Wireless Pers Commun (2012) 63:45–63
DOI 10.1007/s11277-010-0107-4

Collision Localization for IEEE 802.11 Wireless LANs

Sam M. Keene · Jeffrey B. Carruthers

Published online: 15 August 2010

Abstract In this paper, we examine an algorithm for estimating the location of packet collisions, in the presence of bandlimited multipath channels. We propose an improvement to the collision localization algorithm to further enhance its performance to compensate for the increased impairments of the multipath channel. We then examine the collision localization algorithm in conjunction with two common wireless LAN standards, 802.11b and 802.11a/g. We show that for the 802.11b standard, the collision localization algorithm performs well, even in the presence of a multi-path channel. We also show that the 802.11 a/g standards are compatible with collision localization. However, we will see that the IFFT/FFT operations required to perform OFDM transmission limit the effectiveness of collision localization. We therefore also investigate collision localization in conjunction with block-based single carrier transmission, a comparable technology to OFDM, and offers some advantages when used with collision localization. In addition, we also investigate two applications of collision localization. First, we will show that collision localization in conjunction with Viterbi decoding with erasures can, in many cases, allow for corrupted packets to be completely recovered at the receiver. Second, we will also demonstrate that collision localization can be used to combat narrow-band interference, such as Bluetooth, in 802.11 a/g networks.

Keywords Wireless communication · Packet collision · IEEE 802.11

1 Introduction

In wireless networks, a large source of packet loss are collisions [1]. A collision occurs when two packets arrive at the receiver in the same time frame. Typically, the receiver is expecting one packet, but not the other. The interference from the undesired packet can cause a large

S. M. Keene (✉)
The Cooper Union, New York, USA
e-mail: keene@cooper.edu

J. B. Carruthers
Boston University, Boston, USA

number of bit errors, and the packet will fail its checksum. A collision is then declared after the checksum has failed, and the entire packet is deemed to be lost, requiring retransmission. Most packet collision models which are used to analyze MAC protocols generally assume that a collision equals total packet loss [2], in which both packets are completely corrupted and unrecoverable. However, in reality, it is unlikely that the two packets were perfectly aligned in time. It is much more likely that the two packets only overlapped for a short period of time. An example of this is the masked node problem [3], in which DATA and control packets collide. Typically, the DATA packet is likely to be much larger than the control packet. In this case, a large portion of the received packet contains useful data, which would be wasteful to discard. Discarding this information could have a negative impact on the throughput and delay of higher level protocols.

1.1 Related Work

Recently, there has been increased interest in merging the MAC(medium access control) and PHY(physical)layers in wireless networks in order to improve performance [4]. This is a form of cross-layer design, where the boundaries between network layers are blurred in order to improve performance.

To begin, the capture effect has been a widely studied phenomena [5]. Capture effect describes the scenario when a packet is successfully decoded despite being corrupted by a collision. This occurs because although the interfering signal is within range of the receiver, it is far enough away that the interference power will not cause any errors in the desired packet. In [5], it was shown that throughput increases as the probability of capture increases. However, no specific attempts have been made to increase the capture probability at the physical layer. Our work can be viewed as an attempt to significantly increase the probability that capture occurs at the receiver by performing additional physical layer computation.

There have been several attempts to perform physical layer processing to improve performance of the MAC layer protocols. The two main approaches have been packet combining [6] and hybrid-ARQ(Automatic Repeat Request) [7]. Packet combining works by storing all retransmissions of a packet after a collision, and then using the retransmitted packet in order to get a MIMO(Multiple-Input Multiple Output) diversity gain. However, the channel gains are typically highly correlated, thus making a MIMO diversity gain in reality difficult to achieve. The Blind Network Diversity Multiple Access (B-NDMA) protocol [8] attempts to resolve the problem of highly correlated channel gains by introducing artificial channel gains. However, both of these techniques operate under the assumption that the entire packet is lost when a collision occurs. In [9], a similar approach is taken, utilizing spatial diversity in addition to network diversity to address the masked node problem. However, all of the diversity based collision resolution schemes require additional resources, in the form of additional transmissions, additional antennas or both. Our technique can be applied purely at the receiver, and would eliminate the need for additional retransmissions and antennas.

Other attempts have been made recently at the MAC layer to better handle collisions, such as Fast Collision Resolution [10] and related protocols [11]. However, these protocols merely promise faster resolution of collisions by adjusting parameters such as the contention window size and backoff timers. They do not eliminate collision, but seek to minimize their impact. Our work could be used in conjunction with such a scheme to further improve network throughput.

Finally, a recent approach that recognizes the fact that only a small portion of symbols in the packet may be corrupted have been made. In [12], where packets are divided into sub-packets, and only the corrupted sub-packets are retransmitted. However, this approach

requires additional parity bits to be transmitted to encode a checksum in each sub-packet. Moreover, it requires additional retransmission of corrupted bits, whereas our technique requires no additional parity bits, and is able to reconstruct a corrupted packet without requiring a retransmission. Another approach is the ZigZag protocol, [13], which uses multi-user detection to cancel out collisions. However, ZigZag works by eliminating secondary collisions that occur in the retransmission of already collided packets. A technique that can immediately cancel collisions as they occur would be ideal. In addition, the ZipTx protocol [14] highlights the benefits that partial packet recovery offer to wireless LANs, it does so by using per-block CRCs, or by changing the FEC to an incremental Reed-Solomon code. Both of these methods involve changing the 802.11 standard, and make no attempt to locate and cancel the collision.

1.2 Proposed Solution

To address this, we will investigate an algorithm that can isolate the exact region within a packet that has been corrupted by collision. This algorithm will take a corrupted packet and denote the start and end time of a collision. Furthermore, we will show that this collision location information can be used in conjunction with error control coding to significantly improve packet reception rates.

1.3 Paper Organization

In Sect. 2, we will present an overview of the system model including AWGN and a multipath channel. Then, in Sect. 3, a brief review of the collision localization procedure is presented. In Sect. 4, we investigate the effects of multi-path on collision localization and ways to mitigate it in Sect. 5. In Sect. 6, we discuss collision localization in the 802.11 b standard and demonstrate that the algorithm works with this standard. Then, in Sect. 7, we examine the the IEEE 802.11a/g physical layer scheme. The IEEE 802.11a/g standard relies upon OFDM and frequency domain equalization to combat multipath fading. We will see that due to the IFFT/FFT operations of OFDM, that collision localization is limited somewhat in its ability to isolate in collision location in the time domain. We will therefore also show that collision localization works well in conjunction with block based single carrier transmission [15], which is an alternative to OFDM. Moreover, we will show that collision localization when used in conjunction with Viterbi decoding, can effectively cancel out certain packet collisions and completely recover a packet which would otherwise be lost. Finally, in Sect. 9, we show one additional application of the technique, that it can be applied to mitigate Bluetooth interference in 802.11 a networks.

2 System Model

We will use the same system model as in [16]. We model our communication system using discrete symbol, baseband equivalent, M-ary QAM signals in real noise. We assume carrier synchronization and bit synchronization have been correctly performed on the packet of interest, and that the interfering packet does not disrupt the synchronization process. As capture effect [5] is a fairly common phenomenon, this would imply that synchronization is often not disrupted by collision.

Let $\mathbf{X}$ be the signal packet, i.e. the message transmitted from transmitter Tx to the receiver Rx, a one-dimensional vector of length L. The values of $\mathbf{X}$ represent baseband M-ary QAM

symbols taking values from a constellation $\mathbf{S}$ with probability $1/M$. For example, if $M = 4$, then each X_i may take values $\{1 + j, 1 - j, -1 + j, -1 - j\}$. For notational convenience, we will label our signal constellation points S_k where k can range from 0 to $M - 1$. For example, when $M = 4$, $S_0 = 1 + j$, $S_1 = 1 - j$, etc. Let $\mathbf{\Phi}$ be the interfering signal, also M-ary QAM and of length L, however with random start time τ and random duration D. Let τ and D be independent, uniform random integers with range from 1 to L, i.e. $\tau = i$ with probability $1/L$. Therefore,

$$\Phi_i = \begin{cases} 0 & \text{if } 0 < i < \tau \text{ or } \tau + D > 0 \\ S_k & \text{w.p. } 1/M \text{ if } \tau < i < \tau + D \end{cases} \tag{1}$$

Let there be additive white gaussian noise $\mathbf{Z}$ in the system with zero mean and variance σ_n^2. Furthermore, let there be a random fading gain c on the interfering signal, also drawn from a Gaussian distribution with zero mean and variance σ_c^2. We will assume that any fading on the desired signal has been normalized, so that $1/c$ represents the signal to interference ratio (SIR) while the interferer is active. The received signal is therefore the sum of the desired signal, the interference and the noise. We will assume for now that the SNR and SIR $1/c$ are known to the receiver. In addition to AWGN, the transmitted signal $\mathbf{X}$ will also be subjected to a multipath channel. The multipath fading can be modeled as a FIR filter. We will adopt a matrix form for the filtering operation. Let $\mathbf{H}$

The received signal $\mathbf{Y}$ is then the sum of the filtered signal $\mathbf{HX}$, the interference Φ and the noise $\mathbf{Z}$:

$$\mathbf{Y} = \mathbf{HX} + c\mathbf{\Phi} + \mathbf{Z} \tag{2}$$

We assume that $\mathbf{Y}$ is equalized by a linear FIR equalizer, defined by a convolutional matrix $\mathbf{G}$. We also assume that this equalizer has been trained via a short training sequence, and that the equalizer does not adjust its taps after the initial training state. We therefore are assuming a quasi-static channel, in that while the channel may change from packet to packet, it remains static during packet reception. Therefore, the equalized signal $\mathbf{Y}_{\text{eq}}$ is defined as follows:

$$\mathbf{Y}_{\text{eq}} = \mathbf{G}(\mathbf{HX} + c\mathbf{\Phi} + \mathbf{Z}) \tag{3}$$

$$\mathbf{Y}_{\text{eq}} = \mathbf{GHX} + c\mathbf{G\Phi} + \mathbf{GZ} \tag{4}$$

Depending on the severity of the multipath, and the effectiveness of the equalizer, the equalizer term $\mathbf{G}$ should invert the multipath matrix $\mathbf{H}$ such that:

$$\mathbf{GH} \approx \mathbf{I} \tag{5}$$

Therefore:

$$\mathbf{Y}_{\text{eq}} \approx \mathbf{X} + c\mathbf{G}\Phi + \mathbf{GZ} \tag{6}$$

In Sect. 3 we will review the collision localization algorithm that can be performed on $\mathbf{Y}_{\text{eq}}$ to locate the bits affected by the collision. How well the approximation $\mathbf{GH} \approx \mathbf{I}$ holds will significantly affect the performance of the collision localization algorithm, and the effects of a non-ideal equalizer will be examined in Sect. 7.

3 Review of Collision Localization Procedure

We briefly review the collision localization procedure. We start by examining the received packet on a symbol by symbol basis, and determining which symbols are likely to be in

error. Then, by calculating the temporal correlation of the suspected errors, we can estimate the length of the collision. Once the collision length has been estimated, the location of the collision in the packet can be estimated by picking the region in the packet of the collision length that contains the most suspected errors.

Determining which symbols are likely to be errors can be done with a likelihood ratio test. In the likelihood ratio test, there will be two possibilities per received symbol, Collision or No Collision. For the No Collision hypothesis,

$$y_j = x_j + z_j \tag{7}$$

and for the Collision hypothesis,

$$y_j = x_j + c * \phi_j + z_j \tag{8}$$

The number of possibilities depends on the modulation order M.

$$p_{y_j|nc}(y_j|\text{ no collision }) = \frac{1}{M}\sum_{k=0}^{M-1} N(y_j; S_k) \tag{9}$$

Because of numerical issues, the probability that a symbol has been corrupted by collision is assumed to be uniform.

$$p_{y_j|c}(y_j|\text{collision}) = 1 \tag{10}$$

Assuming that the cost for declaring a collision or no collision is equal, the likelihood ratio test is

$$p_{y_j|nc}(y_j|\text{no collision}) \underset{\text{collision}}{\overset{\text{no collision}}{\lessgtr}} = \frac{p_{nc}}{p_c} \tag{11}$$

We found that by assuming that a collision or no collision at each bit to be equally likely gave good results. Once the likely error locations have been computed, auto-correlation can provide a good estimate of the collision duration. Assuming the error location estimate is good, and the majority of errors are caused by the collision and not noise, we can form a binary error-no error vector, with 1 corresponding to an error, 0 corresponding to no error. Let ev be the n element vector of error locations. The auto-correlation of ev would be defined as $R_j = \sum_{j=0}^{n}(ev_j)(ev^*_{n-j})$. Consider the case of a perfect estimate of the collision location. The error vector would be a rectangle, and its auto-correlation would be a triangle. The length of the rectangle can then be deduced from the edges of the triangle.

Once the collision length is determined, the collision must be located in the received packet. This can be done with a sliding window approach. Let the estimated length of collision be $\hat{D}$. Starting at the beginning of the error vector, slide a window of length $\hat{D}$, counting the number of estimated errors in each window. The window that contains the most errors is the most likely region that the collision occurred. The complexity of this algorithm is O N log (N), where N is the number of samples per packet, which is a reasonable complexity for a communication receiver algorithm. For a more detailed derivation of the collision localization algorithm, including a discussion of complexity, see [16] and [17].

4 Performance of Collision Localization in Multipath

In this Section, we examine the effects that multipath fading has upon the collision localization algorithm. For a simple example, consider a two-tap channel with coefficients $\alpha_1 = 1-\beta$

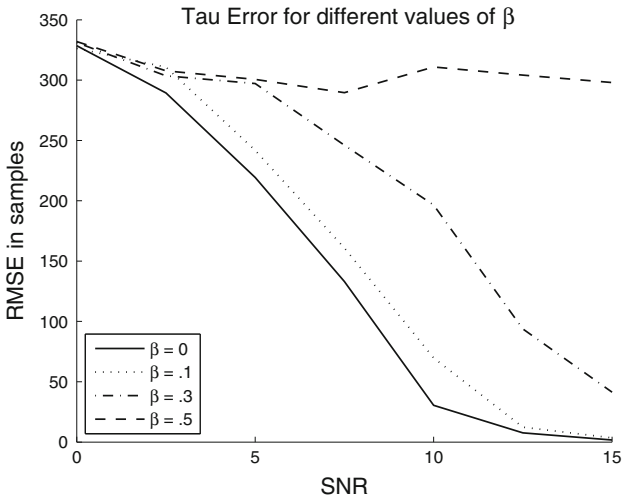


Fig. 1 Collision location estimate error with different values of β, the channel multipath coefficient

and $\alpha_2 = \beta$. We transmitted a 1,000 symbol BSPK signal, with a collision of random size and duration present. The collision has a minimum duration of 20 bits. We examine the performance of the collision localization algorithm for $\beta = 0, \beta = .1, \beta = .3, \beta = .5$. In Fig. 1 is a plot of the mean squared error of the collision start and duration estimates. $\beta = 0$ corresponds to no multi-path, while $\beta = .5$ is fairly severe multipath. The MSE of the duration parameter has similar behavior and therefore the plot is omitted.

From Fig. 1, it is apparent that multipath adversely affects the collision localization algorithm, much like it affects the overall BER of the link. The more severe the multipath, the worse the collision localization algorithm performs. Based on this observation, we are motivated to search for improvements to the collision localization algorithm to better enhance its performance in the presence of a multipath channel.

5 Improving the Collision Localization Algorithm by Windowing

In this section, we propose a modification to the collision localization algorithm to help mitigate the effects of multipath. Recall that our system model is:

$$\mathbf{Y}_{\text{eq}} \approx \mathbf{X} + c\mathbf{G}\mathbf{\Phi} + \mathbf{G}\mathbf{Z} \tag{12}$$

We can improve upon the result in Sect. 4 by noting two things. First, because the interference has been subjected to the filter $\mathbf{G}$, there will be some blurring on the edges. Second, the likelihood ratio test will perform much better if we use several samples at a time. Motivated by this, we propose a small modification to the collision localization algorithm, by introducing a flexible window size W, which will control how many samples of $\mathbf{Y}$ are used in each likelihood ratio test. Instead of determining if a collision is present at a given sample, we will instead determine if a collision is present during a window of length W. The likelihood ratio test is easily modified to perform this operation. The following is the probability there was no collision in the jth window of size W.

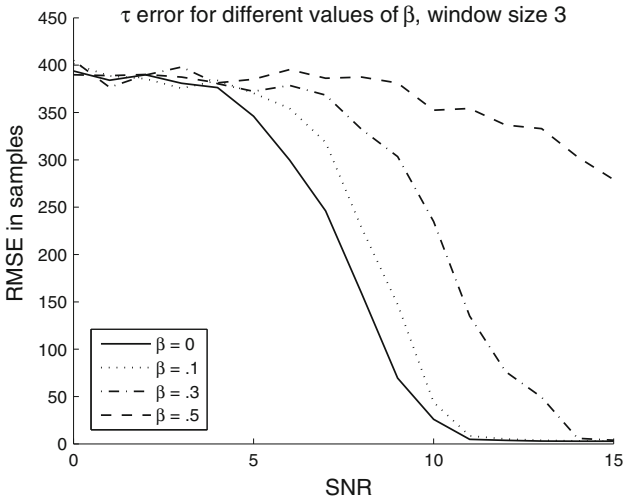


Fig. 2 Collision location estimate error with windowing for different values of β, window length 3

$$p_{y_j,y_{j+1},\ldots y_{j+W}|nc}\left(y_{y_j,y_{j+1},\ldots y_{j+W}}|\text{no collision}\right) = \prod_{i=j}^{j+W}\left(\frac{1}{M}\sum_{k=0}^{M-1}N(y_i;S_k)\right) \quad (13)$$

We again assume the probability of collision to be uniform $p_{y_j|c} = 1$

The windowed likelihood ratio test is now:

$$p_{y_j,y_{j+1},\ldots y_{j+W}|nc}\left(y_{y_j,y_{j+1},\ldots y_{j+W}}|\text{no collision}\right) \underset{\text{collision}}{\overset{\text{no collision}}{\lessgtr}} = 1 \quad (14)$$

Applying the windowed collision localization technique to the multipath example in the previous section yields substantial performance gains. Figure 2 is a plot of the τ error for the same range of β as in the previous section, with a window length of three. The window length of three was chosen because it is slightly larger than the number of taps in the multipath channel. While the most severe multipath case still yields very bad performance, there is an increase of several dB in the remaining cases. The performance of the estimate for D shows similar behavior. It can generally be seen that the performance more rapidly converges to the minimum MSE as the window size increases. The tradeoff in the increased performance is a floor on the MSE is introduced based on the window size. For example, with a window size of five, the MSE floor is 2.5 samples.

6 Collision Localization and IEEE 802.11b

The IEEE 802.11 standard [18] is a widely deployed wireless LAN standard. A large portion of the standard is devoted to specifying the mechanism by which the physical layer is to mitigate multipath fading. As considerable processing is done on the packets at the physical layer, it is important to investigate whether or not the multipath mitigation techniques have an affect on the collision localization algorithm. We begin with the IEEE 802.11b physical layer specification. The components of the physical layer that are of interest to us are the root-raised cosine filtering, spreading code and equalization. We will investigate the

performance of the collision localization algorithm in such a system, and verify its correct operation.

We investigated a large range of data rates by using several values of M. We used the 11-bit Barker code specified in the standard, in two different manners. In the first scenario, we used the Barker code to spread the transmitted symbols, which gave a resulting data rate of 1 Mb/ss when using BSPK modulation. In the second scenario, we simply multiplied the transmitted symbols by the Barker code as if it were a line-code. This corresponds to an 11 Mb/s data rate when the modulation format is 4-ary QAM. 1 and 11 Mb/s represent the minimum and maximum data rates available in the 802.11b standard. The transmit filter was a 49 tap root raised cosine filter, with an upsampling factor of 8 and a roll-off factor of 0.22.

For the multipath channel, we used a 2-tap Rayleigh fading channel, with path delays of 0 and Ts, where Ts is the symbol time. The average path gains were 0 and -5 dB, respectively. The channel was modeled quasi-statically, meaning it remained constant for each packet simulated. With each iteration of the simulation, a new channel was randomly determined and used.

At the receiver, we have a 49 tap root raised cosine filter with a downsampling factor of 8 and a roll-off factor of 0.22. Following the receive filter is an 8-tap linear equalizer. We assume that the equalizer has been correctly trained by a training sequence at the beginning of the packet and does not update its taps after training. As the channel is being modeled as quasi-static, weight updating after training is not necessary. Investigation of non-static channels is left for future work.

Following equalization is despreading and collision localization is the last step performed before demodulation.

We again evaluated transmission of a 1,000 symbol packet, with a collision randomly introduced. Instead of looking at mean squared error, we instead will look at a slightly different performance metric. With each iteration, occasionally the randomly generated multipath channel will have a severe frequency null, which the equalizer is unable to compensate for. This leads to very poor performance when such a channel arises. Although this occurrence is rare, and in general would indicate that the entire packet has been lost, it introduces significant skewness into the mean squared error results.

Due to the skewness introduced by the occasional severe multipath channel, it is useful to look at an alternative metric. We now look at the percentage of time that the τ and D error was within a certain number of symbols. For example, Fig. 3 shows the percentage of time that the τ error was greater than five symbols, for several different modulation formats with the Barker code used to spread the symbols. From this figure, we can see that when the SNR becomes sufficiently high, the percentage of time that the estimate is off by more than five samples is very low, less than 5% for all modulation formats. The performance for the D estimate was very similar. This result could be very useful in conjunction with an ARQ scheme, as the corrupted region of the packet could be marked off, and perhaps padded by a few symbols. The receiver could then request only the corrupted region to be retransmitted, instead of the entire packet, thus saving power and bandwidth. The application of this result to hybrid ARQ is left for future work.

7 Collision Localization and IEEE 802.11a/g

We now examine collision localization in IEEE 802.11 a/g networks. The physical layer in 802.11 a/g networks is characterized by Orthogonal Frequency Division Multiplexing(OFDM). The OFDM signal is wideband and has many desirable properties, mainly

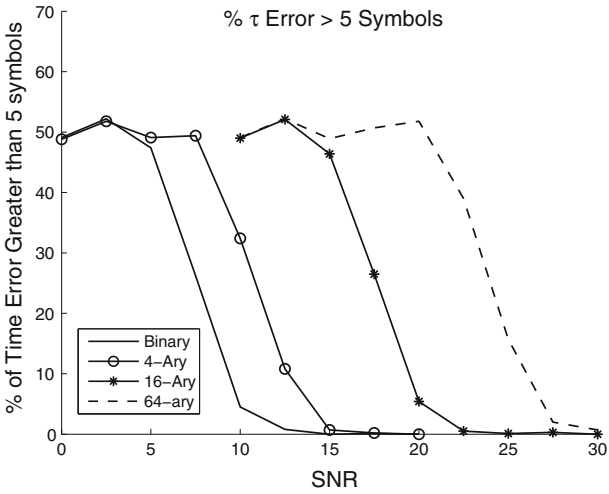


Fig. 3 Percentage of time τ error greater than five symbols with spreading code

offering reduced complexity equalization in multi-path channels. Because of the IFFT and FFT operations required to transmit and receive the OFDM signal, the power of the interference is spread across the entire OFDM symbol. When the entire OFDM symbol is corrupted by collision, then the collision localization procedure works normally. However, when only part of the OFDM symbol is corrupted, the time correlation of error within that symbol is lost. This limits the accuracy of the collision localization algorithm. However, we will show that collision localization is compatible with block-based transmission, an alternative to OFDM. Block-based transmission allows for many of the same benefits of OFDM, but also maintains the temporal correlation of errors, which allows our algorithm to perform well.

We investigate the effect interference has on both OFDM and single-carrier block based transmission, and what the ramifications are for collision localization. It was shown in [19] that frequency domain equalization with single carrier block transmission and frequency domain equalization with OFDM are equivalent. However, as we will see, the two techniques are not equivalent when the desired signal is subjected to a burst of interference.

We adopt the notation used in [20], which compares the two transmission schemes. The model is similar to the one used in Eq. 6, with the addition of FFT and IFFT operations. Let $\mathbf{F}$ be the N-point FFT matrix, where N is the FFT size. The IFFT matrix is the transpose of the FFT matrix, $\mathbf{F}'$. Let the transmitted signal again be $\mathbf{X}$, and the channel be $\mathbf{H}$. However, in contrast to Sect. 2, the matrix $\mathbf{H}$ is now a circulant matrix, due to the presence of the cyclic prefix, which is used in both OFDM and single-carrier block based modulation. The purpose of the cyclic prefix is to eliminate inter-block interference. Its presence has the effect of converting the channel matrix $\mathbf{H}$ into a circulant matrix.

For more information, the reader is referred to [20]. Assuming that the cyclic prefix is sufficiently large, an N-symbol OFDM or single carrier block can be equalized with a frequency domain equalizer.

We assume that an interference signal $\mathbf{\Phi}$ is present, but not affected by the same multipath channel $\mathbf{H}$ as the transmitted signal $\mathbf{X}$. Since we are not trying to remove any channel from the interference, we will simply refer to it as $\mathbf{\Phi}$, which may or may not have been subjected to multipath. We will first look at OFDM in the presence of such interference.

The received OFDM signal is:

$$\mathbf{Y}_{\text{ofdm}} = \mathbf{H}\mathbf{F}'\mathbf{X} + \mathbf{\Phi} + \mathbf{Z} \tag{15}$$

The received signal will be equalized, and the FFT will be taken.

$$\begin{aligned}\mathbf{Y}_{\text{ofdmEQ}} &= \mathbf{F}\mathbf{G}(\mathbf{H}\mathbf{F}'\mathbf{X} + \mathbf{\Phi} + \mathbf{Z})\\ \mathbf{Y}_{\text{ofdmEQ}} &= \mathbf{F}\mathbf{G}\mathbf{H}\mathbf{F}'\mathbf{X} + \mathbf{F}\mathbf{G}\mathbf{\Phi} + \mathbf{F}\mathbf{G}\mathbf{Z}\\ \mathbf{Y}_{\text{ofdmEQ}} &= \mathbf{F}\mathbf{G}\mathbf{H}\mathbf{F}'\mathbf{X} + \mathbf{F}\mathbf{G}\mathbf{\Phi} + \mathbf{Z}_q\end{aligned}$$

Assuming that perfect channel state information(CSI) is available at the receiver, the channel $\mathbf{H}$ can be perfectly inverted. Let $\mathbf{G} = \mathbf{H}^{-1}$. When an OFDM signal is received, it is equalized in the frequency domain, and then the FFT is taken to recover the original signal $\mathbf{X}$. The final, equalized version with interference present is:

$$\mathbf{Y}_{\text{ofdmEQ}} = \mathbf{X} + \mathbf{F}\mathbf{G}\mathbf{\Phi} + \mathbf{Z}_q \tag{16}$$

It was shown in [20] that Z_q has the same correlation and noise power as $\mathbf{Z}$, and is generally white and gaussian. Therefore, assuming that $\mathbf{G}$ perfectly inverts $\mathbf{H}$, the multipath impairment may be completely removed, leaving the transmitted signal affected only by AWGN and the filtered interference term $\mathbf{F}\mathbf{G}\mathbf{\Phi}$.

For the single-carrier case, we have

$$\begin{aligned}\mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{F}(\mathbf{H}\mathbf{X} + \mathbf{\Phi} + \mathbf{Z})\\ \mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{F}\mathbf{H}\mathbf{X} + \mathbf{F}\mathbf{\Phi} + \mathbf{F}\mathbf{Z}\\ \mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{\Lambda}\mathbf{F}\mathbf{X} + \mathbf{F}\mathbf{\Phi} + \mathbf{F}\mathbf{Z}\end{aligned}$$

The inverse of the channel matrix is then applied

$$\begin{aligned}\mathbf{G}\mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{G}\Lambda\mathbf{F}\mathbf{X} + \mathbf{G}\mathbf{F}\mathbf{\Phi} + \mathbf{G}\mathbf{F}\mathbf{Z}\\ \mathbf{G}\mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{F}\mathbf{X} + \mathbf{G}\mathbf{F}\mathbf{\Phi} + \mathbf{G}\mathbf{F}\mathbf{Z}\end{aligned}$$

The inverse FFT is then taken

$$\begin{aligned}\mathbf{F}^{-1}\mathbf{G}\mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{F}^{-1}\mathbf{F}\mathbf{X} + \mathbf{F}^{-1}\mathbf{G}\mathbf{F}\mathbf{\Phi} + \mathbf{F}^{-1}\mathbf{G}\mathbf{F}\mathbf{Z}\\ \mathbf{F}^{-1}\mathbf{G}\mathbf{F}\mathbf{Y}_{\text{sc}} &= \mathbf{X} + \mathbf{F}^{-1}\mathbf{G}\mathbf{F}\mathbf{\Phi} + \mathbf{F}^{-1}\mathbf{G}\mathbf{F}\mathbf{Z}\end{aligned}$$

Note that the matrix $\mathbf{H}$ is a circulant matrix, and is diagonalized by the Fourier transform matrix $\mathbf{F}$

$$\mathbf{H} = \mathbf{F}'\Lambda\mathbf{F} \tag{17}$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where the ith element of Λ corresponds to the ith DFT coefficient of the impulse response for the channel $\mathbf{H}$.

Again, assuming perfect CSI, we let $\mathbf{G}$ be the inverse of the channel, in this case $\mathbf{G} = \Lambda^{-1}$ Since $\mathbf{G}$ is the inverse of the diagonal matrix Λ, and it is also diagonal. As the Fourier transform diagonalizes a circulant matrix. Therefore, the result of the operation $\mathbf{F}^{-1}\mathbf{G}\mathbf{F}$ will result in a circulant matrix, which will be referred to as $\mathbf{C}$. The equalized signal, with interference is then

$$\mathbf{Y}_{\text{scEQ}} = \mathbf{X} + \mathbf{C}\mathbf{\Phi} + \mathbf{Z}_q \tag{18}$$

Note again that the transmitted signal $\mathbf{X}$ has been perfectly recovered, subject only to AWGN and an interference term $\mathbf{C\Phi}$. Similarly to the OFDM case, the noise component Z_q remains Gaussian with the same power. It was shown in [19] that frequency domain equalization with single carrier block transmission and frequency domain equalization with OFDM are equivalent.

However, the key difference between the OFDM equalized signal with interference in Eq. 20 and the single-carrier equalized signal in Eq. 18 is that the interference term in the OFDM signal is multiplied by the FFT matrix and an inverse channel matrix $\mathbf{G}$, while the interference term in the single-carrier signal is multiplied by a circulant matrix $\mathbf{C}$.

This difference has a significant effect on the collision localization algorithm. In the OFDM case, when interference is present for only part of the OFDM signal, the FFT operation spreads the interference noise evenly across the symbol, removing all time correlation. However, in the single-carrier case, the interference is subjected to only a circular convolution by the circulant matrix $\mathbf{C}$. This has the effect of blurring the edges of the interfering signal, but not completely eliminating the time correlation as the FFT operation. This makes the collision localization algorithm significantly more effective when used in conjunction with block-based transmission.

When the entire OFDM symbol or single-carrier block is corrupted by interference, there is no inherent advantage to either transmission scheme, as in both cases, the entire symbol or block is corrupted. Typically, many OFDM symbols or single-carrier blocks will be transmitted in succession. When an interfering signal occurs, the interference generally corrupts multiple symbols. However, interference is unlikely to be perfectly aligned with the OFDM symbol or single-carrier block, and typically the symbols/blocks at the beginning and end of the interference time will be only partially corrupted. Due to the better time-resolution of the interfering signal in the single-carrier based transmission, the collision localization algorithm tends to perform better when used with this format as opposed to OFDM.

Due to the fact that both formats to some extent reduce the temporal coherence of the interfering signal, measuring the performance of the collision localization algorithm in terms of start time τ and duration D as in previous sections is not insightful. However, the collision localization algorithm used in conjunction with Viterbi decoding can be very useful, and will be examined in the following section.

8 Performance of Error Correcting Codes with Frequency Domain Equalization

In this Section, we will examine the performance of the collision localization algorithm in conjunction with frequency domain equalization and forward error correcting codes. We will examine the performance of both OFDM and single-carrier block based transmission formats. It was shown in [17] that collision location information can be used as side-information by the decoder, which in many cases can lead to the cancelation of interference.

We examine performance for several different modulation/code pairs in the 802.11 a/g standard. The interference size is fixed for each simulation and the SNR is varied. The FFT size was 64, and the cyclic prefix used was 16, giving an OFDM symbols size of 80. The same parameters were used for the single-carrier block based transmission. This corresponds to the format used in the IEEE 802.11 a/g standard. For simplicity, no guard bands were used, as co-channel interference was not present. Twenty blocks were transmitted, and the interference was a fixed percentage of that size. The channel consisted of AWGN and both the desired and interfering signal were subjected to a multipath channel. The equalizer was

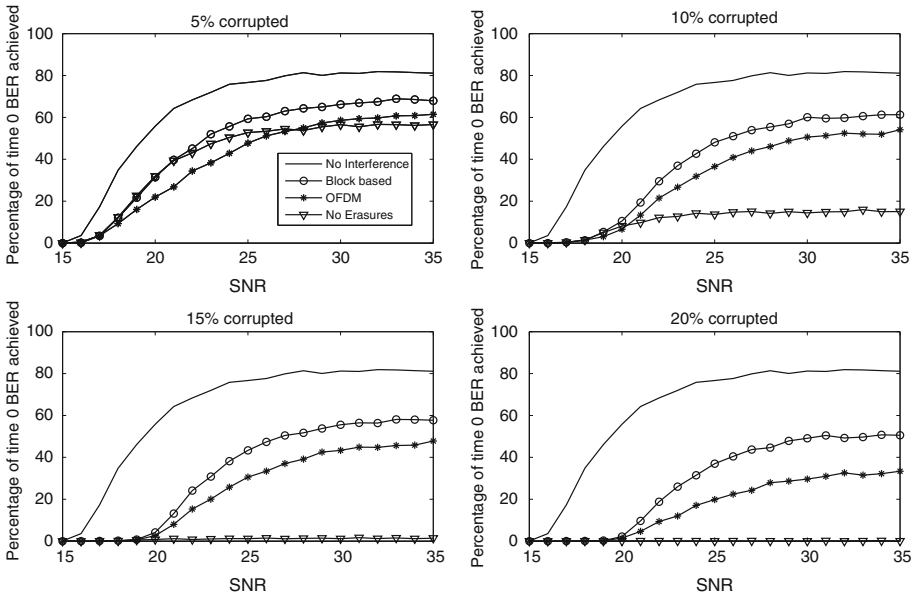


Fig. 4 Packet success probabilities, $M = 16$

trained with a single BPSK 64-point OFDM symbol or single-carrier block, similar to the training done in 802.11 a/g.

As a performance metric, we chose to look at the percentage of time a packet was decoded with zero bit errors, i.e. packet capture. We found this metric to be more informative than the bit error rate (BER). While our algorithm offered improved performance in terms of BER, at first glance, it did not appear to be a major improvement. Upon closer examination, we discovered that occasionally, the quasi-static channel drawn would be particularly bad, having a deep null. When this occurs, it results in a very large BER, as the equalizer cannot invert this channel. The large BER incurred by the occasional deep-null multipath channel significantly skews the BER results, and therefore, it is more instructive to look at a different performance metric, namely the packet success rate.

We then simulated a wide range of modulation/code pairs in the presence of both AWGN and a multi-path channel. The multipath channel was a 3-tap channel with taps at 0, *Ts* and $2 \times Ts$ where *Ts* was the symbol time. The channel had gains of 0, -5 and -10 dB, respectively, and the tap gains were drawn randomly from a Gaussian distribution. A different random channel was applied to both the desired signal and interference. As the path gain of the first tap is random for both the desired and interfering signal, the SIR ratio $1/c$ is no longer fixed, but random. We present results for several different modulation code pairs. We look at packet success rates for several interference ratios, and varied the SNR.

For example in Fig. 4, the modulation rate is $M = 16$, and the code is a rate 1/2 convolutional code with hard decision Viterbi decoding. This corresponds to the 24 Mbs transmission rate of the IEEE 802.11 a/g standard. We look at four different interference ratios, with 5, 10, 15, and 20% of the desired packet being corrupted by collision. In this case, we see a substantial improvement for both the single-carrier and OFDM based transmission with erasure decoding compared to without the erasure decoding technique. At 5% interference, over 60% of the packets are recovered, compared to approximately 50% of the time without

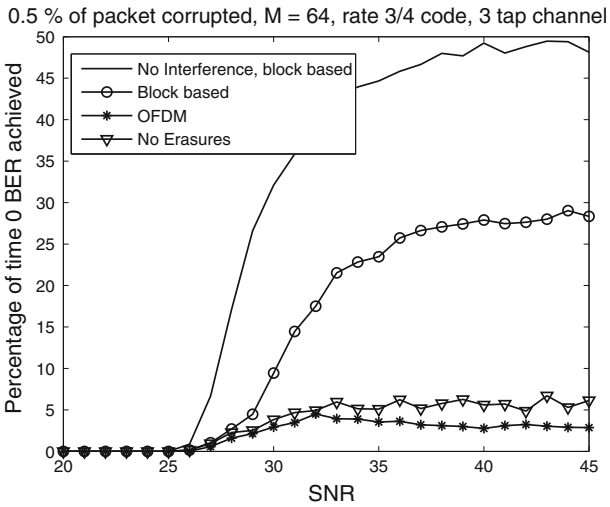


Fig. 5 Packet success probabilities, 0.5% of packet corrupted, $M = 64$, rate 3/4 code

erasure decoding. As the interference increases, without the erasure decoding technique applied, 100% of the packets are lost, while the erasure decoding technique allows for a large percentage of the packets to be recovered. For comparison, the packet success rate without any interference present is also plotted, and for this channel, the maximum packet success rate was also simulated, and was approximately 80%. We found this result to be consistent for all modulation formats in the 802.11 a/g standard long as the code rate remained 1/2.

Additionally, in Figs. 5 and 6 we looked at the 54 Mbs transmission rate, which uses $M = 64$ and a code rate of 3/4. Due to the weak code, the erasure decoding technique is not as effective. We instead look at interference amounts of 0.5, 1%. In this case, only the single-carrier block based transmission format with erasure decoding is successful in recovering any of the corrupted packet. Without interference, in this channel, only 50% of packets are successfully decoded. Without the erasure decoding technique, most packets are lost due to interference. However, with the erasure decoding technique, approximately 25% of the packets are still recoverable. Although the interference percentage is small, consider that a CTS or ACK packet is 14 bytes long, and the typical RTS/CTS threshold is 2,347 bytes. So, the RTS/CTS handshake will only be used when the data length exceeds this amount. $14/2347 = 0.005965$ or 0.5965% of the minimum data packet size. Therefore, the erasure decoding technique used in conjunction with the block-based erasure decoding technique will be useful, as it recovers approximately half of the packets that would be lost due to collision. Even at the highest data rate, the technique could help mitigate DATA/Control Packet collisions, such as those that occur in the masked node problem. However, in this scenario, the block-based modulation format significantly outperforms the OFDM format, highlighting the penalty paid by the lack of temporal correlation of the errors as discussed in Sect. 7.

9 Collision Localization to Combat Narrow-Band Interference in OFDM

Using the same model as in Sect. 7, but now let the interference $\mathbf{\Phi}$ be a Bluetooth signal. The IFFT of the transmitted signal $\mathbf{X}$ is taken, and the received OFDM signal is now:

$$\mathbf{Y}_{\text{ofdm}} = \mathbf{H}\mathbf{F}'\mathbf{X} + \mathbf{\Phi} + \mathbf{Z} \tag{19}$$

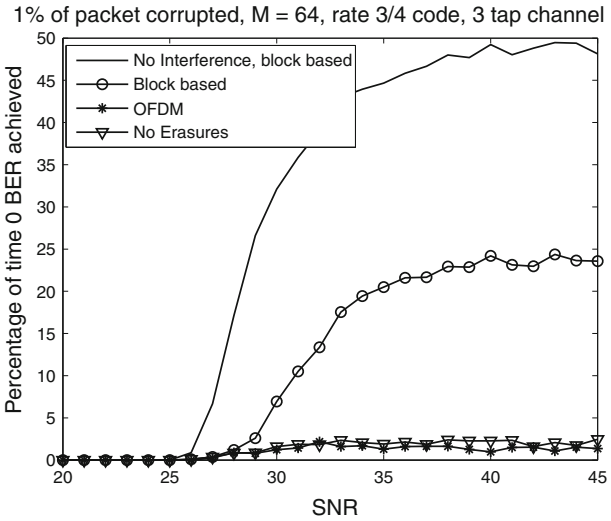


Fig. 6 Packet success probabilities, 1% of packet corrupted, $M = 64$, rate 3/4 code

The received signal will be equalized, and the FFT will be taken.

$$\begin{aligned}\mathbf{Y}_{\text{ofdmEQ}} &= \mathbf{FG}(\mathbf{HF}'\mathbf{X} + \mathbf{\Phi} + \mathbf{Z}) \\ \mathbf{Y}_{\text{ofdmEQ}} &= \mathbf{FGHF}'\mathbf{X} + \mathbf{FG\Phi} + \mathbf{FGZ} \\ \mathbf{Y}_{\text{ofdmEQ}} &= \mathbf{FGHF}'\mathbf{X} + \mathbf{FG\Phi} + \mathbf{Z}_q\end{aligned}$$

The final, equalized version with interference present is:

$$\mathbf{Y}_{\text{ofdmEQ}} = \mathbf{X} + \mathbf{FG\Phi} + \mathbf{Z}_q \tag{20}$$

Since the Bluetooth signal is narrowband, and the OFDM symbol is wideband, the Bluetooth signal will introduce burst errors into the received signal. The collision localization outlined in Sect. 3 can be performed on the equalized signal $\mathbf{Y}_{\text{ofdmEQ}}$ to locate these burst errors.

The IEEE 802.11 a/g implementation uses 52 tones with 12 guardbands and a 64 point FFT to cover a 20 MHz bandwidth, resulting in each OFDM tone occupying 0.3125 MHz. A Bluetooth signal occupies 1 MHz [21], and therefore, a Bluetooth interferer affects approximately three adjacent tones of the OFDM symbol. A Bluetooth signal hops 1,600 time per second, occupying 625 μs If an OFDM packet is 1,000 bytes long and transmitted at 54 Mb/s, the packet duration would be approximately 150 μs [22]. We may therefore assume that the Bluetooth tone does not hop during the OFDM packet. We assume that several OFDM symbols are transmitted in succession with the number of OFDM symbols being N_{ofdm}. The Bluetooth interferer will affect three tones per OFDM symbol. By applying matrix interleaving to the received OFDM packet, the symbols can be rearranged so that like tones of each symbol are placed adjacent to each other. For example, if the OFDM symbol contained four tones f_1 through f_4 and $N_{\text{ofdm}} = 3$, the received OFDM packet would have the following structure:

$$Rx = [f_1, f_2, f_3, f_4, f_1, f_2, f_3, f_4, f_1, f_2, f_3, f_4]$$

After interleaving, the received packet would now be:

$$Rx_{\text{int}} = [f_1, f_1, f_1, f_2, f_2, f_2, f_3, f_3, f_3, f_4, f_4, f_4]$$

By rearranging the received packet, all the errors introduced by the Bluetooth interference will be adjacent, and the collision localization algorithm can be used. As discussed in [17], that by using the collision localization algorithm to provide erasures to a Viterbi decoder can significantly improve performance. To verify that this technique is applicable to narrowband interference into OFDM, we simulated a link with a modulation order of $M = 16$ and a rate $1/2$ code. The OFDM format was as in the IEEE 802.11 a/g standard, with a 64 point FFT, 12 guard bands, and a 16 symbol cyclic prefix. We introduced a narrow-band interferer that corrupted exactly three tones of each OFDM symbol, with a signal to interference ratio (SIR) of 0 dB, for equal power signal and interference. We assume that phase and frequency synchronization have been performed, and that the interference is in phase with the desired signal. To begin, we assume that the channel has been perfectly equalized, and that the Bluetooth interference did not affect the equalization process. The effects of equalization are examined in Sect. 10.

The OFDM packets consisted of 20 OFDM symbols, and we performed 5,000 iterations of the simulation, varying the Signal to Noise Ratio (SNR) from 6 to 20 dB. We used two versions of the collision localization algorithm. Since we know that the interference will cause exactly 3 out of 52 bands to be corrupted, we modify the collision localization algorithm by setting $Pr[\text{collision}] = 3/52$ and $Pr[\text{no collision}] = 49/52$. The likelihood ratio test then becomes

$$p_{y_j|nc}(y_j|\text{no collision}) \underset{\text{collision}}{\overset{\text{no collision}}{\lessgtr}} \frac{Pr[\text{no collision}]}{Pr[\text{collision}]} = \frac{49}{3} \tag{21}$$

The algorithm can be further optimized for speed by noting that the width of the interference is known. Because the received signal has been interleaved so that all corrupted tones are adjacent, then the width of the error burst will be $3 * N_{\text{ofdm}}$. By using this information, we can bypass the correlation and linear fit stages of the collision localization algorithm for substantial computational improvement. In Fig. 7, the BER vs SNR results are plotted. Note that without the erasure correction, the convolutional code is unable to compensate for the Bluetooth interference. The normal and optimized version of the collision localization algorithm effectively cancel out the Bluetooth interference completely, with the optimized version performing slightly better.

10 Bluetooth Cancelation with Equalization

Finally, as the OFDM packet has likely been subjected to an equalizer to compensate for a multipath channel, we need to verify that the collision localization procedure works under these circumstances. We remove the assumption that the signal has been perfectly equalized, and instead train the equalizer with a training signal that has also been subjected to Bluetooth interference. Although the resulting equalizer will be distorted in the frequency bands affected by the Bluetooth interference, the collision localization algorithm is not affected, as it only looking for corrupted symbols. We verified the performance of this approach via simulation.

Again, when evaluating the algorithm with multipath present, we find it more useful to look at packet success rates. We compared three scenarios, the first with perfect erasure location passed to the decoder. The second scenario is with erasures generated by the

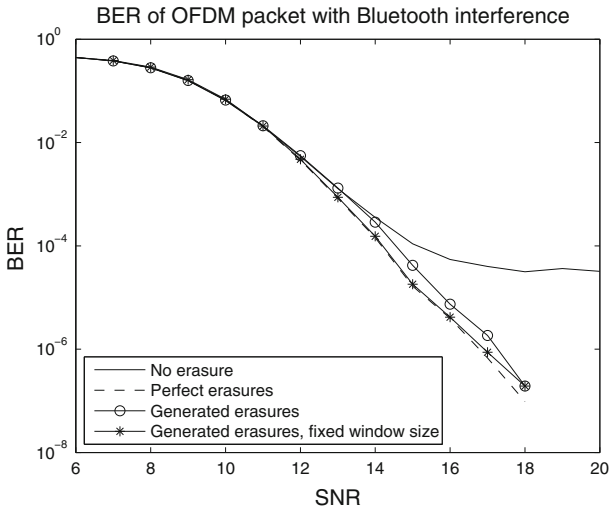


Fig. 7 BER vs. SNR with bluetooth interference in OFDM

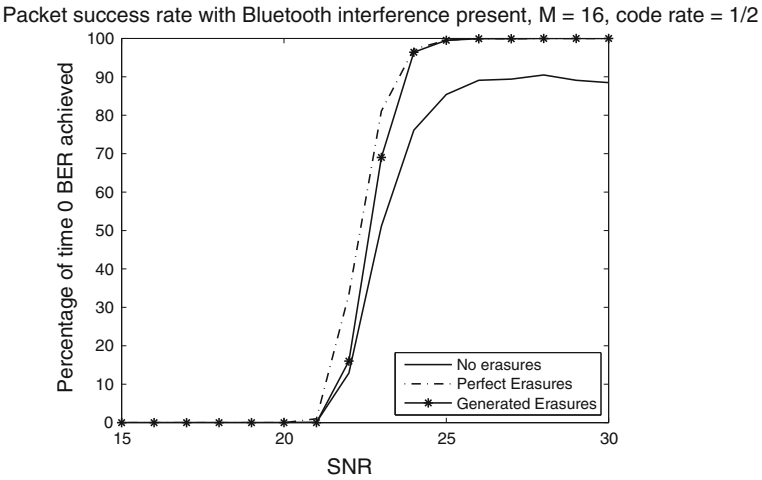


Fig. 8 BER vs. SNR with bluetooth interference in OFDM

collision localization algorithm. The final scenario is with no erasure information provided to the decoder. First, we simulated with $M = 16$, and a code rate of 1/2, corresponding to a the 24 Mbs data rate. The multipath channel has two taps with average gains of 0 and -5 dB, and their values were drawn quasi-statically from a Rayleigh distribution. The assumption is therefore that the multipath channel remains static for the duration of the packet. The taps were located at 0 and *Ts*, where *Ts* is the symbol time. As shown in Fig. 8, we see that as the SNR increases, without the erasure decoding technique, only 80% of the packets are successfully decoded. However, with the erasure decoding technique, 100% of the packets are decoded without errors.

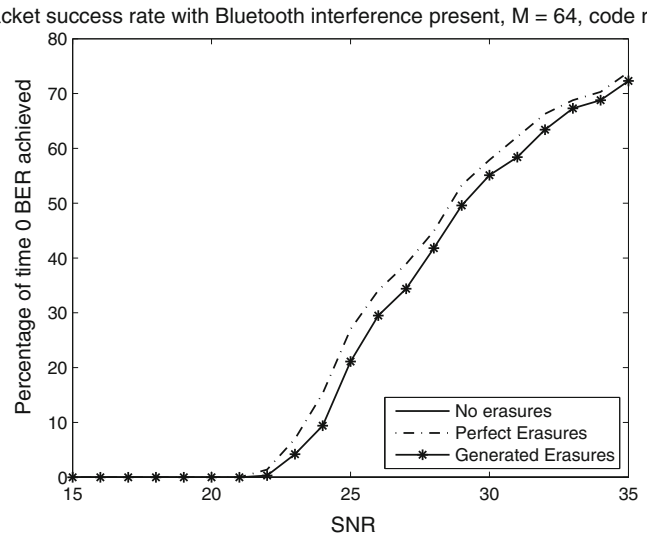


Fig. 9 BER vs. SNR with bluetooth interference in OFDM

Next, we simulated the 48 Mbs data rate present in the 802.11 a/b standard, which has a modulation rate of $M = 64$, and a code rate of 2/3. The results of this simulation are shown in Fig. 9.

We see that in this case, both the perfect and generated erasures allow for recovery of the packet up to 70% of the time when the SNR is above 30 dB. However, without the erasure information, the packet recovery rate 0 at all SNR.

We found in general these result to be consistent over the entire range of data rates in the 802.11 a/g standard. Specifically, when the code rate is 1/2, the Viterbi decoder can cancel interference some of the time, but performs better with the erasure information. When the code rate is 2/3, the viterbi decoder is not capable of canceling out the Bluetooth interference without side information. Finally, when the code rate is 3/4, such as in for the highest rate of 56 Mbs, cancellation of Bluetooth interference is not possible, even with perfect erasure information.

11 Conclusion

In this paper, we examined the collision localization algorithm derived in [16] in typical wireless LAN scenarios. We expanded the physical layer model in [16] to include the presence of a multipath channel. We saw that multipath channels adversely affect the performance of the collision localization algorithm, just as they affect the bit error rate performance of a wireless link. We discovered that when the wireless channel is good enough to communicate through reliably, locating a collision within a packet is possible. We also examined collision localization in conjunction with the IEEE 802.11 a/b/g wireless LAN physical layer standards. We found that in general, collision localization is fully compatible with these standards. However, we did identify that the OFDM modulation technique in the IEEE 802.11 a/g standard degrades the performance of collision localization slightly. We identified an alternative to OFDM that would help resolve the performance issue without sacrificing any of the

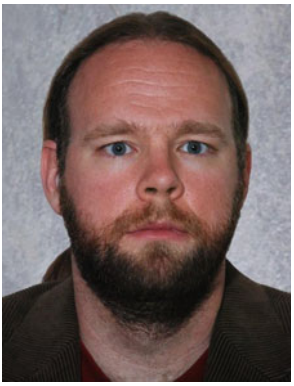
equalization benefit inherent to 802.11 a/g. We examined the performance of the collision localization algorithm in conjunction with multipath channels, and the 802.11 a/g standards by simulation techniques. We then examine a potential application of collision localization. We see that the highly time correlated errors introduced by a collision can be canceled out in many situations by utilizing the collision localization algorithm in conjunction with error control codes and erasure decoding. We examined how often a perfect packet was able to be recovered in the presence of interference. We saw that when the noise level was low, very often the entire packet could be recovered, and when the noise level was high, the packet was completely corrupted by noise. As wireless networks become increasingly dense, collisions will become a greater problem. We have shown that much can be done at the physical layer to resolve what is generally considered to be a MAC layer problem. As we have shown in Sect. 8, the collision localization and erasure decoding technique improves the probability that a packet is correctly decoded in the presence of interference. This can be viewed as increasing the chance that the capture effect has occurred. Therefore, motivated by the results in [5], which indicate that throughput increases as capture probability increases, the collision localization technique has the potential to significantly increase network throughput. Future work will include validating the collision model using collided packets from a real network, and implementing the algorithm in hardware. Finally, the effect on throughput and delay of collision localization in conjunction with erasure decoding needs to be thoroughly examined.

References

1. Toh, C.-K., Delwar, M., & Allen, D. (2002). Evaluating the communication performance of ad hoc wireless network. *IEEE Transaction on Wireless Communications, 1*, 402–414.
2. Kleinrock, L., & Tobagi, F. A. (1975). Packet switching in radio channels: Part I—Carrier sense multiple access modes and their throughput-delay characteristics. In *IEEE Transactions on Communications* (pp. 1400–1416).
3. Ray, S., Carruthers, J. B., & Starobinski, D. (2005). Evaluation of the masked node problem in ad-hoc wireless LANs. *IEEE Transactions on Mobile Computing, 4*, 430–432.
4. Srivastava, V., & Motani, M. (2005). Cross-layer design: A survey and the road ahead. *IEEE Communications Magazine, 43*, 112–119.
5. Hadzi-Velkov, Z., & Spasenovsk, B. (2003). An analysis of CSMA/CA protocol with capture in wireless LANs. *Wireless Communications and Networking Conference, 2*, 1303–1307.
6. Tsatsanis, M. K., Zhang, R., & Banerjee, S. (2000). Network-assisted diversity for random access wireless networks. *IEEE Transactions on Signal Processing, 48*, 702–711.
7. Wang, X., & Orchard, M. T. (2003). On reducing the rate of retransmission in time-varying channels. *IEEE Transcations on Communications, 51*, 900–910.
8. Zhang, R., Sidiroplous, N. D., & Tsatsanis, M. K. (2002). Collision resolution in packet radio networks using rotational invariance techniques. *IEEE Transactions on Communications, 50*, 146–155.
9. Wang, Z., Liu, L., & Zhou, M. (2007). Space and network diversity combination for masked node collision resolution in wireless ad hoc network. *IEEE Transactions on Wireless Communications, 6*, 478–485.
10. Kwon, Y., Fang, Y., & Latchman, H. (2004). Design of MAC protocols with fast collision resolution for wireless local area networks. *IEEE Transactions on Wireless Communications, 3*, 793–807.
11. Wang, C., Li, B., & Li, L. (2004). A new collision resolution mechanism to enhance the performance of the IEEE 802.11 DCF, d. *IEEE Transactions on Vehicular Technology, 53*, 1235–1246.
12. Zhou, Y., & Wang, J. (2006). Optimum subpacket transmission for hybrid ARQ systems. In *IEEE transactions on communications* (pp. 934–942).
13. Gollakota, S., & Katabi, D. (2008). Zigzag decoding: Combating hidden terminals in wireless networks, *SIGCOMM '08*.
14. Lin, K. C.-J., Kushman, N., & Katabi, D. (2008). Ziptx: Harnessing partial packets in 802.11 networks, *MOBICOM '08*.
15. Wang, Z., Ma, X., & Giannakis, B., Georgios (2004). OFDM or single-carrier block transmissions? *IEEE Transactions on Communications*, 3, 380–394.

16. Keene, S., & Carruthers, J. B. (2009). Collision and fade localization within packets for wireless lans, *Wireless Personal Communications*.
17. Keene, S., & Carruthers, J. B. (2007). Improved error correction in wireless lans using erasures decoding with collision localization. In *Proceedings of IEEE global telecommunications conference* (pp. 4451–4455).
18. I. W. Group (2007). IEEE 802.11-2007: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In *IEEE*.
19. Sari, H., Karam, G., & Jeanclaude, I. (1995). Transmission techniques for digital terrestrial tv broadcasting. *IEEE Communications Magazine, 33*, 100–109.
20. Goldsmith, A. (2005). *Wireless commuications*. 40 West 20th Street, New York NY 10011: Cambridge University Press.
21. B. S. I. Group, (1999). *Specification of the Bluetooth System*. Available at www.bluetooth.com.
22. Ghosh, M., & Gadam, V. (2003). Bluetooth interference cancellation for 802.11g WLAN receivers, Communications, 2003. ICC '03. In *IEEE international conference on, vol 2* (pp. 1169–1173).

Author Biographies



Sam M. Keene is an Assistant Professor of Electrical Engineering at the Cooper Union. He received his Ph.D. from Boston University, where his research has focused on cross-layer techniques to resolve collisions in wireless random-access networks. He also received an interdisciplinary certificate in computational science, where he has worked on remote sensing problems for environmental monitoring. Prior to Boston University, he was with The Mathworks, Inc. where he worked for several years as a communications software engineer. His research interests include wireless communication and networks, cross-layer design, and remote sensing.

Jeffrey B. Carruthers is an Associate Professor of Electrical and Computer Engineering at Boston University. He received the Ph.D. in Electrical Engineering from the University of California, Berkeley in 1997. His research interests are in digital communications, wireless communications and free-space optical communications. He has been a senior member of IEEE since 2003.