

# Efficient Packet Recovery in Wireless Networks

Muhammad Naveed Aman<sup>†</sup>, Biplab Sikdar<sup>†</sup> and Wai Kin Chan<sup>‡</sup>

<sup>†</sup>Department of ECSE and <sup>‡</sup>Department of ISE,  
Rensselaer Polytechnic Institute, Troy NY 12180 USA

**Abstract**—Wireless medium access control (MAC) protocols usually provide reliability in the presence of packet errors. The efficiency of these reliability mechanisms is generally low since they require the entire packet to be retransmitted even though only parts of it may have been corrupted. To address this issue, this paper presents an efficient mechanism for the recovery of packets corrupted due to both channel errors and collisions. The proposed mechanism first determines the cause of the errors. Next, the symbols with errors are isolated by using the error vector magnitude (EVM) of received symbols as the feature for detection. Using explicit feedback about which blocks of symbols have errors, only the erroneous blocks are then retransmitted. Our results show that the proposed mechanism increases the efficiency of the MAC protocol by providing higher throughput.

## I. INTRODUCTION

Wireless transmissions are susceptible to two sources of errors: (i) poor channel conditions due to fading, path loss and noise; (ii) collisions due to concurrent transmissions from other nodes. To provide reliability, most wireless MAC protocols include mechanisms to detect errors, and retransmission strategies to recover the corrupted packet. The key limitation of existing mechanisms is that when a packet is corrupted during transmission, the entire packet is retransmitted. This leads to unnecessary and costly overhead since most channel errors and collisions affect only part of the packet. To address this issue, this paper presents an efficient packet recovery mechanism for wireless networks such as IEEE 802.11 which use orthogonal frequency division multiplexing (OFDM) as the physical layer.

The most common method to detect corrupted packets is through error detection codes. Mechanisms such as automatic repeat request (ARQ) are then used to retransmit the corrupted packet [1]. Most ARQ schemes provide reliability by retransmitting the entire corrupted packet. A scheme for marking bits that are likely in error is presented in [2] and it is shown that the throughput may be increased by retransmitting only these bits. However, this approach requires specialized hardware, and has a high overhead feedback strategy. A packet recovery mechanism that works by combining multiple copies of a packet from different access points is proposed in [3]. The overhead and delay associated with combining agents limits the application of this scheme. Moreover, only AWGN channels are considered in [3] and its effectiveness in more practical channels has not been established. In [4], a software-only solution to packet recovery based on harnessing partial packets is presented. This scheme relies on incremental redundancy and coding schemes, which can become a bottleneck

at high data rates. In contrast, the proposed packet recovery mechanism does not require any customized hardware, has lower overhead, and provides real time operation.

The main contribution of this paper is a methodology that detects the symbols that were corrupted during transmission. The transmitter can then use precise retransmissions of only the corrupted symbols. The proposed methodology to localize the errors is based on calculating the EVM of the received symbol. Each received packet with errors is first classified as either a packet corrupted due to a collision or due to channel errors. Based on the cause of corruption, two slightly different methodologies are proposed to isolate the symbols with errors. The receiver then conveys the information about the corrupted symbols to the transmitter using the acknowledgment (ACK) packet. The transmitter retransmits only the corrupted symbols, which are then used by the receiver to reconstitute the packet. Extensive simulation results are used to evaluate the performance of the proposed packet recovery mechanism.

The rest of the paper is organized as follows. Section II presents the proposed packet recovery mechanism. Sections III and IV present mechanisms for detecting the cause of packet errors and localizing the errors, respectively. Section V analyses the proposed mechanism, Section VI presents simulation results, and Section VII concludes the paper.

## II. EFFICIENT PACKET RECOVERY MECHANISM

The proposed packet recovery mechanism is shown in Figure 1. When a packet is corrupted (as determined by the CRC check), we first establish the cause (collision or channel error) using the mechanism described in Section III. If the packet is corrupted by a collision we use a thresholding mechanism to identify the blocks in error, while K-means clustering is used if the packet is corrupted by channel errors, as described in Section IV. Once the blocks in error are identified, a bitmap indicating which blocks are corrupt is sent to the transmitter.

On receiving the bitmap, the transmitter has two options for retransmission: (i) it can resend the erroneous blocks as a separate packet; (ii) it can piggyback the erroneous blocks in the next packet. The problem with the latter approach is that the receiver will be unable to differentiate between new blocks and old blocks because the MAC header in the IEEE 802.11 a/b/g has space for only one sequence number. We thus select the first approach since although it increases the overhead, we do not require any changes to the current IEEE 802.11 implementation. However, IEEE 802.11n has a frame format which can accommodate more than one sequence numbers and can support the second approach.

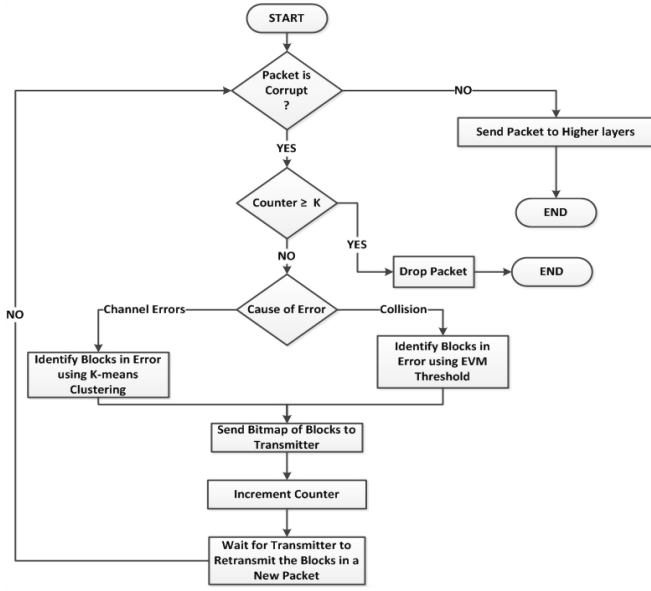


Fig. 1. Proposed Packet Recovery Mechanism

As in IEEE 802.11, the sender tracks the number of retransmission attempts made for each packet. Packets are dropped if the allowed number of attempts ( $K$ ) is exceeded.

### III. DETECTING THE CAUSE OF PACKET CORRUPTION

This section presents a methodology for detecting the cause of packet errors. This information is necessary for accurately isolating the locations of the symbols with errors. We use EVM as the metric to determine the cause of packet corruption.

If  $X_k$  denotes the reference or transmitted signal and  $Y_k$  denotes the received (distorted) signal, then the error vector is  $E_k = Y_k - X_k$ . Let the transmitter and receiver be denoted by  $T_x$  and  $R_x$ , respectively. Consider an OFDM system with  $T$  subcarriers and a frequency flat multipath Rayleigh fading channel. The received time domain OFDM signal  $y_n$  is

$$y_n = H_n * x_n + \eta_n + \zeta_n \quad (1)$$

where  $H_n$  is the Rayleigh channel coefficient,  $\eta_n$  is additive white Gaussian noise with zero-mean and variance  $\sigma_\eta^2$ , and  $\zeta_n$  is the interference due to collision. The reference signal in our model is obtained from the received signal by demodulating the received symbols and then modulating them once again. This re-modulated signal works as an approximation for the transmitted signal.  $x_n$  is the  $n^{\text{th}}$  time domain OFDM signal and can be obtained from  $X_k$ , the M-QAM modulated symbol at the  $k$ th subcarrier as [5],

$$x_n = \text{IDFT}\{X_k\} = \sum_{k=0}^{T-1} X_k e^{j2\pi kn/N} \quad (2)$$

With frequency flat fading, the convolution operator in (1) may be replaced with a multiplication and (1) may be rewritten as

$$y_n = H_n x_n + \eta_n + \zeta_n. \quad (3)$$

Let us introduce a random variable  $Z$  defined as

$$Z = \frac{1}{N \cdot P_0} \sum_{k=0}^{N-1} |E_k|^2. \quad (4)$$

where  $P_0$  is the average power of all the symbols for a given modulation, and  $N$  is the number of received symbols. Let  $e_n$  be the error vector in the time domain i.e.,  $e_n = y_n - x_n$ . We also know that  $e_n = \text{IDFT}\{E_k\}$  and  $y_n = \text{IDFT}\{Y_k\}$ . Thus, applying Parseval's theorem, we can rewrite (4) as

$$Z = \frac{1}{N \cdot P_0} \sum_{n=0}^{N-1} |e_n|^2. \quad (5)$$

Note that  $Z$  is the sum of  $N$  i.i.d. random variables. Using the central limit theorem (for large  $N$ ), we approximate  $Z$  as a Gaussian with probability density function (pdf)

$$f_Z(z) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{-\frac{(z-\mu_Z)^2}{2\sigma_Z^2}}. \quad (6)$$

To obtain the pdf  $f_Z(z)$  of  $Z$  in the presence and absence of collisions, we need to find its mean ( $\mu_Z$ ) and variance ( $\sigma_Z^2$ ) for both cases. We first evaluate  $\mu_Z$  and  $\sigma_Z^2$  for the case when there is no collision. In this case the error vector is given by

$$e_n = H_n x_n + \eta_n - x_n = x_n(H_n - 1) + \eta_n. \quad (7)$$

We assume the path loss law  $l(r) = \frac{1}{r^\alpha}$ , and take  $r_i^{-\alpha}$  as the mean power of the Rayleigh channel ( $H_n$ ). Then,

$$\mu_Z = \varepsilon\{Z\} = \frac{1}{P_0} \left[ \sigma_x^2 \left( \frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}} \right) + \sigma_\eta^2 \right]. \quad (8)$$

To find  $\sigma_Z^2$ , we need to evaluate  $\varepsilon\{Z^2\}$  given as follows:

$$Z^2 = \left( \frac{1}{NP_0} \sum_{n=0}^{N-1} |e_n|^2 \right)^2 = \frac{1}{N^2 P_0^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} |e_{n_1}|^2 |e_{n_2}|^2. \quad (9)$$

We now assume block fading with a block length of  $m$  symbols. Then (9) can be re-written as

$$Z^2 = \frac{1}{N^2 P_0^2} \left[ \left( \sum_{n_1=0}^{m-1} \sum_{n_2=0}^{m-1} |e_{n_1}|^4 + \sum_{n_1=0}^{m-1} \sum_{n_2=m}^{N-1} |e_{n_1}|^2 |e_{n_2}|^2 \right) + \left( \sum_{n_1=m}^{2m-1} \sum_{n_2=m}^{2m-1} |e_{n_1}|^4 + \sum_{n_1=mn_2=0}^{2m-1} \sum_{n_2=0}^{m-1} |e_{n_1}|^2 |e_{n_2}|^2 + \sum_{n_1=mn_2=2m}^{2m-1} \sum_{n_2=2m}^{N-1} |e_{n_1}|^2 |e_{n_2}|^2 \right) + \dots + \left( \sum_{n_1=\beta m}^{N-1} \sum_{n_2=\beta m}^{N-1} |e_{n_1}|^4 + \sum_{n_1=\beta m}^{N-1} \sum_{n_2=0}^{\beta m-1} |e_{n_1}|^2 |e_{n_2}|^2 \right) \right]. \quad (12)$$

Taking the expectation of (12), we get

$$\varepsilon\{Z^2\} = \frac{1}{N^2 P_0^2} \left[ m\varepsilon\{e_n^4\} + \frac{N-m}{m} (\varepsilon\{e_n^2\})^2 \right] \quad (13)$$

where  $\varepsilon\{e_n^2\}$  is given by

$$\varepsilon\{e_n^2\} = \sigma_x^2 \left\{ \frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}} \right\} + \sigma_\eta^2 \quad (14)$$

and  $\varepsilon\{e_n^4\}$  is given by (10). Combining (8) and (13), we can obtain the variance of  $Z$  as  $\sigma_Z^2 = \varepsilon\{Z^2\} - (\mu_Z)^2$ .

Now consider the case when a packet is corrupted by collisions. We assume a network with  $J$  interferers at distance  $r_j$  from  $R_x$ , that transmit with probability  $p$  independent of each other. The starting location of a collision within a packet is assumed to be uniformly distributed between 0 and  $N-1$ .

$$\varepsilon\{e_n^4\} = (3\sigma_x^4) \left( \frac{8}{r_i^{2\alpha}} - \frac{4}{r_i^{\frac{3}{2}\alpha}} + \frac{12}{r_i^\alpha} - 2\sqrt{\frac{2\pi}{r_i^\alpha}} + 1 \right) + 3\sigma_\eta^2 + 6\sigma_x^2\sigma_\eta^2 \left( \frac{2}{r_i^\alpha} - \sqrt{\frac{2\pi}{r_i^\alpha}} + 1 \right) \quad (10)$$

$$\begin{aligned} \varepsilon\{e_n^{*4}\} &= \left[ (3\sigma_x^4) \left( \frac{8}{r_i^{2\alpha}} - \frac{4}{r_i^{\frac{3}{2}\alpha}} + \frac{12}{r_i^\alpha} - 2\sqrt{\frac{2\pi}{r_i^\alpha}} + 1 \right) + 3\sigma_\eta^2 + 6\sigma_x^2\sigma_\eta^2 \left( \frac{2}{r_i^\alpha} - \sqrt{\frac{2\pi}{r_i^\alpha}} + 1 \right) \right] + \\ &6 \left[ \sigma_x^2 \left\{ \frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}} \right\} + \sigma_\eta^2 \right] \left[ 2p\sigma_x^2 \sum_{j=0}^J \frac{1}{r_j^\alpha} \right] + 6p\sigma_x^4 \sum_{j=0}^J \frac{1}{r_j^{2\alpha}} \end{aligned} \quad (11)$$

If a collision starts at symbol  $n_0$ , the error vector is given by

$$e_n^* = \begin{cases} e_n & n < n_0 \\ e_n + \zeta_J & n \geq n_0 \end{cases} \quad (15)$$

where  $\zeta_J$  is given as follows

$$\zeta_J = \sum_{i=0}^J B_i H_{r_i} W_i. \quad (16)$$

Here  $B_i$ 's are i.i.d. Bernoulli random variables with parameter  $p$ ,  $H_{r_i}$  is Rayleigh distributed with mean power  $1/r_i^\alpha$ , and  $W_i$  is the OFDM symbol transmitted by interferer  $i$ , which is approximately i.i.d. Gaussian distributed with zero-mean and variance  $\sigma_x^2$ . We can then rewrite (5) as follows:

$$Z^* = \frac{1}{NP_0} \left( \sum_{n=0}^{n_0-1} |e_n|^2 + \sum_{n=n_0}^{N-1} (|e_n| + \zeta_J)^2 \right). \quad (17)$$

Taking the expectation of (17), we get

$$\varepsilon\{Z^*\} = \frac{1}{P_0} \left[ \sigma_x^2 \left\{ \frac{2}{r_i^\alpha} + 1 - \sqrt{\frac{2\pi}{r_i^\alpha}} + \frac{p(N+1)}{N} \sum_{j=0}^{J-1} \frac{1}{r_j^\alpha} \right\} + \sigma_\eta^2 \right]$$

and similarly, we have

$$\begin{aligned} \varepsilon\{Z^{*2}\} &= \frac{1}{N^2 P_0^2} \left[ \left( \frac{N-1}{2} \right) m \varepsilon\{e_n^4\} + \left( \frac{1}{4} \frac{(N-1)^3}{m} - \right. \right. \\ &\left. \left. \frac{2}{3} (N-1)^2 + \frac{1}{2} (N-1)m \right) (\varepsilon\{e_n^2\})^2 + (N-1) \frac{N+2}{3} \right. \\ &\left. \varepsilon\{e_n^2\} \varepsilon\{e_n^{*2}\} + \frac{(N+1)}{2} m \varepsilon\{e_n^{*4}\} + \left[ \frac{1}{4m} N(N^2 + 5N - 3) \right. \right. \\ &\left. \left. - \frac{1}{3} (5N^2 - N + 2) + \frac{1}{2} m(N+1) + \frac{1}{4m} \right] (\varepsilon\{e_n^{*2}\})^2 \right] \end{aligned} \quad (18)$$

where  $\varepsilon\{e_n^2\}$  and  $\varepsilon\{e_n^4\}$  are given by (14) and (10), respectively.  $\varepsilon\{e_n^{*2}\}$  is given by

$$\varepsilon\{e_n^{*2}\} = \sigma_x^2 \left\{ \frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}} \right\} + \sigma_\eta^2 + 2p\sigma_x^2 \sum_{j=0}^J \frac{1}{r_j^\alpha} \quad (19)$$

and  $\varepsilon\{e_n^{*4}\}$  is given by (11). Thus, we can now obtain the variance of  $Z^*$  as  $\sigma_{Z^*}^2 = \varepsilon\{Z^{*2}\} - (\mu_{Z^*})^2$ .

#### A. Classifying the Cause of Packet Errors

To classify the cause of each packet loss, we first calculate the EVM of each received packet and then compare it against a threshold value. If the calculated EVM is greater than the threshold, the packet is classified as a collision, and vice versa.

The optimum threshold value is determined by choosing the threshold that leads to equal false positive and false negative rates (i.e., the threshold that leads to the crossover error rate).

A false positive is defined as an event where the cause of a packet loss is attributed to a collision, when the actual cause was a weak signal (not a collision). If we denote the threshold by  $\gamma$ , then the probability of a false positive is given by

$$P_e[Z > \gamma] = P_e \left[ 1 - \int_{-\infty}^{\gamma} f_Z(z) dz \right] \quad (20)$$

where  $P_e$  is the symbol error rate. Similarly a false negative is defined as an event where the cause of a packet loss is attributed to a weak signal, when the actual cause was a collision. The probability of a false negative is given by

$$P_{Z^*}[Z^* \leq \gamma] = P_e \left[ \int_{-\infty}^{\gamma} f_{Z^*}(z^*) dz \right]. \quad (21)$$

To obtain the threshold that leads to the crossover error rate, we can find  $\gamma$  by equating (20) and (21). Thus, to get the threshold we need to solve the following equation numerically:

$$\int_{-\infty}^{\gamma} f_Z(z) dz + \int_{-\infty}^{\gamma} f_{Z^*}(z^*) dz = 1. \quad (22)$$

#### IV. IDENTIFYING THE LOCATION OF ERRONEOUS BLOCKS

The error identification mechanism detects blocks of symbols with errors instead of individual symbols with errors. To identify the symbols with errors, we calculate the EVM over blocks of  $m$  symbols ( $m \geq 10$ ). This block based approach serves two purposes: (i) justifying the use of (6) and (ii) lowering the overhead of the feedback sent to the transmitter.

##### A. Error Locations in Packets Corrupted by Collisions

Once a packet is classified as a collision, the receiver compares the EVM of each block with a threshold. The threshold is determined using the mechanism described in Section III. Note that the threshold for identifying blocks with errors is different from the threshold for identifying the cause of a packet loss. The difference arises because the cause of a packet loss is classified using all the symbols in the packet while blocks with errors are detected based only on the symbols within a block. The blocks with EVM greater than the threshold are tagged as erroneous blocks. The receiver then requests the transmitter to retransmit the blocks in error as shown in Figure 1.

##### B. Error Locations in Packets Corrupted by Channel Noise

To detect the locations of the errors in case of corruption due to channel noise, we perform clustering on the EVM

values of the blocks in the packet. Clustering is used instead of thresholding because the difference in the variances of the pdfs of the EVM of blocks affected by channel noise, and the blocks that are unaffected, is small. Ideally, blocks with errors should form one cluster while blocks without error form another cluster.

We use K-means clustering [6] to identify the blocks in error. The process starts with two points randomly chosen as cluster centers. The EVM values of the blocks in the corrupted packet are then assigned to their closest cluster center based on the Euclidean distance. The centroid of the two resulting clusters is then calculated and used as the new cluster centers. The EVM value assignment process is then repeated, and this process continues until the cluster centers stabilize.

## V. EFFICIENCY OF THE RECOVERY MECHANISM

This section evaluates the efficiency of the proposed packet recovery mechanism. We develop models for the throughput for two cases: with and without the proposed mechanism.

Consider a packet with  $N$  OFDM symbols and blocks with  $m$  symbols per block. Let  $K$  denote the maximum number of allowed retransmissions. If we denote the probability of a symbol being corrupted by  $P_{out}^S$ , then we have

$$P[\text{block has an error}] = P_{out_B} = 1 - (1 - P_{out}^S)^m \quad (23)$$

$$P[\text{packet has an error}] = P_{out_P} = 1 - (1 - P_{out}^S)^N. \quad (24)$$

The packet transmission overhead consists of two parts: the fixed overhead consisting of the preamble and the PLCP header, and the MAC header (which depends on the data rate).

### A. Throughput Without Proposed Recovery Mechanism

Let  $X$  be the random variable that a packet transmission needs  $x$  attempts. Then

$$P[X = x] = (P_{out_P})^{x-1} (1 - P_{out_P}) \quad x = 1, 2, \dots \quad (25)$$

The probability of outage, i.e. the probability that the packet needs more than  $K$  transmission attempts, is given by

$$P[X > K] = \sum_{x=K+1}^{\infty} P[X = x] = (P_{out_P})^K. \quad (26)$$

Consider a long period of operation of the network. Such a period can be considered to consist of a sequence of successful packet deliveries (with or without retransmissions) followed by a sequence of one or more outages (where a packet is dropped after  $K$  retransmission attempts). This alternating sequence of successful deliveries and outages forms a renewal process and we calculate the throughput by evaluating the average duration of a renewal period. The number of successful transmissions  $\kappa$  in a period is geometrically distributed as

$$P[\kappa = n] = (P_{out_P})^K (1 - (P_{out_P})^K)^{n-1} \quad (27)$$

for  $n = 1, 2, \dots$ . If the  $i^{\text{th}}$  packet is successful, the number of attempts it takes,  $X_{i|S}$ , is given by

$$P[X_i = x | x \leq K] = \frac{(P_{out_P})^{x-1} (1 - P_{out_P})}{1 - (P_{out_P})^K} \quad (28)$$

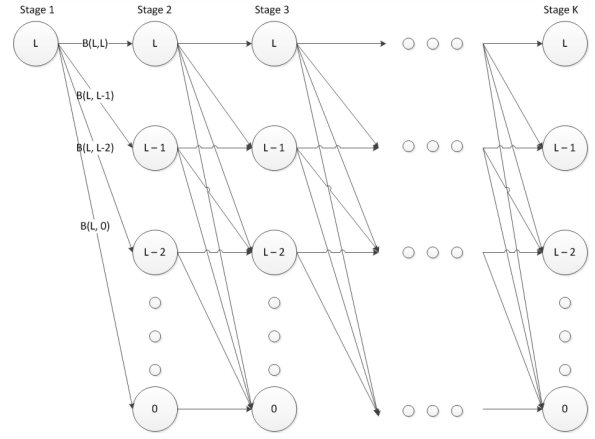


Fig. 2. Retransmission Flow Diagram

for  $x = 1, 2, \dots, K$ . If we have  $n$  successful transmissions, the total time consumed by the transmission attempts is

$$T|n = (X_{1|K} + X_{2|K} + \dots + X_{n|K}) T_s \quad (29)$$

and its expected value is

$$\varepsilon\{T|n\} = \frac{1 - (P_{out_P})^K - K(P_{out_P})^K(1 - P_{out_P})}{(1 - (P_{out_P})^K)(1 - P_{out_P})} n T_s \quad (30)$$

where  $T_s$  is the time to transmit a packet (including the data and overhead). Taking the expectation of (30), we get

$$\varepsilon\{T\} = \frac{1 - (P_{out_P})^K - K(P_{out_P})^K(1 - P_{out_P})}{(1 - P_{out_P})(1 - (P_{out_P})^K)(P_{out_P})^K} T_s. \quad (31)$$

Since the number of successful transmissions is geometrically distributed, its expected value is given by

$$\varepsilon\{n\} = \frac{1}{(P_{out_P})^K}. \quad (32)$$

Consider the period of unsuccessful transmissions. The number of continuous outages,  $n'$  is geometrically distributed with parameter  $P_{out_P}$ . Then,

$$\varepsilon\{n'\} = \frac{1}{1 - (P_{out_P})^K}. \quad (33)$$

Given  $n'$  unsuccessful transmissions, the expected time  $T'$  is

$$\varepsilon\{T'|n'\} = K T_s n' \quad (34)$$

Unconditioning on  $n'$ , we have

$$\varepsilon\{T'\} = \frac{K T_s}{1 - (P_{out_P})^K}. \quad (35)$$

Using (31), (32) and (35) the throughput without the use of the proposed packet recovery mechanism is given by

$$\text{Throughput} = \frac{\varepsilon\{n\} S_p}{\varepsilon\{T\} + \varepsilon\{T'\}} \quad (36)$$

where  $S_p$  is the number of bits in a packet.

### B. Throughput with Proposed Recovery Mechanism

Let  $L$  denote the number of blocks per packet ( $L = \frac{m}{n}$ ). With the proposed recovery mechanism, an outage is an event

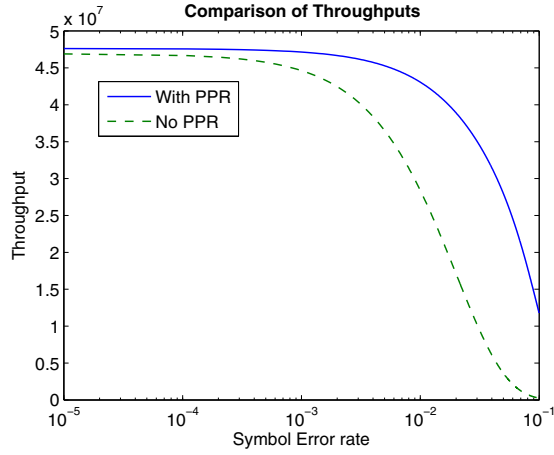


Fig. 3. Comparison of efficiency with and without the proposed packet recovery (PPR) mechanism at 48 Mbps.

where after  $K$  tries at least one block is still undelivered. The probability that  $i$  of  $L$  blocks in a packet are corrupted is

$$B(L, i) = \binom{L}{i} (P_{out_B})^i (1 - P_{out_B})^{L-i} \quad (37)$$

for  $i = 0, 1, \dots, L$ . The transmission of any packet can be considered to proceed in a sequence of stages, as shown in Figure 2. The first stage always involves the transmission of  $L$  blocks. If  $i$  of these  $L$  blocks are corrupted, then the second stage consists of the transmission of  $i$  blocks. The number of blocks transmitted in stage 1 can then be expressed as follows

$$P\{Y_1 = i\} = \begin{cases} 1 & i = L \\ 0 & \text{otherwise} \end{cases} \quad (38)$$

Similarly the number of blocks transmitted in stage 2 is

$$P\{Y_2 = i\} = \begin{cases} B(L, L) & i = L \\ B(L, L-1) & i = L-1 \\ B(L, L-2) & i = L-2 \\ \vdots & \\ B(0, 0) & i = 0 \end{cases} \quad (39)$$

Taking the expectation of (39), we get

$$\varepsilon\{Y_2\} = LP_{out_B}. \quad (40)$$

Proceeding along the same lines and using Figure 2, the number of blocks transmitted in stage 3 is given by

$$P\{Y_3 = j\} = \begin{cases} \sum_{i=j}^L B(i, j)P\{Y_2 = i\} & j = 0, 1, \dots, L \\ 0 & \text{otherwise} \end{cases} \quad (41)$$

The expected value of (41) is given by

$$\varepsilon\{Y_3\} = \sum_{i=0}^L iP_{out_B}P\{Y_2 = i\} = L(P_{out_B})^2. \quad (42)$$

The general form for the number of blocks transmitted in any stage  $n$ , for  $n > 1$  is given by

$$P\{Y_n = j\} = \begin{cases} \sum_{i=j}^L B(i, j)P\{Y_{n-1} = i\} & j = 0, \dots, L \\ 0 & \text{otherwise} \end{cases} \quad (43)$$

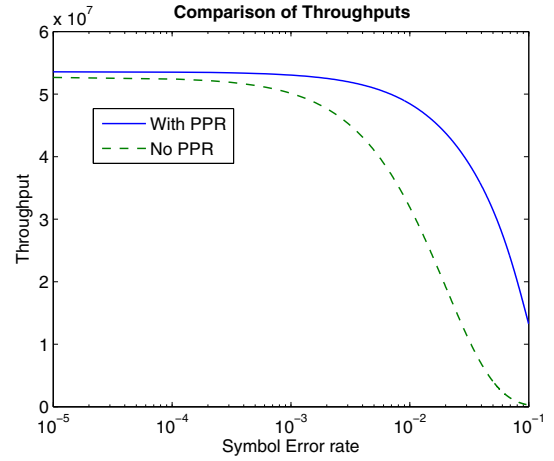


Fig. 4. Comparison of efficiency with and without the proposed packet recovery (PPR) mechanism at 54 Mbps.

and its expected value is given by

$$\varepsilon\{Y_n\} = \sum_{i=0}^L iP_{out_B}P\{Y_{n-1} = i\} = L(P_{out_B})^{n-1}. \quad (44)$$

We can now obtain the total time required to transmit a packet. If we denote the maximum limit of retries by  $K$ , the expected number of blocks transmitted per round is given by

$$\varepsilon\{Y\} = \sum_{i=1}^K \varepsilon\{Y_i\} = \sum_{i=1}^K L(P_{out_B})^{i-1} = L \frac{1 - (P_{out_B})^K}{1 - P_{out_B}}. \quad (45)$$

A packet is dropped if after  $K$  retransmission rounds there is still at least one block missing. If we denote the time to transmit the overhead incurred by each packet by  $C$  and the data rate by  $R$ , the throughput is given by

$$\text{Throughput} = \frac{L \times \text{bits/block} \times P\{Y_K = 0\}}{\left(\frac{L \times \text{bits/block}}{R} + C\right) \frac{1 - (P_{out_B})^K}{1 - P_{out_B}}}. \quad (46)$$

Figures 3 and 4 show the theoretical throughput with and without the proposed packet recovery mechanism at 48Mbps and 54Mbps, respectively. We observe that as the symbol error rate increases, the throughput of our proposed scheme is much higher than the throughput of the case without our scheme.

## VI. RESULTS

In this section we present the simulation results to evaluate the performance of our proposed methodology. The simulations were done using a mix of MATLAB/Simulink and NS2. The transmitter and receiver models were designed according to the IEEE 802.11a specifications [7]. The network simulation was done using NS2, which in turn used the output of the channel model created using MATLAB/Simulink. The wireless nodes in the simulation are connected through a frequency flat multipath Rayleigh fading channel. The channel is realized through the Jake's model [8]. Results were generated for BPSK, QPSK, 16QAM and 64QAM. Due to space limitations, we only show the results for 64QAM, for which two data rates were considered: 48Mbps and 54Mbps. For both cases, there

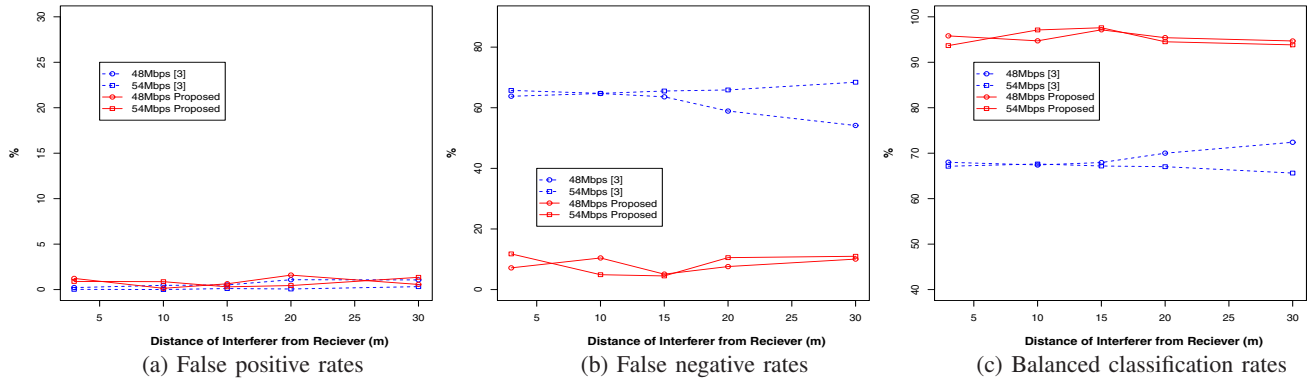


Fig. 5. Comparison of False positive, false negative, and balanced classification rates of the proposed mechanism and the scheme in [2].

were 6 coded bits per subcarrier and 288 coded bits per OFDM symbol. For the 48Mbps case the coding rate was  $2/3$  and data bits per OFDM symbol was 192. The corresponding numbers for the 54Mbps case was  $3/4$  and 216.

### A. Performance of Error Location Detection

We first evaluate the mechanism to identify the blocks in error. We also compare our proposed method with the most accurate error identification scheme in existing literature [2]. The scheme in [2] uses the error vector of each symbol to identify the symbols that are in error. The mechanism proposed in [2] empirically finds the threshold for the error vector.

The performance of the proposed localizing mechanism is evaluated using three metrics: false positive rate (the fraction of error free blocks classified as erroneous), false negative rate (the fraction of erroneous blocks classified as error free) and the Balanced Classification Rate (the average of the number of correct classifications in each class).

Figure 5 compares the false positive, false negative, and balanced classification rates of the proposed error localization mechanism and the method from [2], as a function of the distance between the interferer and the receiver. The proposed scheme outperforms the mechanism in [2] in terms of the false negative and balanced classification rates, and their false positive rates are similar. Note that the accuracy of the mechanism in [2] depends on the training data that is used to set its thresholds. In the simulation results reported here, the training and evaluation scenarios were quite similar, and the good performance is not surprising. In contrast, our mechanism does not depend on any training data and is thus expected to be more robust in a variety of environments. In addition, the mechanism in [2] checks every symbol for an error so encoding the feedback is a significant overhead.

### B. Throughput Performance

Table I shows the percentage improvement in the throughput achieved with and without the proposed packet recovery mechanism for various network sizes. For these simulations, saturated traffic conditions were assumed at each node. All nodes exchanged data with a single base station and the nodes were located on a circle of radius 50 meters around the

TABLE I  
PERCENTAGE IMPROVEMENT IN THE THROUGHPUT ACHIEVED WITH THE PROPOSED PACKET RECOVERY MECHANISM OVER IEEE 802.11

Data Rate	Number of Nodes									
	2	4	6	8	10	12	14	16	18	20
48 Mbps	14.0	13.0	12.7	12.6	12.5	12.4	12.4	12.3	12.3	12.3
54 Mbps	13.9	13.0	12.7	12.5	12.5	12.4	12.4	12.3	12.3	12.3

base station, at regular angular intervals. We observe that the proposed mechanism provides more than 10% improvements in the throughput at all loads. This improvement increases as the distance between the source and the destination nodes increases or the channel becomes poorer.

## VII. CONCLUSIONS

This paper presented a mechanism for the fast and efficient recovery of corrupted packets. The proposed mechanism first divides a packet into blocks and uses the EVM values of the blocks to isolate the errors. The transmitter then retransmits only that portion of a packet which is corrupted. Using both analysis and simulations, it was shown that the throughput of the proposed packet recovery scheme is much higher than that achieved with the conventional IEEE 802.11 protocol.

## REFERENCES

- [1] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [2] K. Jamieson and H. Balakrishnan, "PPR: partial packet recovery for wireless networks," *Proceedings of ACM SIGCOMM*, pp. 409-420, Kyoto, Japan, August 2007.
- [3] G. Woo, P. Kheradpour, D. Shen and D. Katabi, "Beyond the bits: cooperative packet recovery using physical layer information," *Proceedings of ACM MOBICOM*, pp. 147-158, Montreal, Canada, September 2007.
- [4] K. Lin, N. Kushman and D. Katabi, "ZipTx: harnessing partial packets in 802.11 networks," *Proceedings of ACM MOBICOM*, pp. 351-362, New York, NY, September 2008.
- [5] Y. Choo, J. Kim, W. Yang and C. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, Wiley-IEEE press, Singapore, 2010.
- [6] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufman, San Francisco, CA, 2005.
- [7] *High-Speed Physical Layer in the 5 GHz Band*, IEEE Standard 802.11a-1999, 1999.
- [8] W. Jakes, *Microwave Mobile Communications*, New York: Wiley, 1974.