A Wireless MAC Protocol with Efficient Packet Recovery

Muhammad Naveed Aman and Biplab Sikdar
Department of Electrical, Computer and Systems Engineering,
Rensselaer Polytechnic Institute, Troy NY 12180 USA

***Abstract*—Existing wireless medium access control (MAC) protocols provide reliability against corrupted packets by providing mechanisms for packet error detection and retransmission. The efficiency of existing mechanisms for providing reliability is usually 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 a MAC protocol with an efficient packet recovery mechanism for packets corrupted due to both channel errors and collisions. The proposed MAC protocol first determines the cause of the errors in a packet and then uses the acknowledgment (ACK) packets to provide feedback on the sections of the packets that have errors. To minimize the packet recovery time, the proposed MAC protocol allows the sender to retransmit the corrupted sections of the packet immediately, without requiring a new channel access. Using simulations, it is shown that the proposed MAC protocol has higher efficiency and increases the achieved throughput.**

I. Introduction

The causes of errors in wireless networks may be broadly classified as either (i) caused by poor channel conditions resulting from factors such as fading, path loss and noise and (ii) those caused by collisions resulting from simultaneous transmissions from more than one node. The MAC layer of most wireless networks provide reliability against these errors by including mechanisms to detect the presence of errors and retransmitting the corrupted packets. A fundamental limitation with existing protocols that limits the achieved throughput is that the retransmission mechanisms usually resend the entire packet, even though only parts of it may have been corrupted. Since most errors are confined to parts of a packet, retransmitting the entire packet leads to waste of channel bandwidth and is an overhead that should be avoided. To address this issue, this paper presents a wireless MAC protocol capable of efficient packet recover in the presence of errors that may be caused by the channel or collisions.

The presence of errors in a received packet is usually detected through the use of error detection codes, such as the cyclic redundancy check (CRC) used in IEEE 802.11 [1]. Mechanisms such as automatic repeat request (ARQ) are then used to retransmit the corrupted packet [2]. Since CRCs cannot isolate the bits with errors, most ARQ schemes retransmit the entire packet. A methodology for marking the bits that are likely in error is presented in [3] 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 [4]. The overhead and delay associated with combining agents limits the application of this scheme. Moreover, only AWGN channels are considered in [4] and its effectiveness in more practical channels has not been established. In [5], 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.

978-1-4799-4894-9/14/$31.00 © 2014 IEEE.

The main contribution of this paper is a MAC protocol that efficiently recovers packets that are corrupted during transmission. The proposed protocol is based on the use of our mechanism to isolate the cause and location of errors in transmitted packet. The proposed methodology to localize the errors is based on calculating the Error Vector Magnitude (EVM) of the received symbol. The receiver in the proposed protocol first detects the symbols that were corrupted during transmission. It then conveys the locations of the corrupted portions of the packet to the transmitter using the ACK packet and the transmitter can then uses precise retransmissions of only the corrupted symbols. The performance of the proposed MAC protocol has been evaluated using both analysis and simulations.

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

II. A MAC Protocol with Efficient Packet Recovery

The proposed MAC protocols uses carrier sense multiple access with collision avoidance (CSMA/CA) for channel arbitration and its basic operation is based on IEEE 802.11 [1]. The proposed protocol may be considered to an extension for IEEE 802.11 for efficient packet recovery. The operation of the protocol consists of two steps: (i) when a packet is deemed to be corrupt (as determined by a CRC check), the first step is to detect the sections of the packet that are corrupted, and (ii) facilitate the efficient and targeted retransmission of only the

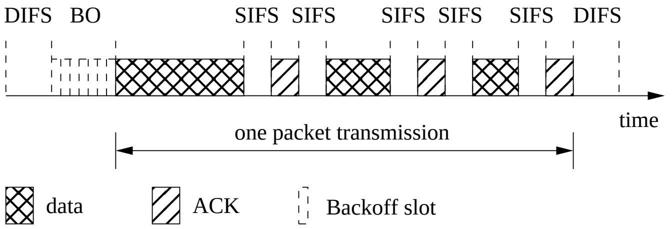

Fig. 1. Proposed MAC Protocol for Efficient Packet Recovery

corrupted sections of the packet.

A. Error Localization

A key part of the packet recovery mechanism is the mechanism that localizes the sections of the packet that are corrupted. For the purposes of error localization, we consider each packet to be divided into a number of equal sized blocks. For each corrupted packet, 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.

B. MAC Protocol

The operation of the proposed MAC protocol is shown in Figure 1. Under CSMA/CA based channel access, each node with a packet to transmit first senses the state of the channel. If the channel is sensed to be idle for an interval greater than the Distributed Inter-Frame Space (DIFS), the node proceeds with its transmission. If the channel is busy, the node defers its transmission till the channel becomes idle. The node then initializes its *backoff timer* with a randomly selected *backoff interval* and decrements this timer every time it senses the channel to be idle. The node transmits its data when the backoff timer reaches zero. An exponential backoff mechanism such as that in IEEE 802.11 is used in the presence of collisions. After receiving a packet, the receiver waits for a Short Inter-Frame Space (SIFS), before it transmits an ACK. However, unlike IEEE 802.11 that only uses positive ACKs, the proposed MAC protocol uses the ACK to convey information about the sections of the packet that are corrupted. Once the blocks in error are identified, a bitmap indicating these blocks is sent to the transmitter using the ACK packet.

On receiving the block-level error bitmap, the sender proceeds to retransmit the packet but only includes the blocks with errors. A key aspect of the proposed protocol is that the retransmissions do not require a new channel access. Once the ACK is received, the sender retransmits the corrupted blocks immediately after a SIFS period. Since all other nodes in the vicinity of the sender have to wait for a DIFS period before they can begin their transmission or backoff, this prevents any collisions. The receiver replies to the retransmitted blocks with an ACK that again uses a bitmap to indicate the blocks that may still be in error. The sender again retransmits the corrupted blocks after a SIFS period and this continues till all the blocks of the packet are received correctly. The MAC layer at the receiver assembles all the blocks of the packet before sending it to the upper layer. An upper limit may be placed on the number of retransmission attempts that are allowed for a block before the current transmission is aborted. Such a limit may be required for fairness and to prevent a single node from monopolizing the channel for too long. The entire packet needs to be retransmitted if the current transmission attempt exceeds the number of retransmissions attempts for its blocks in a single channel access.

The advantage of the proposed protocol stems from three factors. *First*, only the sections of the packet that are corrupted are retransmitted, thereby eliminating the waste of channel resources. *Secondly* and more significantly, the retransmission does not require a separate channel accesses. The channel access time is a major contributor to the delays experienced by the packets [6]. By allowing retransmissions after only a SIFS period as opposed to a full-fledged channel access involving backoffs and deferments to transmissions from other nodes. Finally, the *third* advantage of the proposed MAC protocols is that since it can identify the cause of a corrupted packet, a node does not have to do an exponential backoff in the cases where the errors are caused by channel errors.

1) Extension for RTS/CTS: the description of the protocol above illustrated the operation of the protocol in the absence of request-to-send (RTS) and clear-to-send (CTS) packets. RTS-CTS packet exchange before the data transmission are usually recommended to address the problem of hidden terminals. Under the operation with RTS-CTS based reservation, a node with packet to transmit sends a RTS packet to the receiver (after successful channel access using CSMA/CA) and the receiver responds with a CTS packet if it is willing to accept the packet and is currently not busy. The RTS/CTS packets contain timing information about the length of the ensuing transaction, and all nodes that overhear this exchange defer their transmissions till the current transmission is complete.

With the proposed retransmission mechanism, the timing information contained in the RTS/CTS packets may not be valid if retransmissions are required for any of the blocks of the packet. Consequently it is possible that hidden nodes may interfere with subsequent transmissions. To address this situation, the proposed MAC protocol adds timing information to the ACK packets. When an ACK packet sends a bitmap with the list of blocks with errors, it also needs to include the timing information that the retransmission will take. Nodes overhearing the ACK packet will update their network allocation vector (NAV) and defer their transmission. At the sender's side, no modification is required since the retransmission begins after a SIFS interval while all other nodes have a wait of at least a DIFS interval. The MAC header of the retransmitted packet can also convey the timing information required for the nodes in the vicinity of the sender.

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. 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

$$y_n = H_n * x_n + \eta_n + \zeta_n \tag{1}$$

where H_n is the Rayleigh channel coefficient, η_n is additive white Gaussian noise with zero-mean and variance σ_η^2, and ζ_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^{th} time domain OFDM signal and can be obtained from X_k, the M-QAM modulated symbol at the kth subcarrier as [7],

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

Let us introduce a random variable Z defined as

$$Z = \frac{1}{N \cdot P_0} \sum_{k=0}^{N-1} |E_k|^2. \tag{3}$$

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 (3) as

$$Z = \frac{1}{N \cdot P_0} \sum_{n=0}^{N-1} |e_n|^2. \tag{4}$$

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)

$$f_Z(z) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{-\frac{(z-\mu_z)^2}{2\sigma_Z^2}}. \tag{5}$$

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

$$e_n = H_n x_n + \eta_n - x_n = x_n(H_n - 1) + \eta_n. \tag{6}$$

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

$$\mu_Z \;=\; \varepsilon\{Z\} = \frac{1}{P_0}\left[\sigma_x^2\left(\frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}}\right) + \sigma_\eta^2\right]. \tag{7}$$

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

$$Z^2 = \left(\frac{1}{NP_0}\sum_{n=0}^{N-1}|e_n|^2\right)^2 = \frac{1}{N^2P_0^2}\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}|e_{n_1}|^2|e_{n_2}|^2. \tag{8}$$

We now assume block fading with a block length of m symbols. Then, $\varepsilon\{Z^2\}$ is given by

$$\varepsilon\{Z^2\} = \frac{1}{NP_0^2}\left[m\varepsilon\{e_n^4\} + \frac{N-m}{m}\left(\varepsilon\{e_n^2\}\right)^2\right] \tag{11}$$

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

$$\varepsilon\{e_n^2\} = \sigma_x^2\left\{\frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}}\right\} + \sigma_\eta^2 \tag{12}$$

and $\varepsilon\{e_n^4\}$ is given by (9). Combining (7) and (11), 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 $N-1$. If a collision starts at symbol n_0, the error vector is given by

$$e_n^* = \begin{cases} e_n & n < n_0 \\ e_n + \zeta_J & n \geq n_0 \end{cases} \tag{13}$$

where ζ_J is given as follows

$$\zeta_J = \sum_{i=0}^{J} B_i H_{r_i} W_i. \tag{14}$$

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 σ_x^2. We can then rewrite (4) as follows:

$$Z^* = \frac{1}{NP_0}\left(\sum_{n=0}^{n_0-1}|e_n|^2 + \sum_{n=n_0}^{N-1}\left(|e_n| + \zeta_J\right)^2\right). \tag{15}$$

Taking the expectation of (15), 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

$$\begin{aligned}\varepsilon\{{Z^*}^2\} = \frac{1}{N^2P_0^2}\Bigg[&\left(\frac{N-1}{2}\right)m\varepsilon\{e_n^4\} + \Bigg(\frac{1}{4}\frac{(N-1)^3}{m} - \\ &\frac{2}{3}(N-1)^2 + \frac{1}{2}(N-1)m\Bigg)\left(\varepsilon\{e_n^2\}\right)^2 + (N-1)\frac{N+2}{3} \\ &\varepsilon\{e_n^2\}\varepsilon\{{e_n^*}^2\} + \frac{(N+1)}{2}m\varepsilon\{{e_n^*}^4\} + \Bigg[\frac{1}{4m}N\left(N^2+5N-3\right) \\ &-\frac{1}{3}\left(5N^2-N+2\right) + \frac{1}{2}m(N+1) + \frac{1}{4m}\Bigg]\left(\varepsilon\{{e_n^*}^2\}\right)^2\Bigg]\end{aligned} \tag{16}$$

where $\varepsilon\{e_n^2\}$ and $\varepsilon\{e_n^4\}$ are given by (12) and (9). respectively. $\varepsilon\{{e_n^*}^2\}$ is given by

$$\varepsilon\{{e_n^*}^2\} = \sigma_x^2\left\{\frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}}\right\} + \sigma_\eta^2 + 2p\sigma_x^2\sum_{j=0}^{J}\frac{1}{r_j^\alpha} \tag{17}$$

and $\varepsilon\{{e_n^*}^4\}$ is given by (10). Thus, we can now obtain the

$$\begin{aligned}\varepsilon\{e_n^4\} &= (3\sigma_x^4)\left(\frac{8}{r_i^{2\alpha}} - \frac{4}{r_i^{\frac{3}{2}\alpha}} + \frac{12}{r_i^{\alpha}} - 2\sqrt{\frac{2\pi}{r_i^{\alpha}}} + 1\right) + 3\sigma_\eta^2 + 6\sigma_x^2\sigma_\eta^2\left(\frac{2}{r_i^{\alpha}} - \sqrt{\frac{2\pi}{r_i^{\alpha}}} + 1\right) \qquad (9)\\ \varepsilon\{e_n^{*4}\} &= \left[(3\sigma_x^4)\left(\frac{8}{r_i^{2\alpha}} - \frac{4}{r_i^{\frac{3}{2}\alpha}} + \frac{12}{r_i^{\alpha}} - 2\sqrt{\frac{2\pi}{r_i^{\alpha}}} + 1\right) + 3\sigma_\eta^2 + 6\sigma_x^2\sigma_\eta^2\left(\frac{2}{r_i^{\alpha}} - \sqrt{\frac{2\pi}{r_i^{\alpha}}} + 1\right)\right] + \\ &\quad 6\left[\sigma_x^2\left\{\frac{2}{r_i^{\alpha}} + 1 - 2\sqrt{\frac{\pi}{2r_i^{\alpha}}}\right\} + \sigma_\eta^2\right]\left[2p\sigma_x^2\sum_{j=0}^{J}\frac{1}{r_j^{\alpha}}\right] + 6p\sigma_x^4\sum_{j=0}^{J}\frac{1}{r_j^{2\alpha}} \qquad (10)\end{aligned}$$

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 γ, then the probability of a false positive is given by

$$P_Z[Z > \gamma] = P_e\left[1 - \int_{-\infty}^{\gamma} f_Z(z)\,dz\right] \qquad (18)$$

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

$$P_{Z^*}[Z^* \le \gamma] = P_e\left[\int_{-\infty}^{\gamma} f_{Z^*}(z^*)\,dz\right]. \qquad (19)$$

To obtain the threshold that leads to the crossover error rate, we can find γ by equating (18) and (19). Thus, to get the threshold we need to solve the following equation numerically:

$$\int_{-\infty}^{\gamma} f_Z(z)\,dz + \int_{-\infty}^{\gamma} f_{Z^*}(z^*)\,dz = 1. \qquad (20)$$

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 ($m \ge 10$). This block based approach serves two purposes: (i) justifying the use of (5) 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.

B. Error Locations in Packets Corrupted by Channel Noise

To detect the error locations in case of corruption due to channel noise, we perform K-means clustering [8] 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. The process starts with two points randomly chosen as cluster centers. The EVM 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. 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 [9]. 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 [10]. Results were generated for BPSK, QPSK, 16QAM and 64QAM. We only show the results for 64QAM, for which two data rates were considered: 48Mbps and 54Mbps. For both cases, there 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.

Figures 2 and 3 show the throughput achieved by the proposed MAC protocol and IEEE 802.11 for channel data rates of 48 and 54 Mbps respectively, for a channel signal to noise ratio (SNR) of 40 dB. 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 base station, at regular angular intervals. We observe that the proposed mechanism

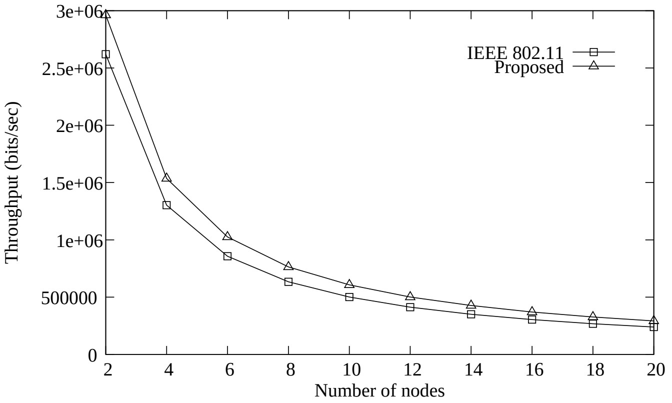

Fig. 2. Comparison of throughput of the proposed packet recovery mechanism and IEEE 802.11 at 48 Mbps and SNR of 40dB.

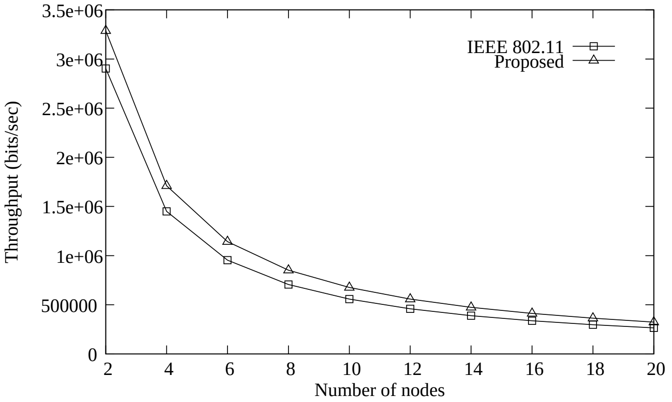

Fig. 3. Comparison of throughput of the proposed packet recovery mechanism and IEEE 802.11 at 54 Mbps and SNR of 40dB.

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.

VI. Conclusions

This paper presented a MAC protocol for the fast and efficient recovery of corrupted packets. The proposed MAC protocol is based on first determining the cause of the errors in a packet and then selectively retransmitting the sections of the packet with losses. Additional efficiency is achieved by the proposed MAC protocol by allowing the sender to retransmit the corrupted sections of the packet immediately, without requiring a new channel access. Our simulation results show that the proposed MAC protocol performs significantly better than the conventional IEEE 802.11 protocol.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE Standard 802.11, 1997.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1987.

[3] K. Jamieson and H. Balakrishnan, “PPR: partial packet recovery for wireless networks,” *Proceedings of ACM SIGCOMM,* pp. 409-420, Kyoto, Japan, August 2007.

[4] 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.

[5] 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.

[6] O. Tickoo abd B. Sikdar, “A Queueing Model for Finite Load IEEE 802.11 Random Access MAC,” *Proceedings of IEEE ICC*, pp. 175-179, Paris, France, June 2004.

[7] Y. Choo, J. Kim, W. Yang and C. Kang, *MIMO-OFDM Wireless Communications with MATLAB,* Wiley-IEEE press, Singapore, 2010.

[8] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* Morgan Kaufman, San Francisco, CA, 2005.

[9] *High-Speed Physical Layer in the 5 GHz Band,* IEEE Standard 802.11a-1999, 1999.

[10] W. Jakes, *Microwave Mobile Communications,* New York: Wiley, 1974.