Interference of Bluetooth and IEEE 802.11: Simulation Modeling and Performance Evaluation

N. Golmie, R. E. Van Dyck, and A. Soltanian
National Institute of Standards and Technology
Gaithersburg, Maryland 20899
Email: nada.golmie@nist.gov, robert.vandyck@nist.gov

Abstract— **The emergence of several radio technologies such as Bluetooth, and IEEE 802.11 operating in the 2.4 GHz unlicensed ISM frequency band may lead to signal interference and result in significant performance degradation when devices are co-located in the same environment. The main goal of this paper is to present a simulation environment for modeling interference based on detailed MAC and PHY models. This framework is then used to evaluate the impact of interference on the performance of Bluetooth and IEEE 802.11. We use several simulation scenarios and measure performance in terms of packet loss, residual number of errors, and access delay.**

Keywords—**WPANs, Bluetooth, IEEE 802.11, Interference.**

I. Introduction

THE proliferation of mobile computing devices including laptops, personal digital assistants (PDAs), and wearable computers has created a demand for wireless personal area networks (WPANs). WPANs allow closely located devices to share information and resources. A key challenge in the design of WPANs is, perhaps, the adaptivity to a hostile radio environment that includes noise, time-varying channels, and abundant electromagnetic interference. Today, most radio technologies considered by WPANs (Bluetooth Special Interest Group [1], HomeRF [2], and IEEE 802.15) employ the 2.4 GHz ISM frequency band. In addition, both WPANs and Wireless Local Area Networks (WLANs) devices implementing the IEEE 802.11 standard specifications [3] will be sharing the same frequency band. It is anticipated that some interference will result from all these technologies operating in the same environment. WLAN devices operating in proximity to WPAN devices may significantly impact the performance of WPAN and vice versa.

The main goal of this paper is to present a tool for modeling the interference of Bluetooth and IEEE 802.11. In addition, we discuss our findings on the performance of these systems when operating in close proximity to each other. Our results are based on detailed models for the MAC, PHY, and wireless channel. Recently, a number of research activities has led to the development of tools for wireless network simulation [4][5]. While some of these tools may include a PHY layer implementation, it is often abstracted to a discrete channel model that does not implement interference per se. Therefore, in order to model interference and capture the time and frequency collisions, we chose to implement an integrated MAC-PHY module.

Previous performance results on Bluetooth and IEEE 802.11 interference include experimental measurements obtained by Kamerman [6]. Furthermore, the probability of an 802.11 packet error in the presence of a Bluetooth piconet has been derived by Ennis [7], then extended by Zyren [8] and Shellhammer [9]. Zurbes *et al.* [10] present simulation results for a number of Bluetooth devices located in a single large room. They show that for 100 concurrent web sessions, performance is degraded by only five percent. Chiasserini and Rao [11] evaluate the effect of Bluetooth voice and data links on the performance of an 802.11 system; they apply traffic shaping techniques to reduce the interference. Conversely, Golmie and Mouveaux [12] study the effect of 802.11 on Bluetooth, using a probability analysis approach and validate the analysis with simulation results. They show that significant packet loss can occur and that access delays for data traffic double. Moreover, the number of residual errors in accepted voice packets can be quite high. Similar results have been obtained by Lansford *et. al.* [13] who use simulation and experimental measurements to quantify the interference resulting from Bluetooth and IEEE 802.11. Their simulation models are based on a link budget analysis and a Q function calculation for the channel and PHY models respectively, in addition to the MAC layer behavior.

This paper is organized as follows. In section II, we give some general insights on the Bluetooth and IEEE 802.11 protocol operation. In section III, we describe in great detail our modeling approach for the MAC, PHY and wireless channel. In section IV, we discuss the accuracy of our model implementation. In section V, we evaluate the impact of interference on both Bluetooth and WLAN performance and present simulation results. Concluding remarks are offered in section VI.

II. Protocol Overview

A. Bluetooth

In this section, we give a brief overview of the Bluetooth technology [1] and discuss the main functionality of its protocol specifications, which consist of several modules, namely, the Radio Frequency (RF), Baseband (BB) and Link Manager (LM). Bluetooth is a short range (0 m - 10 m) wireless link technology aimed at replacing non-interoperable proprietary cables that connect phones, laptops, PDAs and other portable devices together. Bluetooth operates in the ISM frequency band starting at 2.402 GHz and ending at 2.483 GHz in the USA and Europe. 79 RF channels of 1 MHz width are defined. The air interface is based on an antenna power of 1 mW with an antenna gain of 0 dB. The signal is modulated using binary Gaussian Frequency Shift Keying (GFSK). The raw data rate is defined at 1 Mbits/s. A Time Division Multiplexing (TDM) technique divides the channel into 625 μs slots. Transmission occurs in packets that occupy an odd number of slots (up to 5). Each packet is transmitted on a different hop frequency with a maximum frequency hopping rate of 1600 hops/s.

Two or more units communicating on the same channel form a piconet, where one unit operates as a master and the others (a maximum of seven active at the same time) act as slaves. A channel is defined as a unique pseudo-random frequency hopping sequence derived from the master device's 48-bit address and its Bluetooth clock value. Slaves in the piconet synchronize their timing and frequency hopping to the master upon connection establishment. In the connection mode, the master controls the access to the channel using a polling scheme where master and slave transmissions alternate. A slave packet always follows a master packet transmission as illustrated in Figure 1, which depicts the master's view of the slotted TX/RX channel.

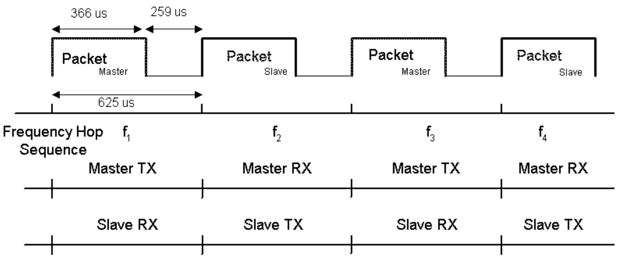

Fig. 1. Master TX/RX Hopping Sequence

There are two types of link connections that can be established between a master and a slave: the Synchronous Connection-Oriented (SCO), and the Asynchronous Connection-Less (ACL) link. The SCO link is a symmetric point-to-point connection between a master and a slave where the master sends an SCO packet in one TX slot at regular time intervals, defined by T_{SCO} time slots. The slave responds with an SCO packet in the next TX opportunity. T_{SCO} is set to either 2, 4 or 6 time slots for $HV1$, $HV2$, or $HV3$ packet formats, respectively. All three formats of SCO packets are defined to carry 64 Kbits/s of voice traffic and are never retransmitted in case of packet loss or error. The ACL link is an asymmetric point-to-point connection between a master and active slaves in the piconet. Several packet formats are defined for ACL, namely $DM1$, $DM3$, and $DM5$ packets that occupy 1, 3, and 5 time slots, respectively. An Automatic Repeat Request (ARQ) procedure is applied to ACL packets where packets are retransmitted in case of loss until a positive acknowledgement (ACK) is received at the source. The ACK is piggy-backed in the header of the returned packet where an ARQN bit is set to either 1 or 0 depending on whether the previous packet was successfully received or not. In addition, a sequence number (SEQN) bit is used in the packet header in order to provide a sequential ordering of data packets in a stream and filter out retransmissions at the destination. Forward Error Correction (FEC) is used on some SCO and ACL packets in order to correct errors and reduce the number of ACL retransmissions.

B. IEEE 802.11

The IEEE 802.11 standard [3] defines both the physical (PHY) and medium access control (MAC) layer protocols for WLANs. In this sequel, we shall be using WLAN and 802.11 interchangeably.

The IEEE 802.11 standard calls for three different PHY specifications: frequency hopping (FH) spread spectrum, direct sequence (DS) spread spectrum, and infrared (IR). The transmit power for DS and FH devices is defined at a maximum of 1 W and the receiver sensitivity is set to -80 dBmW. Antenna gain is limited to 6 dB maximum. In this work, we focus on the 802.11b specification (DS spread spectrum) since it is in the same frequency band as Bluetooth and the most commonly deployed.

The basic data rate for the DS system is 1 Mbits/s encoded with differential binary phase shift keying (DBPSK). Similarly, a 2 Mbits/s rate is provided using differential quadrature phase shift keying (DQPSK) at the same chip rate. Higher rates of 5.5 and 11 Mbits/s are also available using techniques combining quadrature phase shift keying and complementary code keying (CCK); all of these systems use 22 MHz channels. Details of the modulation methods are provided in Section III.

The IEEE 802.11 MAC layer specifications, common to all PHYs and data rates, coordinate the communication between stations and control the behavior of users who want to access the network. The Distributed Coordination Function (DCF), which describes the default MAC protocol operation, is based on a scheme known as carrier-sense, multiple access, collision avoidance (CSMA/CA). Both the MAC and PHY layers cooperate in order to implement collision avoidance procedures. The PHY layer samples the received energy over the medium transmitting data and uses a clear channel assessment (CCA) algorithm to determine if the channel is clear. This is accomplished by measuring the RF energy at the antenna and determining the strength of the received signal commonly known as RSSI, or received signal strength indicator. In addition, carrier sense can be used to determine if the channel is available. This technique is more selective since it verifies that the signal is the same carrier type as 802.11 transmitters. A virtual carrier sense mechanism is also provided at the MAC layer. It uses the request-to-send (RTS) and clear-to-send (CTS) message exchange to make predictions of future traffic on the medium and updates the network allocation vector (NAV) available in stations. Communication is established when one of the wireless nodes sends a short RTS frame. The receiving station issues a CTS frame that echoes the sender's address. If the CTS frame is not received, it is assumed that a collision occurred and the RTS process starts over. Regardless of whether the virtual carrier sense routine is used or not, the MAC is required to implement a basic access procedure (depicted in Figure 2) as follows. If a station has data to send, it waits for the channel to be idle through the use of the CSMA/CA algorithm. If the medium is sensed idle for a period greater than a DCF interframe space (DIFS), the station goes into a backoff procedure before it sends its frame. Upon the successful reception of a frame, the destination station returns an ACK frame after a Short interframe space (SIFS). The backoff window is based on a random value uniformly distributed in the interval $[CW_{min}, CW_{max}]$, where CW_{min} and CW_{max} represents the Contention Window parameters. If the medium is determined busy at any time during the backoff slot, the backoff procedure is suspended. It is resumed after the medium has been idle for the duration of the DIFS period. If an ACK is not received within an ACK timeout interval, the station assumes that either the data frame or the ACK was lost and needs to retransmit its data frame by repeating the basic access procedure.

III. Integrated Simulation Model

In this section, we describe the methodology and tools used to conduct the performance evaluation. The simulation environment consists of detailed models for the RF channel, the PHY, and MAC layers developed in C and OPNET (for the MAC layer). These detailed simulation models will constitute an evaluation framework that is critical to studying the various intricate effects between the MAC and PHY layers. Although interference is typically associated with the RF channel modeling and measured at the PHY layer, it can significantly impact the performance of higher layer applications including the MAC layer. Similarly, changes in the behavior of the MAC layer protocol and the associated data traffic distribution could play an important factor in the interference scenario and affect the overall system performance.

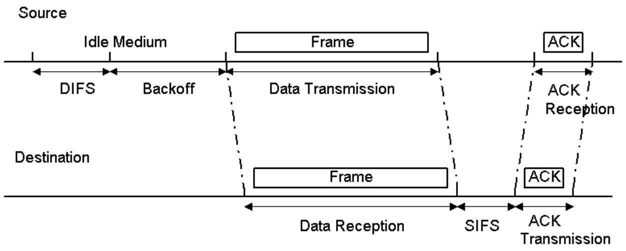

Fig. 2. WLAN Frame Transmission Scheme

A. Channel Model

The channel model consists of a geometry-based propagation model for the signals, as well as a noise model. For the indoor channel, we apply a propagation model consisting of two parts: 1.) line-of-sight propagation (free-space) for the first 8 meters, and 2.) a propagation exponent of 3.3 for distances over 8 meters. Consequently, the path loss in dB is given by

$$L_p = \begin{cases} 32.45 + 20\log(f \cdot d) & \text{if } d < 8 \text{ m} \\ 58.3 \quad + 33\log(d/8) & \text{otherwise,} \end{cases} \tag{1}$$

where f is the frequency in GHz, and d is the distance in meters. This model is similar to the one used by Kamerman [6]. Assuming unit gain for the transmitter and receiver antennas and ignoring additional losses, the received power in dBmW is

$$P_R = P_T - L_p, \tag{2}$$

where P_T is the transmitted power also in dBmW. Eq. (2) is used for calculating the power received at a given point due to either a Bluetooth or an 802.11 transmitter, since this equation does not depend on the modulation method.

Additive white Gaussian noise (AWGN) is used to model the noise at the receivers. This choice was made for two reasons. Firstly, the transmitted powers are large enough and the distances short enough so that the signal-to-noise ratios (SNRs) at the receivers are high. In the absence of interference, the average bit error rate is quite low, even if a flat Rayleigh fading channel is used. Secondly, at a data rate of 1 Mb/s, multipath effects can be neglected. That is, for a typical room, the direct signal and the reflected signals arrive at the receiver at essentially the same time. So while there may be fading, it is relatively frequency non-selective, *i.e.* flat. While this last assumption is less true for the 11 Mb/s 802.11 data rate, it is still fairly realistic for a number of channels.

The transmitters, channel, and receivers are implemented at complex baseband. For a given transmitter, inphase and quadrature samples are generated at a sampling rate of 44×10^6 per second. This rate provides four samples/symbol for the 11 Mb/s 802.11 mode, enough to implement a good receiver. It is also high enough to allow digital modulation of the Bluetooth signal to account for its frequency hopping. Specifically, since the Bluetooth signal is approximately 1 MHz wide, it can be modulated up to almost 22 MHz, which is more than enough to cover the 11 MHz bandwidth (one-sided) of the 802.11 signal. The received complex samples from both the desired transmitter and the interferer(s) are added together at the receiver.

To complete the channel model, the noise must be added to the received samples. Consider a fixed transmitter power and no interference. Then, Eqs (1) and (2) allow one to compute the received signal power for a given distance. The SNR is calculated in dB according to

$$SNR = P_R - S_R, \tag{3}$$

where S_R is the receiver's sensitivity in dBmW. In an actual receiver, the sensitivity is determined primarily by the amount of thermal noise in the electronics; within limits imposed by physics, a better design can lead to a higher sensitivity. For our modeling purposes, the situation is somewhat reversed. One assumes a specific (achievable) sensitivity and uses Eq. (3) to compute the SNR. This quantity is used to set the variance of the random number generator that provides the AWGN noise for each inphase and quadrature sample. Please note that the transmitter and interferer powers can be changed on a packet by packet basis.

A few comments should be made about the relationship among the received signal power, the received interference power, the noise power, and the resulting performance. Analogously to SNR, one can define the signal-to-interference ratio (SIR) in dB as

$$SIR = P_R - P_I, \tag{4}$$

where P_I is the interference power at the receiver. In the absence of interference, the bit error rate (BER) for either the Bluetooth or WLAN system is almost negligible for the transmitter powers and ranges under consideration. In other words, the SNR is high enough so that the BER is less than 10^{-5}. When there is interference from the other system, this factor is what limits performance; the SIR is insufficient to provide an acceptable BER. While the bit error rate is a good measure of the PHY layer performance, we use packet error rate as a metric in the system level performance. The latter depends not only on the SIR, but also on the time and frequency overlap and the traffic distribution of the Bluetooth and WLAN packets. Further discussion is given in Section III-C.

B. PHY Model

The PHY layer includes detailed models of the signal processing in the Bluetooth and the 802.11 transmitters and receivers. As mentioned before, complex baseband implementations are used.

Bluetooth The GFSK modulation used in the Bluetooth system is a type of binary partial response continuous phase modulation. It is a slight generalization of the GMSK modulation [14] used in the GSM cellular system, which uses a modulation index of 0.5; instead, a modulation index of approximately 0.3 is used in Bluetooth. Because of the Gaussian-shaped filter in the transmitter, every data bit is transmitted over two symbol intervals, causing intersymbol interference but reducing the required bandwidth. The information carrying phase is denoted by $\theta(t, \vec{\alpha})$, where t designates time, and $\vec{\alpha}$ represents the data bit vector. The cosine and sine of $\theta(t, \vec{\alpha})$, sampled 44 times per data bit (symbol), give the inphase and quadrature samples.

While there are a number of possible receiver designs, we chose to implement the noncoherent limiter-discriminator (LD) receiver [15] [16]. Its simplicity and relatively low cost should make it the most common type for many consumer applications. The LD receiver consists of four parts: 1.) an intermediate frequency (IF) filter, 2.) the limiter-discriminator, 3.) an integrate and dump filter, and 4.) the detector. The IF filter is responsible for removing noise and interference that are not in the same frequency band as the desired signal. A bandwidth of 1 MHz is used since this provides a good compromise between noise and interference rejection. The limiter-discriminator essentially takes the derivative of the noisy, filtered IF signal. The derivatives are approximated by using finite impulse response (FIR) filters on the inphase and quadrature samples. The integrate and dump filter, as its name implies, integrates over one bit period to obtain the phase. A decision is then made for this bit. The process continues for the entire Bluetooth packet.

802.11b The 1 Mb/s [1] 802.11b system transmits data using differential binary phase shift keying. With DBPSK modulation, the information is conveyed by the phase difference between adjacent transmitted symbols. Thus, it is not necessary to have a coherent phase reference in the receiver. To provide some interference protection, the modulated signal is spread using a Barker sequence with code length equal to eleven [3]. That is, each bit duration is divided into eleven consecutive segments called chips. During each chip, the transmitted signal is multiplied by either ± 1, depending on the code [17]. Because the chip rate is 11×10^6 per second, the two-sided bandwidth of this signal is approximately 22 MHz. After spreading, the signal is fed into a pulse-shaping filter that provides further control on the spectral shape. A couple of comments are in order. Firstly, the use of spread spectrum processing does not change the receiver's performance in an AWGN channel. However, it does have a dramatic effect on the performance in an interference-limited environment. Similarly, changing design parameters, such as the rolloff factor in the pulse-shaping filter, may have a significant impact on performance in interference-limited conditions. We chose a value that minimizes the impact of interference. In the receiver, a matched filter is first used on the noisy, possibly corrupted samples. Next, the signal is multiplied by the spreading sequence and then summed. Finally, a binary decision is made on each bit.

To achieve 11 Mb/s in an environment with fading and interference, a more sophisticated modulation scheme is required if the bandwidth is to be kept constant. This is done using a type of coded modulation. The basic idea is that uncoded quadrature phase shift keying provides two bits per symbol. If the symbol rate is kept constant at 11×10^6 per second then a maximum data rate of 22 Mb/s is possible. However, half of these bits are used to provide a coding gain using complementary code keying (CCK) [18]. Essentially, each group of eight information bits chooses a sequence of eight consecutive symbols that forms a codeword. As before, the inphase and quadrature components of these symbols are transmitted. The receiver looks at the group of eight symbols and decides which was the most likely transmitted codeword. While one can implement this decoding procedure by correlating against all 256 possible codewords, we chose a slightly sub-optimal, but considerably faster architecture similar to the Walsh-Hadamard transform. Because it is sending more data in the same bandwidth, the CCK system is less robust to interference than the 1 Mb/s DSS modulation. However, this is compensated, at least to some extent, by the fact that the data packets are often shorter. Further discussion of these points is given in the experimental results section.

C. *MAC Model*

We used *OPNET* to develop a simulation model for the Bluetooth and IEEE 802.11 protocols. For the IEEE 802.11 protocol, we used the model available in the OPNET library. For Bluetooth, we partially implemented the Baseband and L2CAP layers according to the specifications [1]. We assume that a connection is already established between the master and the slave and that the synchronization process is complete. The connection type is either SCO for voice or ACL for data traffic.

A MAC protocol generally consists of a collection of components, each performing a special function, such as the support of higher layer traffic, the synchronization process, the bandwidth allocation, and contention resolution mechanism. In this sequel, we highlight the features that are the most relevant to our work on interference, namely, we give a brief description of the MAC state machine, the frequency hopping, the interface to the physical layer, and the error detection and correction schemes.

MAC State Machine Each of the Bluetooth and IEEE 802.11 MAC protocols is implemented as a state machine that consists of a number of states. Transitions from one state to another are generally triggered by the occurrence of events such as the reception or transmission of packets. Higher layer message arrivals require packet encapsulation and often segmentation if the message is too long. The information available in the packet determines the type of packet processing and encapsulation required. For example, Bluetooth ACL connections require L2CAP encapsulation in addition to baseband encapsulation, while SCO connections only require baseband encapsulation. The packet is then enqueued and awaits a transmission opportunity. Since SCO packets need to be transmitted at fixed intervals, Bluetooth SCO packets have priority over Bluetooth ACL packets.

Transmission of packets follow each protocol's rules. Bluetooth transmission is based on a polling mechanism where the master controls the usage of the medium including its own transmission. In order to model the slotted nature of the channel, where a master device has to start its transmission in an odd numbered slot, while an even numbered slot is reserved for a slave transmission start, a virtual clock is implemented that generates self-interrupts every 625 μs and triggers the start of a new time slot. At this point, the device is set in either a transmit or receive

[1] The symbol rate is the same as the bit rate, since this is a binary modulation scheme.

mode depending on whether it is a master or slave. On the other hand, the IEEE 802.11 protocol uses a CSMA/CA that allows a station to access the medium if the station is not waiting for an acknowledgement from a previous packet or receiving a packet, after the medium has been idle for a period of time. The station cannot start its transmission before these conditions are met.

Frequency Hopping Frequency usage constitutes another major component of the protocol model. Bluetooth uses a frequency hopping mechanism that sweeps 79 channels of the frequency band available at a maximum rate of 1600 hops/s depending on the packet size. Both master and slave devices are synchronized and follow the same random frequency hopping sequence. This frequency sequence is derived at the master and slave devices and depends on the master's clock and its Bluetooth address. The algorithm for generating the sequence works as follows. Given a window of 32 contiguous frequencies in the 2.4-2.479 GHz range, a sequence of 32 frequencies is chosen randomly. Once all 32 frequencies in that set have been visited once, a new window of 32 frequencies is selected. This new window includes 16 of the frequencies previously visited and 16 new frequencies. For the IEEE 802.11, we focus in this study on the IEEE 802.11 Direct Sequence mode which uses a fixed frequency that occupies 22 MHz of the frequency band. The center frequency is selected among 11 available channels.

Error Detection and Correction Error detection and correction is an essential component in the interference study. For IEEE 802.11, errors are detected by checking the Frame Check Sequence (FCS) that is appended to the packet payload. In case an error is found, the packet is dropped and is then later retransmitted. Otherwise, a positive ACK notifies the source of a correct reception. For Bluetooth, the device first applies the error correction algorithm corresponding to the packet encapsulation used. The encapsulation of voice packets such as $HV1$ and $DM5$ is shown in Figure 3. $HV1$ packets have a total size packet length of 366 bits including a header and an access code of 126 bits. $HV1$ packets use a payload of 80 information bits, a 1/3 FEC rate and are sent every $T_{SCO} = 2$ or 1250 μs. In case of an error occurrence in the payload, the packet is never dropped. A 1/3 FEC is applied to the packet header while a Hamming code ($d = 14$) is applied to the access code. Uncorrected errors in the header and access code lead to a packet drop. In addition, errors in the payload are corrected using a 1/3 FEC rate.

Table I summarizes the error occurrences in the packet and the actions taken by the protocol.

On the other hand, $DM5$ packets use a 2/3 rate FEC to correct payload errors as shown in Figure 3. Errors in the header or access code are corrected by a 1/3 FEC and a Hamming code, respectively. Uncorrected errors lead to dropping packets and the application of the ARQ and SEQN schemes.

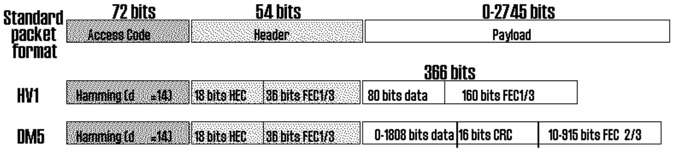

Fig. 3. Bluetooth Packet Format

TABLE I
SUMMARY OF ERROR OCCURRENCES IN THE PACKET AND ACTIONS TAKEN IN CASE ERRORS ARE NOT CORRECTED

Error Location	Error Correction	Action Taken
Access Code	Hamming Code, $d = 14$	Packet is dropped
Packet Header	1/3 FEC	Packet is dropped
HV1 payload	1/3 FEC	Packet is accepted
DM5 payload	2/3 FEC	Packet is dropped

Statistics Collection At the MAC layer, a set of performance metrics are defined to include access delay, probability of packet loss, and residual number of errors in the Bluetooth voice packets. The access delay measures the time it takes to transmit a packet from the time it is passed to the MAC layer until it is successfully received at the destination. The access delay for the Bluetooth LAN traffic is measured at the L2CAP layer in order to account for retransmission delays. Packet loss measures the number of packets discarded at the MAC layer due to errors in the bit stream. This measure is calculated after performing error correction. The residual number of errors in the Bluetooth voice packets measures the number of errors that remain in the packet payload after error correction is performed.

D. MAC Layer to PHY Layer Interface

The OPNET MAC models were interfaced to the physical layer models described in the previous section in order to simulate the overall system. The step-by-step simulation process works as follows. Traffic is generated by sources located above the MAC layer. The message is then passed to the MAC layer where it undergoes encapsulation and obeys the MAC transmission rules. The packet is then sent to an interface module before it is passed to the PHY layer.

This interface module is required to capture all changes in the channel state (mainly in the energy level). Consider the Bluetooth transmitter-channel-receiver chain of processes. For a given packet, the transmitter creates a set of signal samples that are corrupted by the channel and input

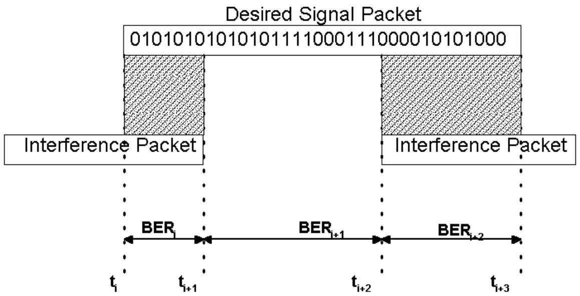

Fig. 4. Packet Collision and Placement of Errors

to the receiver; interference may be present for all or only specific segments of the packet, as shown in Figure 4. A similar chain of processing occurs for an 802.11b packet. The interface module is designed to process a packet at a time.

At the end of each packet transmission, the MAC layer generates a data structure that contains all the information required to process the packet. This structure includes a list of all the interfering packets with their respective duration, timing offset, frequency, and transmitted power. The topology of the scenario is also included. The data structure is then passed to the physical layer along with a stream of bits representing the packet being transmitted. The physical layer returns the bit stream after placing the errors resulting from the interference.

One of the main advantages of this interface is that it allows one to change any of the transmitter, channel, or receiver models. For example, the Bluetooth receiver used in this paper is the simple limiter-discriminator. To study how much a sophisticated Viterbi receiver would improve performance, one only needs to replace a single physical layer module and recompile the OPNET simulation. All the other code, especially the OPNET MAC layer models, remains unchanged. Other experiments have been done using a flat Rician fading model instead of the AWGN one.

DSP vs. Table Implementation

In order to speed up the simulation process, we replace each transmitter-channel-receiver process with a table-based approach combined with a binary symmetric channel. BER tables for different values of SIR and for different frequency offsets were derived. For a segment of a packet where the interference is stationary, the SNR and SIR are computed using the transmitters' powers, the topology, and the path loss model. Thus, using the calculated SIR and the given frequency offset of the intended signal with respect to the interference signal, the average BER can be extracted by a simple table lookup operation. Errors are then generated for each bit of the packet segment using the binary symmetric channel with crossover probability equal to the average BER of the segment. The SNR in these tables is assumed to be very high (greater than 30 dB), which is the case for interference-limited environments. Still, the software can check this assumption by comparing the SIR to this value.

Using tabulated BER values, as opposed to running the detailed signal processing receiver and channel simulation models in real-time, gives a speed up factor of about 120; a combined MAC and signal processing simulation takes approximately 240 minutes to transmit about 48000 Bluetooth packets and 9500 WLAN packets, while a MAC simulation using tabulated BER values takes about 2 minutes for the same number of Bluetooth and WLAN packets. The main question is the accuracy of this approach; this topic is discussed in Section IV-B.

IV. Simulation Model Validation

A. *Results Accuracy*

Since the implementation of the PHY layer required choosing a number of design parameters, the first step in the validation process is comparing the PHY results against theoretical results. Complete BER curves of the Bluetooth and 802.11b systems are given in [19]; for the AWGN and flat Rician channels without interference, all the results match very closely to analytical bounds and other simulation results (*cf.* refences in [19]). This paper also discusses the design choices in greater detail. Moreover, the literature on jamming in military systems [20] provides good bounds on the physical layer performance. Basically, Bluetooth can be considered an on-off tone jammer for 802.11b, while the latter can be modeled as a broadband noise jammer for Bluetooth. Next, the simulation results for both the MAC and PHY models were compared and validated against analytical results for packet loss given different traffic scenarios [12].

B. *Table Implementation Accuracy*

Figure 5 gives the BER in terms of the SNR for varying SIR and for co-channel interference. To create the table, the curves are sampled every 0.5 dB in both SNR and SIR. A couple of points need to be made: (1) For a fixed level of SIR, one notices that the change in BER for a 0.5 dB step in SNR is quite small, even at low SNR. For example, a change in BER from 0.25 to 0.2 is not particularly important, since it is still so high that a packet will most likely be lost. (2) For a fixed SNR, a 0.5 step in SIR also gives a small relative change in BER, especially for SIRs below 2 dB. As the SIR goes above 2 dB, the BER drops below 10^{-2}, and the system performance becomes increasingly good. For the overall system performance, it does not re-

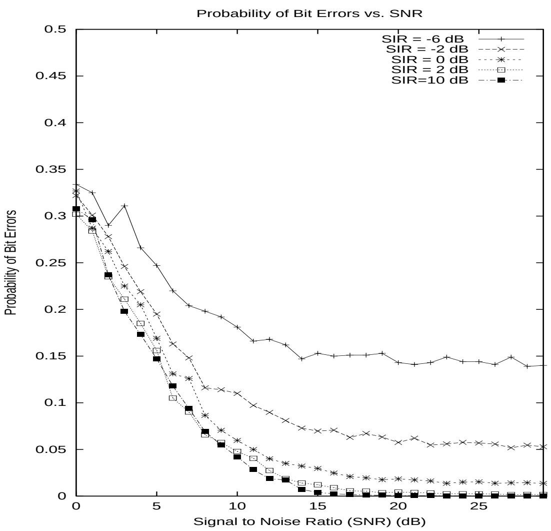

Fig. 5. Impact of WLAN Interference on BT; Co-Channel Interference

ally matter if the BER is 10^{-5}, 10^{-6}, or smaller.

The table implementation does not impact the MAC performance results. A sanity check experiment was conducted to validate the simulation results obtained with the BER tables and compare them to the results obtained using the signal processing simulation model. Therefore, we run a set of two experiments, one with the BER tabulated values and one with the integrated DSP models, keeping all other simulation parameters the same. Figure 6 gives the packet loss and the residual number of errors for a Bluetooth system being interfered with by an 802.11 Direct Sequence system (1 Mbits/s). The results are shown using tabulated BER values for the Bluetooth receiver and using the DSP receiver in real time. As the results indicate, using tabulated BER values instead of the simulation model for the DSP receiver does not affect the packet loss metric. The error is ~ 3.5 % at low distances and ≈ 15 % at higher distances (3 meters). The residual number of errors for the tabulated BER values is consistently lower than for the DSP models (about half); however it follows similar trends. This phenomenon is inherent to the tabulated BER values, which are only accurate to the first decimal.

While we may be looking into increasing the accuracy of the BER tabulations in the future (in order to increase the accuracy of the residual number of errors results), we chose to use this BER tabulation technique for all other PHY receivers, namely the WLAN (1, 11 Mbits/s DSSS and 1 Mbits/s FH). Similar validation experiments were conducted and are consistent with the results presented here.

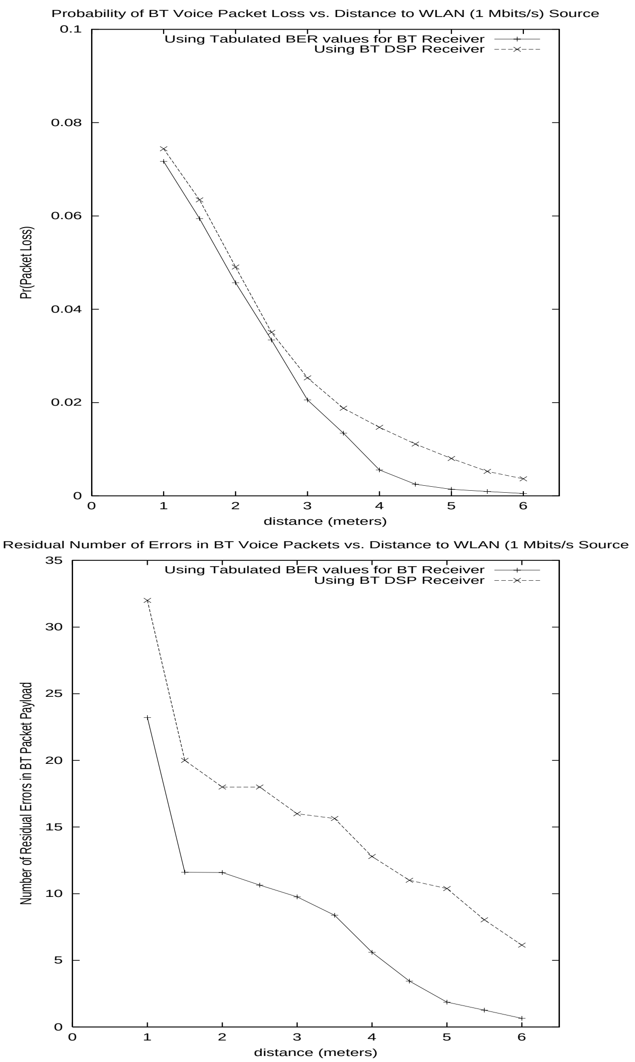

Fig. 6. $\boxed{\frac{(a)}{(b)}}$ MAC results with tabulated BER and DSP model for Bluetooth voice packets with 802.11 interference. (a) Probability of packet loss. (b) Residual errors.

V. Simulation Results

We present simulation results to evaluate the performance of Bluetooth in the presence of WLAN interference and vice versa. All simulations are run for 30 seconds of simulated time. The performance measurements are logged at the slave device for Bluetooth and at the Mobile device for the WLAN. The mean access delay result is normalized by the mean delay when no interference is present. We use the configuration and system parameters shown in Table II.

For Bluetooth, we consider two types of application, namely voice and internet traffic. For voice, we assume a symmetric stream of 64 kbits/s each way using $HV1$ packet encapsulation. For modeling internet traffic, we consider a LAN access application. This is typically a

TABLE II

SIMULATION PARAMETERS

Simulation Parameters	**Values**
Propagation delay	5 μs/km
Length of simulation run	30 seconds
Bluetooth Parameters	**Values**
LAN Packet Interarrival Time	29.16 ms
ACL Baseband Packet Encapsulation	DM5
SCO Baseband Packet Encapsulation	HV1
Transmitted Power	1 mW
Slave Coordinates	(0,0)
Master Coordinates	(1,0)
WLAN Parameters	
Packet Interarrival Time for 1 Mbits/s	10.56 ms
Packet Interarrival Time for 11 Mbits/s	2.52 ms
Transmitted Power	25 mW
AP Coordinates	(0,15)
Mobile Coordinates	(0,d)
Packet Header	224 bits
Slot Time	$2 * 10^{-5}$ seconds
SIFS Time	$1 * 10^{-5}$ seconds
DIFS Time	$5 * 10^{-5}$ seconds
CW_{min}	31
CW_{max}	1023
Fragmentation Threshold	None
RTS Threshold	None
Short Retry Limit	4
Long Retry Limit	7

connection between a PC and an Access Point or between two PCs, and it allows for exchanging TCP/IP or UDP-like traffic. Both slave and master devices generate IP packets according to the distribution presented in Table III. The packet interarrival time is exponentially distributed with a mean equal to 29.16ms, which corresponds to a load of 30 % of the channel capacity (248 kbits/s for both directions). Packets are encapsulated with $DM5$ Baseband packets after the corresponding PPP, RFCOMM, and L2CAP packet overheads totaling 17 bytes are added.

TABLE III

IP TRAFFIC: MESSAGE SIZE DISTRIBUTION

Message Size (bytes)	64	128	256	512	1024	1518
Probability	0.6	0.06	0.04	0.02	0.25	0.03

For the WLAN, we use the IP traffic distribution presented in Table III. We set the offered load to 30% of the channel capacity, which corresponds to mean packet interarrival times of 2.52 ms and 10.56 ms for the 11 Mbits/s and the 1 Mbits/s systems, respectively.

We present the results from four different simulation experiments that show the impact of WLAN interference on Bluetooth devices and vice versa for different applications, namely voice and data traffic. Table IV provides a summary of these four cases, while Figure 7 shows the experimental topology. Please note that the WLAN access point (AP) is fixed at (0,15), while the WLAN mobile is free to move along the vertical axis, *i.e.* its coordinates are (0,d). The Bluetooth devices are fixed at the given locations. In the first two experiments, the mobile is the generator of the 802.11 data, while the AP is the sink. In the last two experiments the traffic is generated at the AP.

TABLE IV

SUMMARY OF THE EXPERIMENTS

Experiment	Desired Signal	Interferer	WLAN AP	WLAN Mobile
1	BT Voice	802.11	Sink	Source
2	BT LAN	802.11	Sink	Source
3	802.11	BT Voice	Source	Sink
4	802.11	BT LAN	Source	Sink

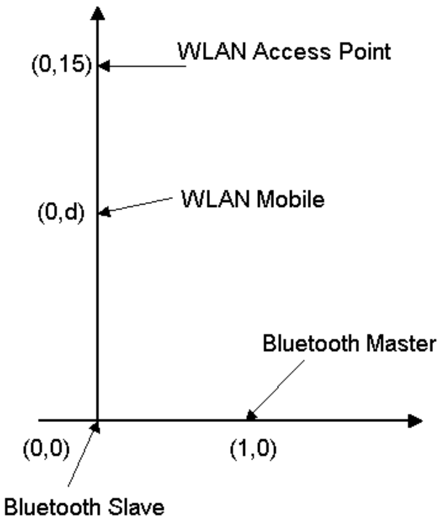

Fig. 7. Experiment Topology

Experiment 1 - We study a voice application generating a symmetric stream of 64 kbits/s each way between the Bluetooth master and slave. The interference is from the mobile sending data packets to the AP and receiving acknowledgments (ACKs) from it. Since most of the WLAN traffic is originating close to the Bluetooth slave, the slave may suffer from serious interference. Figure 8(a) shows the probability of Bluetooth voice packet loss at the slave as a function of the distance to the mobile for interference from both 1 Mb/s and 11 Mb/s 802.11 WLANs.

Consider the 1 Mb/s case first. At one meter, approximately eight percent of the packets are dropped, due to an error in either the access code or the packet header. Even when the packet is accepted, it may still contain a significant number of residual payload errors as shown in Figure 8(b). These errors are measured *after* the FEC decoding is applied. While six errors may not seem to be many, in an eighty bit payload they will lead to poor voice quality. The packet loss is still significant even up to a distance of three meters.

The average length of a 1 Mb/s WLAN packet is 3,168 bits. Thus, its transmission time is on the order of five Bluetooth slots. HV1 packets are being transmitted in every Bluetooth slot, but on different frequencies. Since the direct sequence spreading requires a bandwidth of 22 MHz, there is a significant probability that a WLAN packet may cause interference to multiple Bluetooth packets. In other words, although Bluetooth is hopping to a new frequency for each slot, the 802.11 interference is present in roughly 22 of the 79 channels. Yet with an average interarrival time for the WLAN packets of 10.56 ms, many HV1 packets are successfully received between the transmissions of the WLAN packets.

For the 11 Mb/s case, the general trends are similar. However, the probability of packet loss is slightly lower. Because both the 1 and 11 Mb/s 802.11 modulations use the same bandwidth, the time overlap, not the frequency overlap, is the main factor affecting performance. At 11 Mb/s, it takes only 491 μs, on average, to transmit a packet [2]; therefore, the Bluetooth and WLAN packets are about the same length. Thus, a WLAN packet will usually only interfere with a single Bluetooth one.

Experiment 2 - We focus on a LAN access application. Bluetooth is being used to send data from the master to the slave, and the mobile is still the source of WLAN packets. Figure 9(a) shows the probability of Bluetooth LAN packet loss versus the distance to the mobile, again for both WLAN data rates. While up to almost fourteen percent of the packets may be lost, the use of ARQ still allows the system to be useful. Since a packet sent by the master is acknowledged (positively or negatively) in the next slot, the access delays remain quite small, as seen in Figure 9(b). Even at half a meter with the 11 Mb/s WLAN interference, the access delay is just doubled.

One observation is that for Bluetooth LAN packets, the effect of the different 802.11 data rates is reversed. The probability of packet loss is now higher when the 11 Mb/s system is the interferer. This result is also due to the traffic distributions. The Bluetooth LAN packets have a distribution with an average length that needs two DM5 packets, where each packet requires 2,871 μs for transmission. Now, the Bluetooth and 1 Mb/s WLAN packets are approximately the same length, so it is most likely that a WLAN packet corrupts no more than one Bluetooth packet. The 11 Mb/s WLAN packets are much shorter, and so a number of them can occur during the transmission of the Bluetooth packet. If the Bluetooth LAN packet is on the same frequency as any of these WLAN packets, it will probably be corrupted.

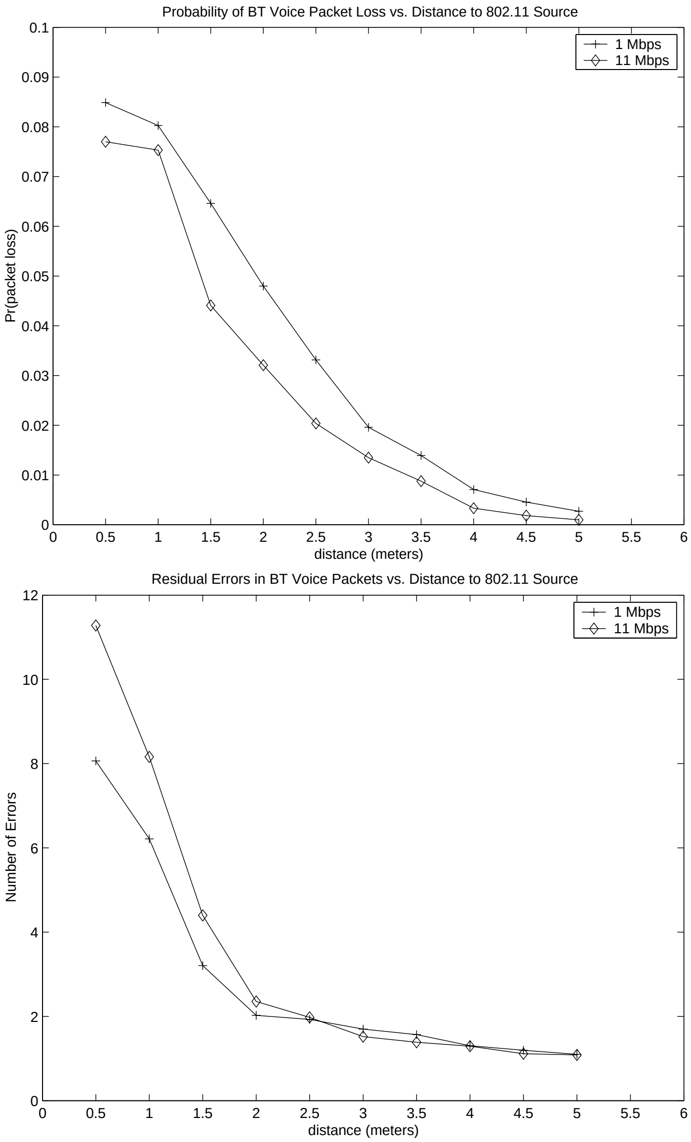

Fig. 8. $\boxed{\frac{(a)}{(b)}}$ Experiment 1. Bluetooth voice packets with 802.11 interference. (a) Probability of packet loss. (b) Residual errors.

Experiment 3 - Next, we are interested in the effect of the Bluetooth voice packets on the 802.11 system. Let the AP be the source of WLAN data packets and the mobile be the receiver. Because the data packets are generally longer then the ACKs, this is a more critical scenario then when the mobile is the source. Figure 10(a) shows the probability of WLAN packet loss as a function of distance to the Bluetooth slave.

For a half meter distance, about sixty five percent of the 1 Mb/s packets are lost. This phenomenon occurs despite the frequency hopping of Bluetooth. The loss rate is so

[2] Including the packet header transmitted at 1 Mb/s.

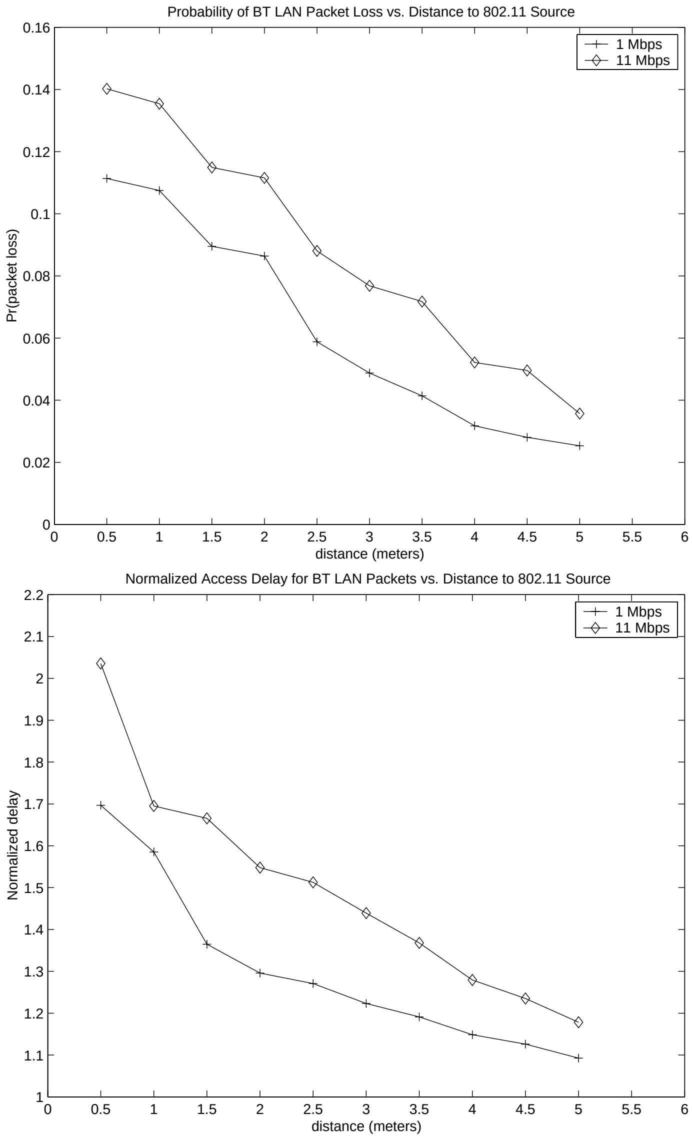

Fig. 9. $\boxed{\frac{(a)}{(b)}}$ Experiment 2. Bluetooth data packets with 802.11 interference. (a) Probability of packet loss. (b) Access delay.

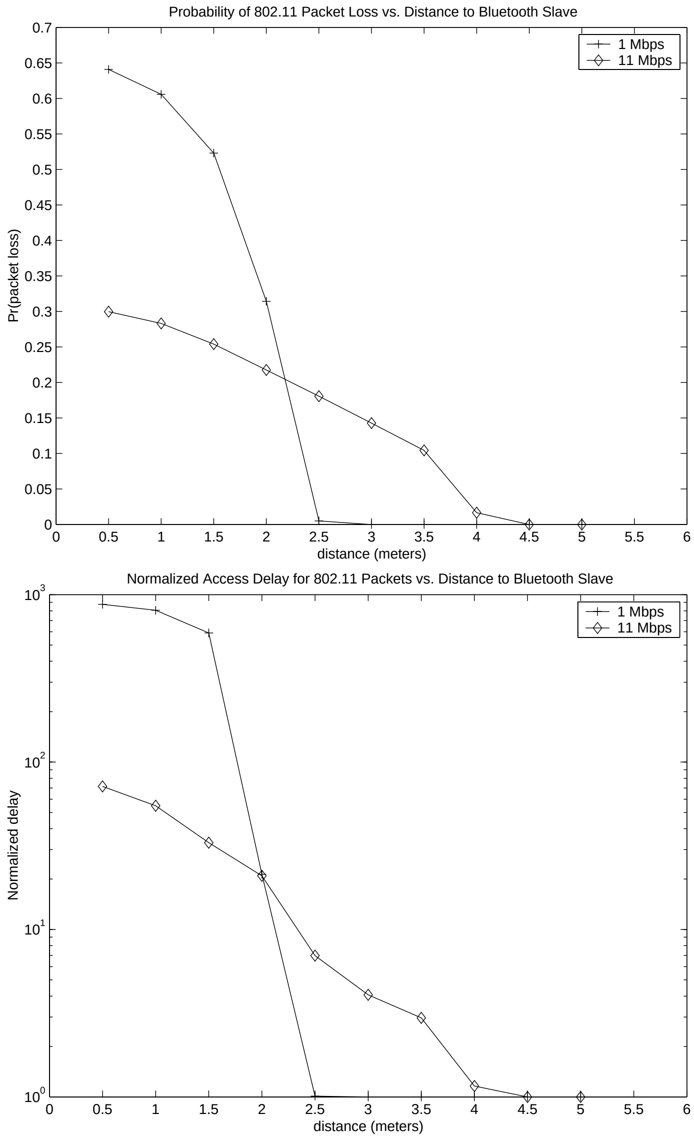

Fig. 10. $\boxed{\frac{(a)}{(b)}}$ Experiment 3. 802.11 packets with Bluetooth voice packets as interference. (a) Probability of packet loss. (b) Delay.

high due to the relatively long length of an 802.11 packet compared to a Bluetooth one. Since 802.11 does not have any error correction, all it takes is a single bit error to effectively erase the packet. When transmitting HV1 voice packets, the Bluetooth system sends many packets during the transmission time of an 802.11 packet. While there is approximately a 22/79 chance that a single packet is in the 802.11 band, this probability must be multiplied by the number of Bluetooth slots occurring doing the WLAN packet transmission. Also, note that the access delay is increased by almost three orders of magnitude due to the interference, as shown in Figure 10(b).

Still considering the 1 Mb/s mode, one sees that the performance significantly improves as the distance exceeds two meters. There appears to be almost a strong "threshold effect." The cause of this phenomenon is the direct sequence spreading, which is reasonably robust to a narrowband interferer such as Bluetooth. Below two meters, the received interference power, based on the topology and transmitter powers, is so much that the 802.11 receiver makes many bit errors. Above this distance, the Barker code correlation effectively spreads the Bluetooth interference while de-spreading the desired signal. Then, the performance of the 1 Mb/s system is better than the 11 Mb/s system.

The 11 Mb/s system has a 0.3 probability of packet loss at a range of half a meter. This probability drops almost linearly to a value near 0.1 for a range of 3.5 meters; the

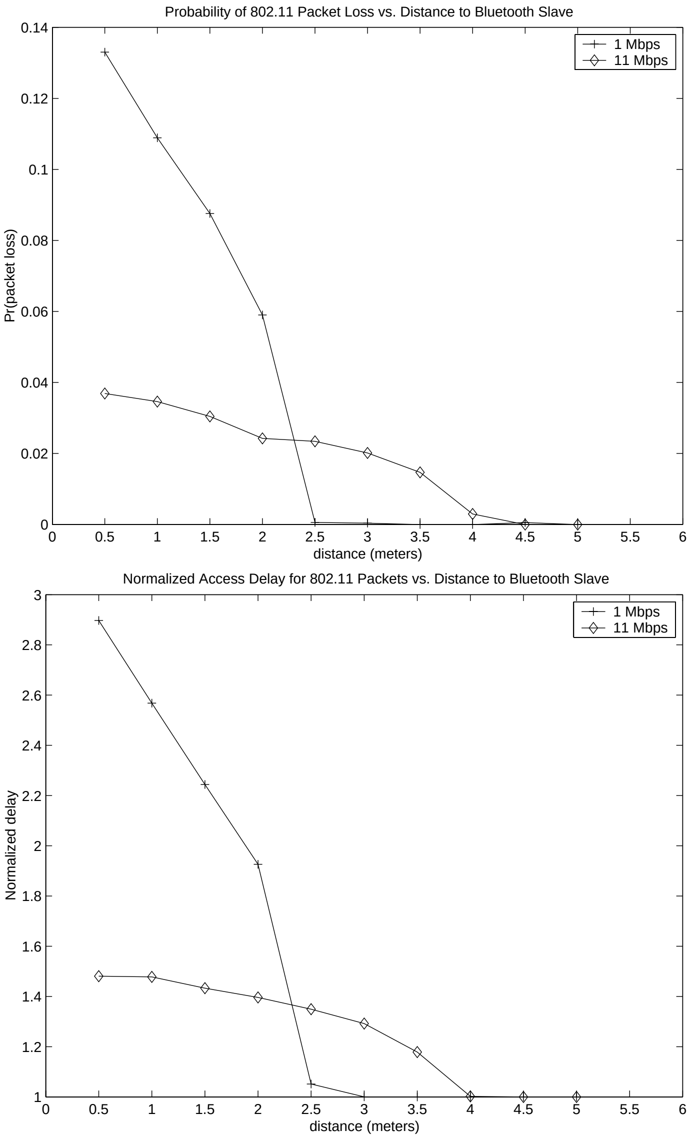

Fig. 11. $\boxed{\frac{(a)}{(b)}}$ Experiment 4. 802.11 packets with Bluetooth LAN packets as interference. (a) Probability of packet loss. (b) Delay.

slope does not increase until after this distance. Thus, there is not as clear a threshold. Since the 11 Mb/s WLAN packets are more than six times shorter than the 1 Mb/s ones, there is a lower probability of overlap in time with the Bluetooth packet. This accounts for the lower packet loss probabilities at distances under two meters. However, the CCK modulation is not as robust at the direct sequence spreading. So, it is unable to provide as low a bit error rate as the DS modulation for distances in the approximate range of 2.5 to 4.5 meters.

Experiment 4 - Let the mobile be the receiver of the WLAN packets from the AP, and consider how the Bluetooth data packets degrade the WLAN performance. Figure 11(a) shows that for both data rates, the probability of an 802.11 packet being lost is much smaller for Bluetooth LAN interference than for Bluetooth voice interference (Experiment 3). The main reason for this difference is that the average interarrival time of the Bluetooth packets is now 29.16 ms. Again, we see a distance where the performance of the 1 Mb/s system becomes better than the 11 Mb/s system, both in terms of probability of packet loss and delay. Beyond four meters, both systems show very little effects from interference, and the higher speed system again becomes the preferred choice. It should be noted that depending on the topology and the transmitter powers, the exact distance where one data rate becomes better than another will change. Yet, it is conjectured that these results will hold for very general scenarios.

Figure 11(b) shows the access delays. At a distance of half a meter, the 1 Mb/s case requires a delay less than three times the delay with no interference, while the 11 Mb/s case has a delay about 1.5 times its optimal value. Please compare this to the previous experiment, where the delay for the 1 Mb/s case is about 900 times greater, and the delay for the 11 Mb/s is approximately 70 times. Not surprisingly, the streaming voice packets cause substantially more interference.

VI. Concluding Remarks

We presented results on the performance of Bluetooth and WLAN operating in the 2.4 GHz ISM band based on detailed channel, MAC, and PHY layer models for both systems. The evaluation framework used allows us to study the impact of interference in a closed loop environment where two systems are affecting each other, and explore the MAC and PHY layer interactions in each system.

Our results indicate that scenarios using Bluetooth voice traffic may be the worst of all interference cases (65% of packet loss for the WLAN 1 Mbits/s system). Also, we note that Blutooth voice may be severely impacted by interference with packet loss of $\approx$ 8%. Moreover, the results suggest that the data rate in the WLAN system may be a factor in the performance, and, the recommended rate for WLAN depends on the topology and the parameters used. Therefore, one may want to exploit the data rate scaling algorithm available in the WLAN system for improving performance. Additionally, results could be obtained with the WLAN Frequency Hopping systems and compared to the Direct Sequence system presented here.

Although the results depend on a number of parameters including traffic distribution, we believe that similar trends should apply for most practical scenarios. Still, there may be some benefit in looking at more complicated scenarios with more than two devices of each type and in studying higher layer traffic such as TCP/IP. Other future directions include exploring acquisition mechanisms for WLAN and Bluetooth and their respective performance in an interference-limited environment. Finally, we hope that the work presented here could represent a first step in the development of coexistence mechanisms.

REFERENCES

[1] Bluetooth Special Interest Group, "Specifications of the Bluetooth System, vol. 1, v.1.0B 'Core' and vol. 2 v1.0B 'Profiles'," December 1999.

[2] K. J. Negus, A. P. Stephens, and J. Lansford, "HomeRF: Wireless Networking for the Connected Home," in *IEEE Personal Communications*, February 2000, pp. 20–27.

[3] IEEE Std. 802-11, "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification ," June 1997.

[4] "BlueHoc: Bluetooth Performance Evaluation Tool," in *Open-Source*, http://oss.software.ibm.com/deverloperworks/opensource/bluehoc, 2001.

[5] M. Takai, R, Bagrodia, A. Lee, and M. Gerla, "Impact of Channel Models on Simulation of Large Scale Wireless Networks," in *Proceedings of ACM/IEEE MSWIM'99*, Seattle, WA, August 1999.

[6] A. Kamerman, "Coexistence between Bluetooth and IEEE 802.11 CCK: Solutions to avoid mutual interference," in *IEEE P802.11 Working Group Contribution, IEEE P802.11-00/162r0*, July 2000.

[7] G. Ennis, "Impact of Bluetooth on 802.11 Direct Sequence," in *IEEE P802.11 Working Group Contribution, IEEE P802.11-98/319*, September 1998.

[8] J. Zyren, "Reliability of IEEE 802.11 WLANs in Presence of Bluetooth Radios," in *IEEE P802.11 Working Group Contribution, IEEE P802.15-99/073r0*, Santa Rosa, California, September 1999.

[9] S. Shellhammer, "Packet Error Rate of an IEEE 802.11 WLAN in the Presence of Bluetooth," in *IEEE P802.15 Working Group Contribution, IEEE P802.15-00/133r0*, Seattle, Washington, May 2000.

[10] S. Zurbes, W. Stahl, K. Matheus, and J. Haartsen, "Radio network performance of bluetooth ," in *Proceedings of IEEE International Conference on Communications, ICC 2000*, New Orleans, LA, June 2000, vol. 3, pp. 1563–1567.

[11] C.F. Chiasserini, R. Rao, "Performance of IEEE 802.11 WLANs in a Bluetooth Environment," in *IEEE Wireless Communications and Networking Conference, WCNC 2000*, Chicago, IL, September 2000.

[12] N. Golmie and F. Mouveaux, "Interference in the 2.4 GHz ISM band: Impact on the Bluetooth access control performance," in *Proceedings of IEEE ICC'01*, Helsinki, Finland, June 2001.

[13] J. Lansford, R. Nevo, and B. Monello, "Wi-Fi (802.11b) and Bluetooth Simultaneous Operation: Characterizing the Problem," in *Mobilian White Paper*, www.mobilian.com, September 2000.

[14] P. Varshney and S. Kumar, "Performance of GMSK in a land mobile radio channel," in *IEEE Transactions on Vehicular Technology*, Aug. 1991, vol. 40, pp. 607–614.

[15] M. K. Simon and C. C. Wang, "Differential versus limiter-discriminator detection of narrow-band FM," in *IEEE Transactions on Communications*, Nov. 1983, vol. COM-31, pp. 1227–1234.

[16] T. Ekvetchavit and Z. Zvonar, "Performance of Phase-locked Loop Receiver in Digital FM Systems," in *Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 1998, vol. 1, pp. 381–385.

[17] D. L. Schilling, L. B. Milstein, R. L. Pickholtz, R. W. Brown , "Optimization of the processing Gain of an M-ary Direct Sequence Spread Spectrum Communication System," in *IEEE Trans. on Communications*, Aug 1980, pp. 1389–1398.

[18] K. Halford, S. Halford, M. Webster, and C. Andren, "Complementary code keying for rake-based wireless communication," in *Proceedings of the 1999 International Symposium on Circuits and Systems*, 1999, vol. 4, pp. 427–430.

[19] A. Soltanian and R. E. Van Dyck, "Physical layer performance for coexistence of Bluetooth and IEEE 802.11b," in *Virginia Tech Symposium on Wireless Personal Communications*, June 2001.

[20] M.K. Simon, J.K. Omura, R. A. Scholtz, and B.K. Levitt, *Spread Spectrum Communications*, Computer Science Press, 1985.