the hardware vendors implement their WLAN chips, it is difficult to pinpoint the exact causes of the bit error patterns that we have discovered from our experiments. However, we believe that identifying repeatable and predictable bit error patterns that are caused by hardware, not induced by channel fading, is important in itself because the patterns may provide valuable insights for modeling sub-frame bit errors.

In the future, in addition to further experimentation and investigation with more devices and different environment (e.g. outdoor), we plan to take into account these bit error patterns when designing more efficient dynamic intra-packet encoding schemes and packet retransmission mechanisms. Exploring the bit error behavior in wireless sensor networks, such as IEEE 802.15.4 wireless networks, will be another interesting area of future work.

VIII. Acknowledgement

We thank the anonymous reviewers for their insightful comments. We thank Harry Worstell, Saeed Ghassemzadeh, Aaron Schulman and T. Charles Clancy for their support and valuable inputs. We thank Rittwik Jana, Chonggang Wang and Dongwoon Bai for useful discussions. Bobby Bhattacharjee and Bo Han were supported in part by NSF award NeTS-NBD 0626636. Part of this work was done when Bo Han was a summer intern at AT&T Labs – Research.

References

[1] M. Zorzi and R. R. Rao. Perspectives on the Impact of Error Statistics on Protocols for Wireless Networks. *IEEE Personal Communications*, 6(5):32–40, October 1999.

[2] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *Proceedings of the ACM MOBICOM 2005*, pages 16–30, August-September 2005.

[3] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *Proceedings of the ACM SIGCOMM 2007*, pages 409–420, August 2007.

[4] P. S. Sindhu. Retransmission Error Control with Memory. *IEEE Transactions on Communications*, 25(5):473–479, May 1977.

[5] S. Choi, Y. Choi, and I. Lee. IEEE 802.11 MAC-Level FEC Scheme with Retransmission Combining. *IEEE Transactions on Wireless Communications*, 5(1):203–211, January 2006.

[6] G. Woo, P. Kheradpour, D. Shen, and D. Katabi. Beyond the Bits: Cooperative Packet Recovery Using Physical Layer Information. In *Proceedings of the ACM MOBICOM 2007*, pages 147–158, September 2007.

[7] K. C.-J. Lin, N. Kushman, and D. Katabi. ZipTx: Harnessing Partial Packets in 802.11 Networks. In *Proceedings of the ACM MOBICOM 2008*, September 2008.

[8] E. N. Gilbert. Capacity of a Burst-Noise Channel. *Bell Systems Technical Journal*, 39:1253–1265, September 1960.

[9] E. O. Elliot. Estimates of Error Rates for Codes on Burst-Noise Channels. *Bell Systems Technical Journal*, 42:1977–1997, September 1963.

[10] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proceedings of the ACM MOBICOM 1995*, pages 2–11, November 1995.

[11] A. Kopke, A. Willig, and H. Karl. Chaotic Maps as Parsimonious Bit Error Models of Wireless Channels. In *Proceedings of INFOCOM 2003*, pages 513–523, March-April 2003.

[12] D. Eckhardt and P. Steenkiste. Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network. In *Proceedings of the ACM SIGCOMM 1996*, pages 243–254, August 1996.

[13] 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.

[14] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proceedings of the ACM SIGCOMM 2004*, pages 121–132, August-September 2004.

[15] ANSI/IEEE Std 802.11 1999 Edition (R2003). IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2003.

[16] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-Based Models of Delivery and Interference in Static Wireless Networks. In *Proceedings of the ACM SIGCOMM 2006*, pages 51–62, September 2006.

[17] L. Qiu, Y. Zhang, F. Wang, M.-K. Han, and R. Mahajan. A General Model of Wireless Interference. In *Proceedings of the ACM MOBICOM 2007*, pages 171–182, September 2007.

[18] C. C. Tan and N. C. Beaulieu. On First-Order Markov Modeling for the Rayleigh Fading Channel. *IEEE Transactions on Communications*, 48(12):2032–2040, December 2000.

[19] M. Zorzi, R. R. Rao, and L. B. Milstein. Error Statistics in Data Transmission over Fading Channels. *IEEE Transactions on Communications*, 46(11):1468–1477, November 1998.

[20] H.-S. W. So, K. Fall, and J. Walrand. Packet Loss Behavior in a Wireless Broadcast Sensor Networks. Technical report, University of California, Berkeley, 2003.

[21] J. Zhao and R. Govindan. Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. In *Proceedings of SenSys 2003*, pages 1–13, November 2003.