A CART Based Mechanism for Collision Detection in IEEE 802.11

Muhammad Naveed Aman and Biplab Sikdar
Department of ECSE, RPI, Troy, NY 12180 USA

Abstract—The ability to detect and distinguish packet errors due to collisions from those caused by channel errors can significantly impact the performance of medium access control (MAC) protocols such as IEEE 802.11. In particular, such mechanisms affect the backoff mechanism as well as rate adaptation algorithms. This paper presents a real-time algorithm based on classification and regression trees (CART) for distinguishing packet corruption and losses due to channel errors from those caused by collisions with other simultaneous transmissions. Using a set of four metrics, we propose a classifier tree that reduces classification errors by considering the impact of channel variations and collisions on bit errors from multiple, disparate perspectives. Extensive simulation results are used to verify the superior performance of the proposed technique over existing mechanisms.

I. INTRODUCTION

The use of wireless technologies for network access has become very popular over the last decade. Among the various wireless access mechanisms available, IEEE 802.11 or WiFi is the most popular for local area networks (LANs). Significant work has been done in the recent past to improve the performance of IEEE 802.11. However, various factors associated with the use of carrier-sense multiple access (CSMA) as the underlying protocol for sharing the medium leads to certain inefficiencies that have not been addressed yet. This paper addresses the problem of collision detection in wireless networks using IEEE 802.11 as the MAC protocol.

The inability of a wireless radio to transmit and listen at the same time implies the IEEE 802.11 relies on CSMA with collision avoidance (CA) rather than CSMA with collision detection (CD) as done by Ethernet or IEEE 802.3. In addition, unlike IEEE 802.3 where all nodes sharing a collision domain are able to receive each other transmissions, nodes using IEEE 802.11 may be too far from some nodes to receive their transmissions (and vice versa) but still cause collisions for each other [20]. The inability of a wireless node to “detect” a collision has two detrimental effects: First, bandwidth is wasted on colliding transmissions instead of terminating them. Second, a node unnecessarily performs backoff and defers from accessing the channel even when a packet loss is caused due to channel errors, since the node is unable to distinguish between a channel error and a collision. This is because IEEE 802.11 uses a mechanism where acknowledgment (ACK) packets are used to infer the successful delivery of a packet. In IEEE 802.11, if a node does not receive an ACK for its transmission, it infers the packet was lost due to a collision and performs an exponential backoff. However, in wireless environments a packet loss may also be due to channel errors caused by weak signals and multipath fading.

In addition to helping to optimize the use of the backoff mechanism in IEEE 802.11, the ability to detect collisions and distinguish the packet losses caused by collisions from those caused by channel errors can have a positive impact on other mechanisms as well. For example, greedy data-rate adaptation has been proposed for many high data-rate applications. Aggressive rate adaptation algorithms like AMRR and SampleRate [1], attempt to increase the throughput by using higher data rates with higher loss rates [23]. Thus, if the signal to noise ratio (SNR) is low, a packet modulated at a high data rate may be corrupted by the environment to the extent that the receiver drops it. However, IEEE 802.11 infers this packet loss as a collision and starts the exponential backoff process. By the time the rate adaptation algorithms attribute the packet loss to a weak signal (usually after multiple failed retransmission attempts) the node experiences significant (and unnecessary) loss of throughput.

From the discussion above it is obvious that understanding the root cause of a packet loss is an important factor for improving the performance of wireless networks. Existing literature has shown that the correct classification of packet errors can increase the throughput by 20-60% and reduce retransmissions by 40%, depending upon the channel conditions [2]. This paper addresses the problem of developing a mechanism to accurately determine the root cause of a packet loss by accurately classifying the loss as either caused by a collision or due to channel errors.

A small number of techniques has been proposed in literature for collision detection in wireless networks. A mechanism for collision detection called CARA has been proposed in [5], and is based on the use of multiple RTS/CTS packets. RRAA [4] uses the CARA based RTS/CTS scheme to infer whether a packet loss is due to a collision or weak signal. A method to isolate physical packet errors from collision packet errors using RTS/CTS and packet fragmentation is given in [6]. These approaches require the observation and transmission of multiple RTS/CTS packets, thus requiring a long time to isolate the cause of a packet loss. In contrast, our approach to collision detection is more direct and is based on metrics that can be obtained immediately from the received packet, thus giving us immediate results in real time. In [2] a scheme for collision detection is proposed using three channel quality related metrics and a metric vote. Simulation results presented in this paper show that our proposed technique leads to
significant reduction in the false alarm rates and improvement in the detection accuracy.

In this paper we propose a mechanism based on the use of classification and regression trees to determine the root cause of a packet loss and classify losses as collisions or channel errors. The proposed mechanism uses a vector of metrics associated with the quality of signal reception at a receiver. These metrics and extensive simulation data are used to generate a classification and regression tree for the root cause analysis. We show that the data gathered related to the four metrics can be used to create a very accurate CART model to detect collisions. Extensive simulation results are used to show the superior performance of the proposed collision detection mechanism over the ones proposed in literature.

The rest of the paper is organized as follows. In Section II we present the system model and details of the simulator used. Section III presents the error vector magnitude based metric used in this paper. Section IV presents our CART based collision detection mechanism and performance evaluation results. Finally, Section V concludes the paper.

II. SYSTEM AND SIMULATION MODEL

In this section we describe the system model assumed in this paper. The transmitter and receiver models used for this paper follow the IEEE 802.11a specifications [26]. The transmitter and receiver pairs are assumed to be connected through a multipath channel. We assume a frequency flat multipath Rayleigh fading channel using the Jake’s model [22].

The transmitter produces a sequence of random message bits. These bits are passed through a convolutional encoder, interleaver, and modulator whose parameters are data rate dependent. The modulation parameters dependent on the data rate are given in Table I. The transmitter uses OFDM modulation to map each of the N modulated symbols to a separate subcarrier creating N parallel streams. Finally the N parallel streams are converted to a series of samples by the parallel to serial conversion block.

The receiver first performs serial to parallel conversion, then removes the cyclic prefixes and then performs an FFT on the input. The resulting signal is then passed through a frequency domain equalizer [25], which uses the training symbols. After the equalization, the pilots are separated from the signal and the signal passes through a demodulator, deinterleaver, and a Viterbi decoder.

This system model was also used for all our simulations. The simulation model of the network was created in MATLAB Simulink. Figure 1 shows a network scenario with a transmitter, a receiver, and an interferer that was used for the simulation studies. The frame size of the transmitter was fixed at 32 OFDM symbols per frame, while to get the effect of collisions occurring between packets of different sizes, the frame size of the interferer was chosen to be uniformly distributed between 1 and 32 symbols. The OFDM parameters and timing related parameters for the simulation model are given in Table II. To model highly mobile users the simulator uses a maximum doppler frequency of 100Hz. We consider data rates of 12Mbps (QPSK), 24Mbps (16QAM), and 48Mbps (64QAM) in our simulations. We can set the effect of collisions occurring between packets of different sizes, the frame size of the interferer was chosen to be uniformly distributed between 1 and 32 symbols. The OFDM parameters and timing related parameters for the simulation model are given in Table II. To model highly mobile users the simulator uses a maximum doppler frequency of 100Hz. We consider data rates of 12Mbps (QPSK), 24Mbps (16QAM), and 48Mbps (64QAM) in our simulations. We can set the probability of collisions in the simulator and the simulator can tag the packets involved in a collision, which in turn gives us control over our experiments.

III. EVM FOR COLLISION DETECTION

In this section we introduce a key metric related to the quality of the wireless link that is used by the proposed mechanism for detecting collisions.

Let us consider an OFDM system with BPSK modulation. Assuming the scenario given in Figure 1, a received signal after passing through a multipath fading channel can be

<table>
<thead>
<tr>
<th>Data rate (Mbits/s)</th>
<th>Modulation</th>
<th>Coding rate (R)</th>
<th>Coded bits per subcarrier ( (N_{BPSC}) )</th>
<th>Coded bits per OFDM symbol ( (N_{CBPS}) )</th>
<th>Data bits per OFDM symbol ( (N_{DBPS}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>BPSK</td>
<td>1/2</td>
<td>1</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>BPSK</td>
<td>3/4</td>
<td>1</td>
<td>48</td>
<td>36</td>
</tr>
<tr>
<td>12</td>
<td>QPSK</td>
<td>1/2</td>
<td>2</td>
<td>96</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>QPSK</td>
<td>3/4</td>
<td>2</td>
<td>96</td>
<td>72</td>
</tr>
<tr>
<td>24</td>
<td>16QAM</td>
<td>1/2</td>
<td>4</td>
<td>192</td>
<td>96</td>
</tr>
<tr>
<td>36</td>
<td>16QAM</td>
<td>3/4</td>
<td>4</td>
<td>192</td>
<td>144</td>
</tr>
<tr>
<td>48</td>
<td>64QAM</td>
<td>2/3</td>
<td>6</td>
<td>288</td>
<td>192</td>
</tr>
<tr>
<td>54</td>
<td>64QAM</td>
<td>3/4</td>
<td>6</td>
<td>288</td>
<td>216</td>
</tr>
</tbody>
</table>

**TABLE II: OFDM and Timing related Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{SYM} ): Samples per OFDM symbol</td>
<td>80</td>
</tr>
<tr>
<td>( N_{FFT} ): FFT Length</td>
<td>64</td>
</tr>
<tr>
<td>( N_{SD} ): Number of data subcarriers</td>
<td>48</td>
</tr>
<tr>
<td>( N_{SP} ): Number of pilot subcarriers</td>
<td>4</td>
</tr>
<tr>
<td>( N_{ST} ): Number of total subcarriers</td>
<td>52 ((N_{SD} + N_{SP}))</td>
</tr>
<tr>
<td>( N_{TRAIN} ): Number of training symbols</td>
<td>2</td>
</tr>
<tr>
<td>( \Delta_f ): Subcarrier frequency spacing</td>
<td>0.3125 MHz (=20 MHz/64)</td>
</tr>
<tr>
<td>( T_{FFT} ): IFFT/FFT period</td>
<td>3.2( \mu )s(1/( \Delta_f ))</td>
</tr>
<tr>
<td>( T_{PREAMBLE} ): Preamble duration</td>
<td>8( \mu )s</td>
</tr>
<tr>
<td>( T_{GI} ): Guard Interval (GI) duration</td>
<td>0.8( \mu )s(( T_{FFT} / 4 ))</td>
</tr>
<tr>
<td>( T_{SYM} ): Symbol interval</td>
<td>8( \mu )s</td>
</tr>
<tr>
<td>( T_{LONG} ): Training sequence duration</td>
<td>8( \mu )s(( T_{GI} + 2T_{FFT} ))</td>
</tr>
</tbody>
</table>
where \( i_{n,k} \) is the \( n^{th} \) time domain OFDM transmit symbol over sub-carrier \( k \), \( r_{n,k} \) is the \( n^{th} \) time domain OFDM received symbol over sub-carrier \( k \), \( H_n \) is the channel coefficient in frequency domain, while \( \eta_{n,k} \) is the additive white Gaussian noise (AWGN), and \( \zeta_n \) is the interference due to a collision.

Error vector magnitude (EVM) is widely used as a performance metric for link quality in digital communication systems [26]. EVM expresses the difference between the expected complex voltage value of a demodulated symbol and the actual received symbol. EVM can be described as [3]:

\[
EVM_{RMS} = \sqrt{\frac{\sum_{n=0}^{T-1} \sum_{k=0}^{N-1} |r_{n,k} - i_{n,k}|^2}{T \cdot N \cdot P_0}}
\]  

(2)

where \( P_0 \) is the average power of the symbols, \( T \) is the number of received symbols and \( N \) is the number of subcarriers. Let us now evaluate EVM with the help of two cases. Firstly, in the absence of any collisions the interference (i.e. collision) term in Equation 1 disappears. Substituting this expression for the received signal \( r_{n,k} \) in Equation 2, we get

\[
EVM_{RMS} = \sqrt{\frac{\sum_{n=0}^{T-1} \sum_{k=0}^{N-1} |H_n i_{n,k} + \eta_{n,k} - i_{n,k}|^2}{T \cdot N \cdot P_0}}
\]  

(3)

To obtain the EVM in the presence of collisions, we substitute Equation 1 into Equation 2, to obtain

\[
EVM_{RMS} = \sqrt{\frac{\sum_{n=0}^{T-1} \sum_{k=0}^{N-1} |H_n i_{n,k} + \eta_{n,k} + \zeta_{n,k} - i_{n,k}|^2}{T \cdot N \cdot P_0}}
\]  

(4)

From Equations 3 and 4, it is clear that the EVM value for packets involved in a collision will be higher than the other packets (error free or packets lost due to channel errors). This is the intuition behind using EVM as a metric for collision detection. Figure 2 shows a comparison of the cumulative distribution function (CDF) of EVM for packets involved in collisions and packets that are collision-free. Figure 3 shows the CDF of error burst lengths in the presence and absence of collisions. By observing Figure 2 and Figure 3, we can conclude that the packets involved in a collision will have higher EVM values as compared to collision free packets. This motivates the use of EVM in our technique for differentiating collisions from channel errors.

EVM has many advantages when used as a metric for link quality: it is easy to obtain (vector signal analyzers can be used), it contains additional information about the phase error and amplitude error, and it can be calculated even before a packet is completely decoded etc.

IV. A CART BASED MECHANISM FOR COLLISION DETECTION IN IEEE 802.11

In this section we develop a mechanism based on CART for distinguishing between collisions and channel errors in IEEE 802.11. A set of four metrics is used as inputs to the CART model. Before we present the CART model for collision detection, we first evaluate the effectiveness of each metric individually under various network settings.

A. Metric Evaluation

This section evaluates the effectiveness of different metrics for the purpose of collision detection, when they are used in CART models. Since capture effect has a profound effect on the performance of collision detection techniques [2], we evaluate the performance of the metrics under two network settings: high capture effect and low capture effect. Capture effect is a phenomenon in which the transmission of an attenuated signal is suppressed by the transmission of a strong signal. This means that in the presence of high capture effect, i.e if a collision occurs between a strong signal and a weak signal (attenuated due to larger distance from the receiver), the stronger signal will still be recognizable at the receiver and thus detecting a collision is a challenge. However, in the presence of low capture effect collision detection is less challenging.

We evaluate the effectiveness of a metric based on the following: (i) Probability of false alarm (\( P_{FA} \)) - that is, the cases where a metric predicts a collision when actually there was no collision (the packet may have been corrupted due to weak signal), (ii) Probability of miss detection (\( P_{MD} \)) - that is, the cases where a metric predicts no collision when actually...
there is a collision (the interfering packet may be too weak to affect the original packet), and (iii) the accuracy - that is, the fraction of packets that were classified correctly.

We consider EVM and three other metrics proposed in [2], namely (1) bit error rate per packet ($BER_{pp}$), (2) received signal strength (RSS) which is the aggregate signal plus interference measured in dB and (3) errors per symbol (EPS) which is the average number of bits in error among all the symbols which are in error. For each category (high and low) of capture effect, we create CART models using each metric individually and evaluate the missclassification rates for these models.

Our evaluation of the four metrics is based on simulations that were used to gather data about collisions in a wireless network. The simulator described in section II is used to simulate 5474 packets (32 OFDM symbols per packet) for each of our simulations.

1) Low Capture Effect: To create scenarios with low capture effect, we use simulation topologies where the transmitter and interferer are both at the same distance from the receiver. Thus the scenario given in Figure 1 is simulated with $d_1 = d_2$, with the distances ranging from 3m to 30m. The CART models trained using the data obtained from the simulations for each of the metrics individually result in the missclassification rates shown in Figure 4.

The missclassification rates associated with each of the CART models used in Figure 4 are given in Table III. Table III shows that in the presence of low capture effect, EPS and EVM can detect collisions with high accuracy and low missclassification rates. We can conclude that in this case the use of EVM is sufficient for collision detection because of its advantages over the other metrics, as discussed in section III.

2) High Capture Effect: To create scenarios with high capture effect, we use simulation topologies where the transmitter and interferer are at different distances from the receiver, with the interferer placed at a greater distance from the receiver as compared to the transmitter. Thus the scenario given in Figure 1 is simulated with $d_1 < d_2$, with $d_1 = 5m$ and $d_2$ ranging from 8m to 40m. The CART models trained using the data obtained from the simulations for each of the metrics individually result in the missclassification rates shown in Figure 5.

Figure 5 shows that as the capture effect increases (i.e. as $d_2$ increases) the accuracy of the metrics reduces while the missclassification rate increases. We observe that the accuracy of EVM and EPS remains above 80%, while the accuracy of BER and RSS deteriorate much quickly. We can also see from Figure 5b, that the $P_{FA}$ for EPS, RSS, and EVM are below 10% on the average and are stationary. However, the $P_{FA}$ for
RSS is monotonically increasing with the distance of the interferer. Figure 5c shows that the $P_{MD}$ increases monotonically with the distance of the interferer for all the metrics. However, the $P_{MD}$ for RSS and BER increase at much higher rate. The effect of interferer distance on $P_{MD}$ can be interpreted as follows: As the distance of the interferer from the receiver increases, the effect of the collision reduces i.e., the stronger signal from the transmitter suppresses the transmission from the interferer; thus increasing the probability of miss detecting a collision.

The impact of capture effect on accuracy is noticeable and no single metric has sufficient accuracy for use by techniques such as rate-adaption whose operation is critically dependent on the ability to unambiguously differentiate between collision and channel error related losses. Thus to improve the accuracy and safeguard against the weaknesses of any individual metric, the next section proposes a classifier based on a combination of metrics.

### B. Proposed Collision Detection Model

The results from Section IV-A1 and IV-A2 show that while a single metric CART model is sufficient for collision detection in the presence of low capture effect, they can not detect collisions effectively in the presence of high capture effect. So to improve the accuracy and reduce the missclassification error rates we propose a CART model using multiple predictors.

Using the same simulation data obtained in section IV-A2, we trained a CART model using multiple predictors (metrics), denoted by $X_i$, $i = 1, \cdots, 4$ corresponding to the four metrics described earlier. The objective of the CART model is to find a set of rules which can be used to determine if the value of a dependent variable $Y$ (denoting if a packet loss was caused by a collision or channel errors) from known values of the explanatory variables $X_i$, $i = 1, \cdots, 4$. Using our simulations, we first provide a set of initial data for $X_i$ where the cause of packet loss, i.e. $Y$ is labeled. We then build trees using this initial data and the goal of this process is to maximize the homogeneity of the values of the dependent variable $Y$ in the various partitions. Our proposed classifier is shown in Figure 6.

In Figure 6, the first decision is made is on the EPS value of a packet. The EPS metric fails to detect collisions in situations where the errors (caused by a collision) in a packet can be corrected by the error correcting codes (ECC). This situation arises when the transmission of the interferer is weakened by a fade or attenuation (high capture effect). This causes the EPS value to be zero (or close to zero). However EVM and RSS are two metrics which can give us more insight into the situation. Unlike EPS, EVM and RSS are not based on bits that can either be “on” or “off”. Rather, EVM and RSS are more closely related to the channel dynamics. Moreover, EVM gives a better picture of the channel distortion because of its properties discussed in Section III. Thus the left branch of the classifier in Figure 6 uses RSS and EVM for the purpose of classifying packets which are affected by the high capture effect. On the other hand if the EPS value of a packet is greater than zero, this means that the effect of fading or attenuation is not significant on the transmission of the interferer and the number of errors caused by the collision will be sufficient enough to give us abnormally high values of RSS and EPS. Thus the classifier uses RSS and EPS to classify packets that are affected by low to moderate capture effect (right branch of Figure 6).

### C. Results

In this section we present simulation results to evaluate the performance of the proposed CART based collision detection mechanism and compare it with the state of the art in literature for collision detection. For our comparison, we use the scheme proposed in [2] which used the following three metrics: RSS, BER, and EPS. The method proposed in [2] uses a metric vote, i.e., if any of the metrics indicates a collision, the algorithm infers a collision.

The missclassification rates using our proposed classifier as well as the classifier presented in [2] are shown in Figure 7. These results correspond to the high capture effect scenarios. Since a single metric is sufficient for the low capture effect scenarios, the results for those scenarios are trivial. Figure 7 shows that the proposed CART based classifier trained using multiple metrics gives high accuracy while keeping the missclassification rates low. We also observe that the proposed scheme outperforms the classifier presented in [2]. We note that while $P_{MD}$ is an important metric, $P_{FA}$ will have a more profound effect on the performance of collision detection. A higher $P_{FA}$ will result in an increase in the number of backoffs and retransmissions. Thus we would like to keep $P_{FA}$ at a low level. The results show that there is a significant reduction in the false alarms and a significant improvement in the accuracy with the proposed classifier as the capture effect increases.

### V. Conclusion

This paper proposed a classification and regression tree based methodology to determine if a packet loss in IEEE 802.11 networks was caused by channel errors or due to a
collision with another simultaneous transmission. The proposed methodology uses a vectors of four metrics related to the channel quality as input to the classifier in order to minimize the classification errors. Simulation results are used to evaluate the performance of the proposed collision detection mechanism and verify its superior performance over existing classifiers.

REFERENCES

[31] www.r-project.org