

IEEE 802.11 Performance Enhancement via Concatenation and Piggyback Mechanisms

Yang Xiao, *Senior Member, IEEE*

Abstract—The IEEE 802.11 medium access control (MAC) is a very robust protocol for the best effort service in the wireless medium. However, many studies have reported that it is not very efficient. One of the fundamental problems of MAC inefficiency is overhead. In this paper, we propose two novel mechanisms to reduce overhead of the IEEE 802.11 protocols: 1) concatenation mechanism (CM); and 2) piggyback mechanism (PM). Performance analysis is conducted under both the best-case scenario and the saturation scenario. Studies show that both proposed schemes have greatly improved the system performance.

Index Terms—Concatenation, IEEE 802.11, medium access control, performance analysis, piggyback.

I. INTRODUCTION

WIRELESS local area networks (LANs) have quickly found a significant niche as the cost decreases and the data rate becomes higher. The IEEE 802.11, 802.11b, and 802.11a specifications provide up to 2, 11, and 54 Mb/s data rates [1]–[3], respectively, whereas the industry is seeking data rates over 100 Mb/s [4]. However, the medium access control (MAC) protocol that they are based upon is not very efficient. Many studies have been proposed to improve performance by finely tuning MAC to obtain marginal improvement. To dramatically improve performance, the fundamental problems of MAC inefficiency need to be figured out. In [4], the authors have reported that a throughput upper limit (TUL) exists even when the data rate goes infinitely high. This indicates that overhead (headers, interframe spaces, backoff time, and acknowledgments) is one of the fundamental problems of MAC inefficiency [4]. The inefficiency becomes much more severe when the data rate becomes much higher [5].

In this paper, we propose two novel mechanisms—a concatenation mechanism (CM) and a piggyback mechanism (PM)—to reduce overhead and to improve performance. The idea of the CM is to concatenate multiple frames in a station's queue, if available, into a single transmission, which is not one real frame transmission but multiple frame transmissions concatenated together one by one. The total length of the concatenated frames should be smaller than a threshold, called concatenation threshold. If there are no multiple frames available, the CM will not be used and normal distributed coordination function (DCF) procedures will be used by default. The concatenated frames are expected to be relatively small. Examples for avail-

able frames can be formed from short UDP packets, real-time voice/audio packets, short request-reply client-server packets, and transmission control protocol-ACK (TCP-ACK) packets, etc. One small frame can also be concatenated with a relatively large frame, e.g., a TCP-ACK packet follows a relatively large frame. The second proposed mechanism is the PM, in which a receiver station is allowed to piggyback a data frame to a sender station once if the receiver station has a frame to send to the sender. From the MAC point of view, the MAC layer does not care what its upper layer is [Internet protocol (IP) or others later on] but provides two good optional mechanisms that are useful. The proposed mechanisms are especially useful for the IEEE 802.11n, established in September 2003, emphasizing higher throughput in the next-generation wireless LANs.

There have been many performance studies for the original IEEE 802.11 MAC [4]–[15]. Bianchi [6] proposed a simple and accurate analytical model to compute saturation throughput. Ziouva and Antonakopoulos (ZA) [7] improved Bianchi's model to derive saturation delay. Wu *et al.* [8] improved Bianchi's model to consider retry limit. Xiao [9], [16] proposed an analytical model for priority schemes.

In this paper, performance analysis for the proposed schemes is conducted under both the best case scenario and the saturation scenario. At the best-case scenario, performance is analyzed in terms of the maximum throughput (MT), the minimum delay (MD), the TUL, and the delay lower limit (DLL). At the saturation scenario, we evaluate the CM and the PM in terms of saturation throughput and saturation delay with an improved analytical model that is more consistent with the standard.

This paper is organized as follows. An overhead analysis is provided in Section II, indicating that overhead magnifies itself when many short frames exist, or when the data rate becomes much higher for next-generation wireless LANs. We propose the CM in Section III, and the CM's benefits are summarized. The proposed PM is presented in Section IV. Performance studies at the best case scenario are conducted in Section V. Saturation performance is studied in Section VI. We conduct simulation validations in Section VII. Numerical results are presented in Section VIII. We conclude this paper in Section IX.

II. OVERHEAD ANALYSIS

IEEE 802.11 employs a carrier sense multiple access with collision avoidance MAC protocol with binary exponential backoff [1]. As illustrated in Fig. 1, a station with a frame to transmit monitors the channel activities until an idle period equal to a distributed interframe space (DIFS) is detected. After sensing an idle DIFS, the station waits for a random backoff

Manuscript received November 26, 2002; revised February 14, 2004; accepted June 21, 2004. The editor coordinating the review of this paper and approving it for publication is B. Li.

The author is with the Department of Computer Science, University of Memphis, Memphis, TN 38152 USA (e-mail: yangxiao@ieee.org).

Digital Object Identifier 10.1109/TWC.2005.853875

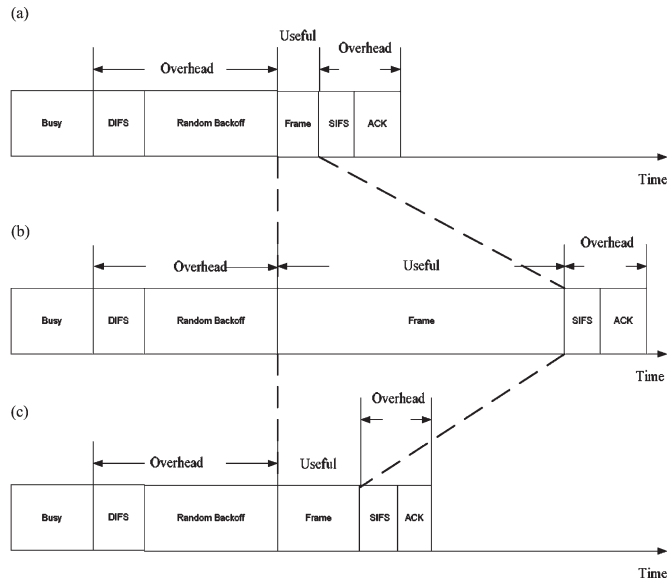


Fig. 1. IEEE 802.11 DCF overhead. (a) Short frame transmission. (b) Long frame transmission. (c) Long frame transmission over higher data rate.

interval before transmitting. The backoff time counter is decremented in terms of slot time as long as the channel is sensed idle. The counter is stopped when a transmission is detected on the channel and reactivated when the channel is sensed idle again for more than a DIFS. The station transmits its frame when the backoff time reaches zero. At each transmission, the backoff time is uniformly chosen in the range $[0, CW - 1]$, where CW is the current backoff window size. At the very first transmission attempt, CW equals the minimum backoff window size CW_{min} . After each unsuccessful transmission, CW is doubled until a maximum backoff window size value CW_{max} is reached. Once it reaches CW_{max} , CW shall remain at the value CW_{max} until it is reset. CW shall be reset to CW_{min} after every successful attempt to transmit or the retransmission counter reaches the retry limit L_{retry} . In the latter case, the frame will be dropped. After the destination station successfully receives the frame, it transmits an acknowledgment frame (ACK) following a short interframe space (SIFS) time. If the transmitting station does not receive the ACK within a specified ACK timeout, or it detects the transmission of a different frame on the channel, it reschedules the frame transmission according to the previous backoff rules. In the optional request-to-send (RTS)/clear-to-send (CTS) mode, before transmitting a data frame, a short RTS frame is transmitted. If the RTS frame succeeds, the receiver station responds with a short CTS frame. Then, a data frame and an ACK frame will follow. All four frames (RTS, CTS, data, ACK) are separated by an SIFS time.

Fig. 1 shows virtual successful transmission times of a short frame, a long frame, and a long frame over a higher data rate, respectively. Overhead is one of the fundamental problems of MAC inefficiency, and it includes headers [MAC header, frame check sequence (FCS), and physical (PHY) header that are not shown in the figure], interframe spaces (DIFS and SIFS), backoff time, and acknowledgments. When a short frame is transmitted [Fig. 1(a)], frame transmission time is relatively small if compared to overhead time, and throughput

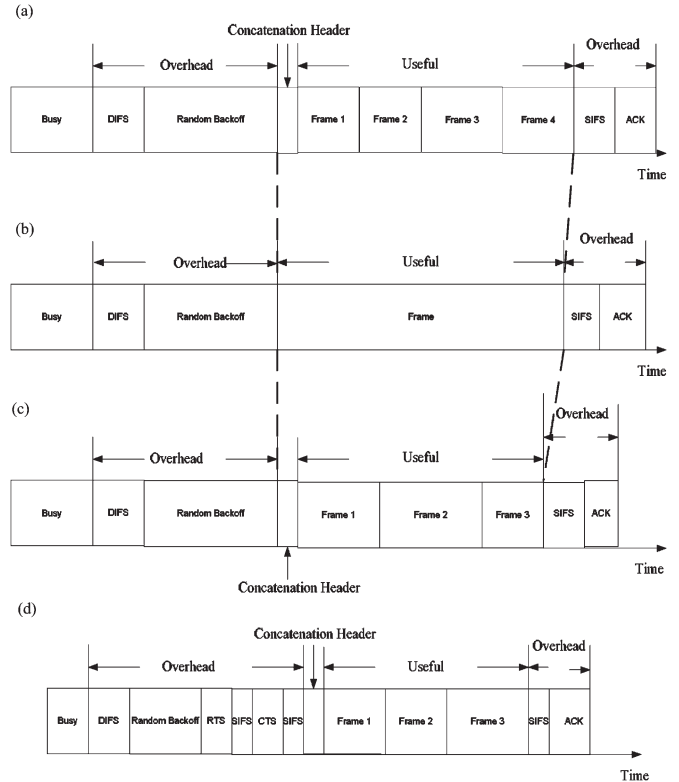


Fig. 2. Concatenation of multiple frames (figure is not drawn on scale). (a) Concatenating multiple short frames. (b) Long frame transmission. (c) Concatenating multiple long frame over higher data rate. (d) Concatenation for the RTS/CTS.

will be small. When a long frame is transmitted [Fig. 1(b)], compared to overhead time, frame transmission time is relatively large, and throughput is improved. Note that it is sometimes difficult to control payload size since it comes from higher layers, e.g., frames from a streaming media such as voice frames that are very small. For next-generation wireless LANs, the data rate will become much higher [5], and even when a long frame is transmitted [Fig. 1(c)], frame transmission time is relatively small, whereas most of the overhead time still remains. Therefore, the normalized throughput decreases. In summary, when many short frames exist, or when the data rate becomes much higher for next-generation wireless LANs, overhead time becomes relatively large, and the system will show relatively poor throughput.

III. CONCATENATION MECHANISM

The idea of a CM is to concatenate multiple frames into a single transmission. Frames can be concatenated if they are available, and they have the same source and destination addresses. Fig. 2(a) shows that four frames are concatenated into a large virtual frame. Fig. 2(c) shows that three frames are concatenated. Fig. 2(d) shows that three frames are concatenated in the RTS/CTS mode. Note that the virtual frame of the CM is not a real frame but a burst of concatenated real frames.

The CM has many benefits. First, since transmitting longer frames may have a better throughput than transmitting shorter frames, adopting the CM, the system can achieve the throughput of transmitting longer frames. The second and the most

important benefit is that the CM can reduce overhead. Without the CM, each frame transmission needs a separate set of overhead (headers, interframe spaces, backoff time, and acknowledgments). With the CM, instead of several sets of overhead for different frames, only one set of overhead will be used. Finally, the CM can reduce the average delay. Without the CM, the second or a later concatenated frame in a virtual frame is transmitted at a much later time. With the CM, it will be transmitted almost at the same time as the first frame is transmitted. The tradeoff is that the processing delay will increase a little.

A large virtual frame includes a concatenation header (CH) frame and concatenated frames. In the CH, the frame control-type field indicates that it is a concatenation virtual frame, and the payload includes the count of concatenated frames (2 B, where B = bytes) and a total length field (2 B). After the destination station receives the CH frame, it will receive the followed concatenated frames one by one and acknowledges the last concatenated frame only. The destination station can easily identify boundaries of concatenated frames using preambles and FCS.

A good processing time to concatenate frames is an idle time such as backoff time. Some frames could be concatenated beforehand, but most of the time, frames must be concatenated at real time when frames are in the queue waiting for transmission. The virtual frame format of the CM is designed in such a way that the processing time of combining and decomposing frames is insignificant since concatenated frames are transmitted one by one separately and received one by one separately. Note that it could be designed differently and more efficiently, sacrificing complexity and processing time of combining and decomposing frames, e.g., combining all concatenated frames into a real big frame with one header instead of many concatenated frames. One may suggest having large frames in an upper layer. However, the MAC layer cannot control what happens in upper layers, whereas, due to long delay of MAC contentions, many originally separated packets of the application layer in time may be queued together in an MAC queue. The virtual frame of the CM is a not a real frame but a virtual frame consisting of multiple real frames. In other words, concatenated frames are not copied together to become a frame. Instead, concatenated frames are transmitted one by one separately and received one by one separately with both the MAC header and the PHY header without any extra effort. Note also that the CH is not a field but a real frame.

The next question is how big a virtual frame should be. One possible solution is that the number of concatenated frames should not be larger than a threshold (such as 2, 3, or 4), and the total length of the virtual frame should be smaller than another threshold, called the concatenation threshold, which is smaller than or equal to the fragmentation threshold. For example, the threshold can be 1000 B. The purpose of the CM is not to build a huge virtual frame but a reasonable size of the virtual frame since huge virtual frames may cause a bad effect on fairness, and when collided, a longer virtual frame is lost. Note that for IEEE 802.11a, the length field of the physical layer convergence protocol (PLCP) frame can indicate less value than 4096 B, and the maximum size of the MAC frame is generally 2346 B [3]. The concatenation threshold should

be much smaller than 2346 B. Our results indicate that from the collision aspect under a clear channel condition, the concatenation threshold could be as large as 1500 B without affecting throughput and delay. Therefore, we recommend that the concatenation threshold is chosen as $\min(\alpha, \beta, \theta)$, where $\alpha = 1500$ B is the Ethernet frame size, β is the fragmentation threshold, and θ is the measured average frame size among relatively large frames during a measurement interval such as a beacon interval.

Effects of the CM on fairness are discussed as follows. With an appropriate concatenation threshold defined as the one above, fairness with the CM may be improved instead of degraded since many smaller frames may be concatenated into relatively large frames so that the overall fairness may be improved. For example, assume that there are only two stations in the system: Station A keeps transmitting long frames (all with 1000 B), and station B keeps transmitting short frames (all with 100 B). It is clear that it is unfair for station B. If all short frames can be concatenated and the concatenation threshold is set to be 1000 B, the overall fairness will be improved.

Effects of wireless channel error rate on the CM are discussed as follows. A larger frame has difficulties to go through the channel when channel bit error rate (BER) is very high. Therefore, there is a tradeoff for the frame size (or concatenation threshold): A larger frame has a better throughput in a better channel condition but a worse throughput in a worse channel condition with much interference and noise. The ideal approach is to obtain an optimal frame size (or concatenation threshold) based on the channel error condition. However, 1) to obtain a realistic channel error condition is still an open issue; 2) BER should consider the path from a source to its destination, instead of in either the source or the destination; and 3) a station cannot know its BER by sending frames but may get some knowledge by receiving an error frame, in which the BER is involved by a path from a specific source to this station at a specific time. To the best of our knowledge, there is no realistic approach to obtain accurate BER in a real system. Therefore, how to obtain either an optimal frame size for the original MAC or an optimal concatenation threshold for our proposed CM is still an open issue, and there is no easy and realistic solution that can be used in reality. Instead, in this paper, we recommend that the concatenation threshold not be larger than the measured average frame size among relatively large frames during a measurement interval. In other words, we believe that as long as the threshold is not larger than other frames on average, it will be alright. This approach is a very realistic one.

The CM is not a reversed mechanism of fragmentation. In fact, the proposed CM requires that the total length of the concatenated frames is less than the fragmentation threshold. Therefore, there will be no concatenated frame that was originally generated by a previous fragmentation mechanism. On the other hand, the concatenated virtual frame will not be fragmented since the total length is less than the fragmentation threshold. Furthermore, concatenated frames are not one frame but multiple frames.

The CM can be implemented in both stations and access points in which a queue is implemented so that when higher

layer data arrive and cannot be sent immediately, the waiting frames are put into the queue. Such a scenario happens since the frame in the head of the queue often experiences backoff, collisions, and waiting for an ACK. Examples for available frames can be formed from a short user datagram protocol (UDP) packets, real-time voice/audio packets, TCP-ACK packets, and so on. The above-mentioned packets are all small in size, and therefore, good throughputs are expected by using the CM. One small frame can also be concatenated with a relatively large frame, e.g., a TCP-ACK packet follows a relatively large frame. In case that there are no available frames in the queue (in other words, concatenation rarely occurs), normal DCF procedures will be used by default, and the system will have the same performance as before.

When the RTS/CTS mode is used, the RTS threshold (defined as dot11RTSThreshold [1]) is turned on, i.e., if the total length of a frame is larger the RTS threshold, the RTS/CTS mode is automatically used. We could use the total length of the virtual frame as the size of a frame to use the RTS/CTS mode, as shown in Fig. 2.

Since the CM does not change any system behavior such as the backoff behavior except that frames become larger, it is easy to see that the CM should have a better performance intuitively.

IV. PIGGYBACK MECHANISM

The idea of the PM is that a receiver station is allowed to piggyback a data frame to the sender station once if the receiver station has a frame to send to the sender. Such a mechanism is beneficial for cases when the receiver has a frame planned to send to the sender station so that the piggybacked frame does not need to compete the channel again. The piggybacked frame does not need to be one in the front of the queue but the nearest frame to the front with the destination to the sender. The overall performance will be improved greatly.

Fig. 3(a) and (b) shows overhead with/without piggyback for the basic access mode. As illustrated in Fig. 3(a), if the destination has a frame to send to the source after receiving a frame, it needs to compete the channel again by at least a DIFS time and a random backoff. On the other hand, in Fig. 3(b), the destination can piggyback a frame to the source with the ACK information included. The source then sends an ACK to acknowledge the piggybacked frame after an SIFS time. The benefit of the PM is that the piggybacked frame does not need to wait for DIFS and to complete the channel again.

Fig. 3(a) and (b) indicates that piggyback will reduce overhead and improve performance. If there is no frame available to piggyback, similar to the CM, normal DCF operations will be performed by default, and the system will have the same performance as before.

Fig. 3(c) and (d) shows overhead with/without piggyback for the RTS/CTS mechanism. As illustrated in Fig. 3(c), if the destination has a frame to send to the source after receiving a frame, it needs to complete the channel again by at least a CTS frame time, an RTS frame time, two SIFS times, a DIFS time, and a random backoff. On the other hand, in Fig. 3(d), the destination can piggyback a frame to the source with the

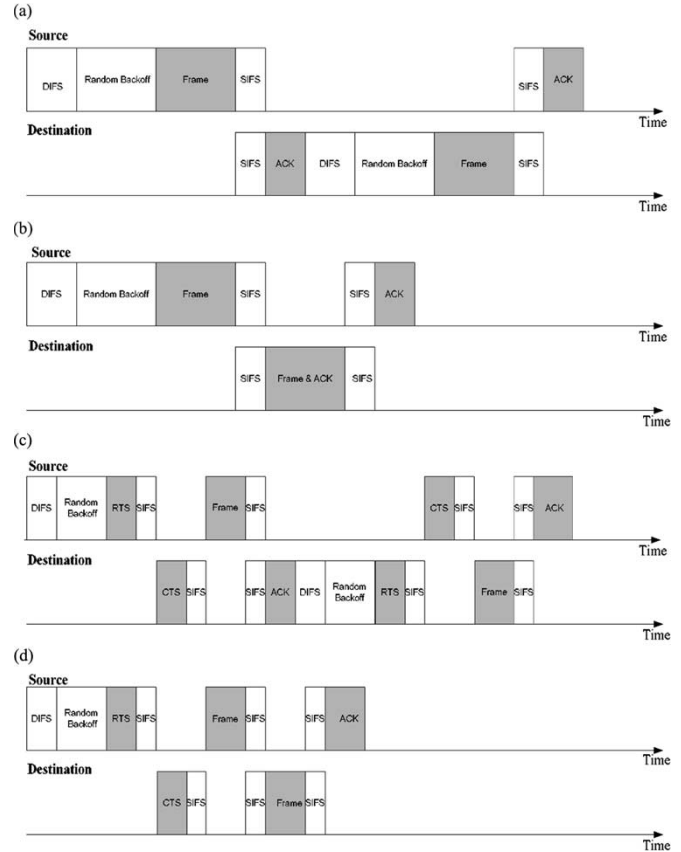


Fig. 3. Overhead with/without piggyback (figure is not drawn on scale). (a) Basic access mode without piggyback. (b) Basic access mode with piggyback. (c) RTS/CTS mode without piggyback. (d) RTS/CTS mode with piggyback.

ACK information included. The source then sends an ACK to acknowledge the piggybacked frame after an SIFS time. The overhead is reduced a lot.

V. PERFORMANCE ANALYSIS AT THE BEST-CASE SCENARIO

MT and MD can be achieved when the system is at the best case scenario [4]: 1) The channel is an ideal channel without errors; 2) at any transmission cycle, there is one and only one active station that always has a frame to send, and other stations can only accept frames and provide acknowledgments. TUL (DLL) is the MT (MD) when the data rate goes infinitely high [4].

Let T_{slot} , T_{SIFS} , T_{DIFS} , and CW_{min} denote a slot time, an SIFS time, a DIFS time, and the minimum backoff window size, respectively. Let T_{P} , T_{PHY} , and T_{SYM} denote transmission times of a physical preamble, a PHY header, and a symbol, respectively. Let $L_{\text{H_DATA}}$, L_{ACK} , and L_{DATA} denote lengths of MAC overhead, an ACK, and the payload, respectively, in bytes. Let T_{DATA} and T_{ACK} denote transmission times of a data frame and an ACK, respectively. Let N_{DBPS} denote data bits per orthogonal frequency-division multiplexing (OFDM) symbol. Let $\bar{CW} = (CW_{\text{min}} - 1)T_{\text{slot}}/2$ denote the average backoff window size. For IEEE 802.11a, MT, MD, TUL, and DLL for the original DCF are given approximately as follows [4],

where $T_{\text{DATA}} = F(L_{\text{H_DATA}} + L_{\text{DATA}})$, $T_{\text{ACK}} = F(L_{\text{ACK}})$, $T_{\text{PY}} \doteq T_{\text{P}} + T_{\text{PHY}}$, and $H_1 \doteq T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$:

$$\begin{aligned} \text{MT} &= \frac{8L_{\text{DATA}}}{(T_{\text{DATA}} + H_1 + \overline{\text{CW}})} \\ \text{MD} &= T_{\text{DATA}} + T_{\text{DIFS}} + \overline{\text{CW}} \\ \text{TUL} &= \frac{8L_{\text{DATA}}}{(2T_{\text{PY}} + T_{\text{DIFS}} + T_{\text{SIFS}} + \overline{\text{CW}})} \\ \text{DLL} &= T_{\text{PY}} + T_{\text{DIFS}} + \overline{\text{CW}} \\ F(L) &= T_{\text{PY}} + T_{\text{SYM}} \times \text{Ceiling} \left[\frac{(16 + 6 + 8L)}{N_{\text{DBPS}}} \right]. \end{aligned}$$

Let T_{RTS} and T_{CTS} denote the transmission times of an RTS frame and a CTS frame, respectively. Let L_{RTS} and L_{CTS} denote frame lengths of an RTS frame and a CTS frame, respectively, in bytes. For the RTS/CTS mode, MT, MD, TUL, and DLL for the original DCF are given approximately as follows, where $T_{\text{RTS}} = F(L_{\text{RTS}})$, $T_{\text{CTS}} = F(L_{\text{CTS}})$, and $H_2 = T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{DATA}} + T_{\text{ACK}} + T_{\text{DIFS}} + T_{\text{SIFS}}$:

$$\begin{aligned} \text{MT}^{\text{RTS}} &= \frac{8L_{\text{DATA}}}{(H_2 + 2T_{\text{SIFS}} + \overline{\text{CW}})} \\ \text{MD}^{\text{RTS}} &= H_2 - T_{\text{ACK}} + T_{\text{SIFS}} + \overline{\text{CW}} \\ \text{TUL}^{\text{RTS}} &= \frac{8L_{\text{DATA}}}{(4T_{\text{PY}} + T_{\text{DIFS}} + 3T_{\text{SIFS}} + \overline{\text{CW}})} \\ \text{DLL}^{\text{RTS}} &= 3T_{\text{PY}} + T_{\text{DIFS}} + T_{\text{SIFS}} + \overline{\text{CW}}. \end{aligned}$$

A. Concatenation Mechanism

Since we are considering the best-case scenario, we assume that destination stations are all the same. We further assume that all data frames are of the same size, and at all times, N frames are concatenated. We will lose these assumptions in Section V-C. Let T_{CH} denote the transmission time of a CH. The CM, MT, MD, TUL, and DLL are easily derived approximately as follows: $\text{TUL}_{\text{CM}} = N \times \text{TUL}$ and $\text{DLL}_{\text{CM}} = \text{DLL}/N$, where $T_{\text{CH}} = F(L_{\text{H_DATA}} + 4)$

$$\begin{aligned} \text{MT}_{\text{CM}} &= \frac{8L_{\text{DATA}}}{\left[T_{\text{DATA}} + \frac{(T_{\text{CH}} + H_1 + \overline{\text{CW}})}{N} \right]} \\ \text{MD}_{\text{CM}} &= T_{\text{DATA}} + \frac{(T_{\text{CH}} + T_{\text{DIFS}} + \overline{\text{CW}})}{N}. \end{aligned}$$

With the CM, TUL increases N times, and DLL decreases N times. Furthermore, assuming that T_{CH} is relatively small, we observe that the effect of the CM on MT and MD is that the overhead time is decreased by N times. For the RTS/CTS mode, MT, MD, TUL, and DLL for the CM are given approximately as follows: $\text{TUL}_{\text{CM}}^{\text{RTS}} = N \times \text{TUL}^{\text{RTS}}$, $\text{DLL}_{\text{CM}}^{\text{RTS}} = \text{DLL}^{\text{RTS}}/N$, and $H_3 = T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{ACK}} + 3T_{\text{SIFS}} + T_{\text{DIFS}}$:

$$\begin{aligned} \text{MT}_{\text{CM}}^{\text{RTS}} &= \frac{8L_{\text{DATA}}}{\left[T_{\text{DATA}} + \frac{(T_{\text{CH}} + H_3 + \overline{\text{CW}})}{N} \right]} \\ \text{MD}_{\text{CM}}^{\text{RTS}} &= T_{\text{DATA}} + \frac{(T_{\text{CH}} + M + \overline{\text{CW}})}{N}. \end{aligned}$$

For the RTS/CTS mode, we have the following observations, which are similar to the basic access mechanism. With the CM, TUL increases N times, and DLL decreases N times. Furthermore, assuming that T_{CH} is relatively small, we observe that the effect of the CM on MT and MD is that the overhead time is decreased by N times.

B. Piggyback Mechanism

Since we are considering the best-case scenario, the best performance will be achieved when there are only two active nodes—one sender and one receiver at each transmission-ACK cycle. The sender always has frames ready to send to the receiver, and the receiver always has one frame ready to send to the sender after the sender gets the access the channel. We will lose this assumption in Section V-C.

Let $L_{\text{DATA-1}}$ and $L_{\text{DATA-2}}$ denote payload sizes of the first transmitted data frame and the piggybacked data frame, respectively, in bytes. Let $T_{\text{DATA-1}}$ and $T_{\text{DATA-2}}$ denote their transmission times, respectively. Let $H_4 = T_{\text{DATA-1}} + T_{\text{DATA-2}} + T_{\text{DIFS}} + T_{\text{SIFS}}$. The PM, MT, MD, TUL, and DLL are easily derived approximately as

$$\begin{aligned} \text{MT}_{\text{PM}} &= \frac{8(L_{\text{DATA-1}} + L_{\text{DATA-2}})}{(H_4 + T_{\text{SIFS}} + \overline{\text{CW}})} \\ \text{MD}_{\text{PM}} &= \frac{(H_4 + \overline{\text{CW}})}{2} \\ \text{TUL}_{\text{PM}} &= \frac{8(L_{\text{DATA-1}} + L_{\text{DATA-2}})}{(3T_{\text{PY}} + T_{\text{DIFS}} + 2T_{\text{SIFS}} + \overline{\text{CW}})} \\ \text{DLL}_{\text{PM}} &= T_{\text{PY}} + \frac{(T_{\text{DIFS}} + T_{\text{SIFS}} + \overline{\text{CW}})}{2}. \end{aligned}$$

Comparing with original MAC, we observe that the PM reduces T_{DIFS} and the average backoff time in half, except that it introduces an SIFS time. Roughly speaking, two frames will share one set of overhead of one original frame. Therefore, overhead is reduced, and performance will be improved. Let $H_5 = T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{DATA-1}} + T_{\text{DATA-2}} + T_{\text{DIFS}} + 3T_{\text{SIFS}}$. For the RTS/CTS mode, MT, MD, TUL, and DLL for the PM are given approximately as

$$\begin{aligned} \text{MT}_{\text{PM}}^{\text{RTS}} &= \frac{8(L_{\text{DATA-1}} + L_{\text{DATA-2}})}{(H_5 + T_{\text{SIFS}} + T_{\text{ACK}} + \overline{\text{CW}})} \\ \text{MD}_{\text{PM}}^{\text{RTS}} &= \frac{(H_5 + \overline{\text{CW}})}{2} \\ \text{TUL}_{\text{PM}}^{\text{RTS}} &= \frac{8(L_{\text{DATA-1}} + L_{\text{DATA-2}})}{(5T_{\text{PY}} + T_{\text{DIFS}} + 4T_{\text{SIFS}} + \overline{\text{CW}})} \\ \text{DLL}_{\text{PM}}^{\text{RTS}} &= 2T_{\text{PY}} + \frac{(T_{\text{DIFS}} + 3T_{\text{SIFS}} + \overline{\text{CW}})}{2}. \end{aligned}$$

C. Availability of CM and PM

In the previous two subsections, we assume that there are always frames available for both the CM and the PM. The term of availability is defined as follows: There are frames available

to be concatenated for the CM or to be piggybacked for the PM. However, frames may not be always available. If there are no frames available, normal DCF procedures will be used by default.

In reality, the most likely situation is that frames are available in one time and are not in another time. In such a mixed traffic situation, we use the superscript “mixed” to denote it. Let α_{CM} and α_{PM} denote probabilities that frames are available for the CM and the PM, respectively. For the CM with the basic access mode, we have

$$\begin{aligned} MT_{CM}^{\text{mixed}} &= \frac{1}{\left[\frac{(1-\alpha_{CM})}{MT} + \frac{\alpha_{CM}}{MT_{CM}} \right]} \\ MD_{CM}^{\text{mixed}} &= (1 - \alpha_{CM})MD + \alpha_{CM}MD_{CM} \\ TUL_{CM}^{\text{mixed}} &= \frac{N \times TUL}{[N - (N - 1)\alpha_{CM}]} \\ DLL_{CM}^{\text{mixed}} &= \frac{[N - (N - 1)\alpha_{CM}] DLL}{N}. \end{aligned}$$

For the CM with the RTS/CTS mode, we have

$$\begin{aligned} MT_{CM}^{\text{RTS mixed}} &= \frac{1}{\left[\frac{(1-\alpha_{CM})}{MT^{\text{RTS}}} + \frac{\alpha_{CM}}{MT_{CM}^{\text{RTS}}} \right]} \\ MD_{CM}^{\text{RTS mixed}} &= (1 - \alpha_{CM})MD^{\text{RTS}} + \alpha_{CM}MD_{CM}^{\text{RTS}} \\ TUL_{CM}^{\text{RTS mixed}} &= \frac{N \times TUL^{\text{RTS}}}{[N - (N - 1)\alpha_{CM}]} \\ DLL_{CM}^{\text{RTS mixed}} &= \frac{[N - (N - 1)\alpha_{CM}] DLL^{\text{RTS}}}{N}. \end{aligned}$$

For the PM with the basic access mode, we have

$$\begin{aligned} MT_{PM}^{\text{mixed}} &= \frac{1}{\left[\frac{(1-\alpha_{PM})}{MT} + \frac{\alpha_{PM}}{MT_{PM}} \right]} \\ MD_{PM}^{\text{mixed}} &= (1 - \alpha_{PM})MD + \alpha_{PM}MD_{PM} \\ TUL_{PM}^{\text{mixed}} &= \frac{1}{\left[\frac{(1-\alpha_{PM})}{TUL} + \frac{\alpha_{PM}}{TUL_{PM}} \right]} \\ DLL_{PM}^{\text{mixed}} &= (1 - \alpha_{PM})DLL + \alpha_{PM}DLL_{PM}. \end{aligned}$$

For the CM with the RTS/CTS mode, we have

$$\begin{aligned} MT_{PM}^{\text{RTS mixed}} &= \frac{1}{\left[\frac{(1-\alpha_{PM})}{MT^{\text{RTS}}} + \frac{\alpha_{PM}}{MT_{PM}^{\text{RTS}}} \right]} \\ MD_{PM}^{\text{RTS mixed}} &= (1 - \alpha_{PM})MD^{\text{RTS}} + \alpha_{PM}MD_{PM}^{\text{RTS}} \\ TUL_{PM}^{\text{RTS mixed}} &= \frac{1}{\left[\frac{(1-\alpha_{PM})}{TUL^{\text{RTS}}} + \frac{\alpha_{PM}}{TUL_{PM}^{\text{RTS}}} \right]} \\ DLL_{PM}^{\text{RTS mixed}} &= (1 - \alpha_{PM})DLL^{\text{RTS}} + \alpha_{PM}DLL_{PM}^{\text{RTS}}. \end{aligned}$$

VI. SATURATION PERFORMANCE ANALYSIS

A. Analytical Model

Since the proposed CM does not change any system behavior such as the backoff behavior except that frames become larger, it is easy to see that the CM should have a better performance intuitively. For the proposed PM, the backoff behavior does not change if we can treat the first data frame and second piggyback data frame as one data frame transmission. We will use saturation throughput and saturation delay to evaluate the proposed mechanisms. To be in a saturation situation, we assume that each station always has frames ready to send. For the CM, we assume that all data frames are of the same size, and at all times, N frames are concatenated. We further assume that the destination stations are all the same. We will lose these assumptions in Section VI-B. For the PM, we assume that the receiver always has one frame ready to send to the sender after the sender gets the access the channel. We will lose these assumptions in Section VI-B, too.

Let W_0 denote CW_{\min} for convenience. Let j ($j = 0, 1, \dots, L_{\text{retry}}$) denote the j th backoff stage, and let W_j denote CW in the j th retry/retransmission (or the j th backoff stage). The relationships among W_j , W_0 , CW_{\max} , and L_{retry} are given as

$$W_j = \begin{cases} 2^j W_0, & \text{for } j = 0, 1, \dots, m-1 \\ & \text{if } L_{\text{retry}} > m \\ 2^m W_0 = CW_{\max}, & \text{for } j = m, \dots, L_{\text{retry}} \\ & \text{if } L_{\text{retry}} > m \\ 2^j W_0, & \text{for } j = 0, 1, \dots, L_{\text{retry}} \\ & \text{if } L_{\text{retry}} \leq m \end{cases} \quad (1)$$

where $m = \log_2(CW_{\max}/W_0)$ and $W_0 = CW_{\min}$.

As Bianchi's model [6] and ZA's model [7], $b(t)$ is defined as a random process representing the value of backoff counter at time t , and $s(t)$ is defined as the random process representing the backoff stage j , where $0 \leq j \leq L_{\text{retry}}$, and $b(t)$ is uniformly chosen in the range $(0, 1, \dots, W_j - 1)$. Let p denote the probability that a transmitted frame collides. p is also equal to the probability that a station in the backoff stage senses that the channel is busy. The bidimensional random process $\{s(t), b(t)\}$ is a discrete-time Markov chain under the assumption that the probability p is independent of the backoff procedure [6], [7]. Therefore, the state of each station is described by $\{j, k\}$, where j stands for the backoff stage, and k stands for the backoff delay and takes values $(0, 1, \dots, W_j - 1)$.

The state transition diagram is shown in Fig. 4. As illustrated in the figure, in the L_{retry} th backoff stage, the frame is dropped if a collision occurs. From Fig. 4, observe that our model is different from Bianchi's model [6] and ZA's model [7] in many aspects: The backoff counter is stopped when a transmission is detected on the channel; there is no nonbackoff stage; and there exists a finite retransmission limit L_{retry} . Let $b_{j,k} = \lim_{t \rightarrow \infty} \Pr\{s(t) = j, b(t) = k\}$ be the stationary distribution

of the Markov chain. In steady state, we can derive the following relations through chain regularities:

$$\begin{aligned}
 b_{j,0} &= p^j b_{0,0}, \quad 0 \leq j \leq L_{\text{retry}} \\
 b_{j,k} &= \frac{W_j - k}{W_j} \frac{1}{1-p} b_{j,0} \\
 & \quad 0 \leq j \leq L_{\text{retry}} \text{ and } 1 \leq k \leq W_j - 1 \\
 \sum_{j=0}^{L_{\text{retry}}} \sum_{k=0}^{W_j-1} b_{j,k} &= 1 \\
 b_{0,0} &= \frac{1}{\left\{ \sum_{j=0}^{L_{\text{retry}}} \left[1 + \frac{1}{1-p} \sum_{k=1}^{W_j-1} \frac{W_j-k}{W_j} \right] p^j \right\}}. \quad (2)
 \end{aligned}$$

Let τ be the probability that a station transmits during a generic slot time. A station transmits when its backoff counter reaches zero, i.e., the station is at any of states $\{j, 0\}$. Let n denote the number of stations. The probability p is that a station in the backoff stage senses that the channel is busy. p is also the probability that a transmitted frame collides

$$\tau = \sum_{j=0}^{L_{\text{retry}}} b_{j,0} = b_{0,0} \frac{1 - p^{L_{\text{retry}}+1}}{1 - p} \quad (3)$$

$$p = 1 - (1 - \tau)^{n-1}. \quad (4)$$

Plugging (2) into (3), we can solve unknown parameters numerically from (3) and (4). Let p_b denote the probability that the channel is busy. It happens when at least one station transmits during a slot time. Let p_s denote the probability that a successful transmission occurs in a slot time. We have $p_b = 1 - (1 - \tau)^n$ and $p_s = n\tau(1 - \tau)^{n-1}$.

Let L^* denote the length of the longest packet in a collision. Let δ , L , T_s , and T_c denote the duration of an empty slot time, the payload size, the average time that the channel is sensed busy because of a successful transmission, and the average time that the channel has a collision, respectively. Let T_H , $T_{E(L)}$, $T_{E(L^*)}$, and T_{CH} denote times of transmitting the header (including MAC header, PHY header, and/or tail), a payload with length $E(L)$, a payload with length $E(L^*)$, and a CH, respectively. Let $L2$ denote the payload length of the piggybacked data frame in bytes, and let $T_{E(L2)}$ denote its average transmission time. We have $T_s = T_H + T_{E(L)} + H_1$, $T_s^{\text{CM}} = T_{CH} + N(T_H + T_{E(L)}) + H_1$, $T_s^{\text{PM}} = T_{E(L)} + T_{E(L2)} + 2T_H + T_{\text{SIFS}} + H_1$, $T_c = T_H + T_{E(L^*)} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$, $T_c^{\text{CM}} = T_{CH} + N(T_H + T_{E(L)}) + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$, and $T_c^{\text{PM}} = T_H + T_{E(L^*)} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$.

For IEEE 802.11a, we have $T_H + T_{E(L)} = F(L_{H_DATA} + E(L))$, $T_H + T_{E(L^*)} = F(L_{H_DATA} + E(L^*))$, and $T_H + T_{E(L2)} = F(L_{H_DATA} + E(L2))$.

For the RTS/CTS mode, we have $T_c^{\text{RTS}} = T_{\text{RTS}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$, $T_s^{\text{RTS}} = T_H + T_{E(L)} + H_3$, $T_s^{\text{RTS}}_{\text{CM}} = T_{CH} + N(T_H + T_{E(L)}) + H_3$, and $T_s^{\text{RTS}}_{\text{PM}} = 2T_H + T_{E(L)} + T_{E(L2)} + T_{\text{SIFS}} + H_3$.

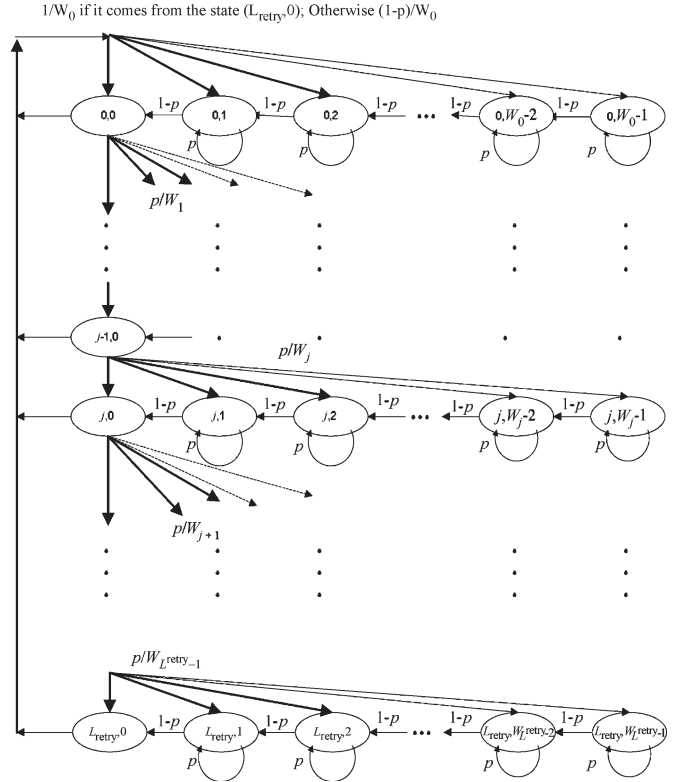


Fig. 4. State transition diagram.

Let S denote the normalized saturation throughput. We have $S = GT_{E(L)}$, $S_{\text{CM}} = GNT_{E(L)}$, and $S_{\text{PM}} = G(T_{E(L1)} + T_{E(L2)})$, where $G \doteq p_s / ((1 - p_b)\delta + p_s T_s + [p_b - p_s]T_c)$.

Let P_{drop} denote the frame-dropping probability. We have $P_{\text{drop}} = p^{L_{\text{retry}}+1}$. Saturation delay is the average delay under the saturation condition, and it includes the medium access delay (due to backoff, collisions, etc.), transmission delay, and interframe spaces (such as SIFS). The average backoff delay depends on the value of a station's backoff counter and the duration when the counter freezes due to others' transmissions. Let X denote the random variable representing the total number of backoff slots that a frame encounters without considering the case when the counter freezes. The probability that the frame is successfully transmitted after the j th retry [which is the $(j + 1)$ th transmission] is $p^j(1 - p)/(1 - P_{\text{drop}})$, which uses conditional probability on a successful transmission with probability $1 - P_{\text{drop}}$. The average number of backoff slots after the j th retry is $\sum_{h=0}^j (W_h - 1)/2$. Note that only successful transmissions are considered. Let B denote the random variable representing the total number of slots when the counter freezes, which a frame encounters. We have $E(X) = \sum_{j=0}^{L_{\text{retry}}} [p^j(1 - p)] / (1 - p^{L_{\text{retry}}+1}) \sum_{h=0}^j (W_h - 1)/2$ and $E(B) = (E(X)/(1 - p))p$. We can treat $E(X)$ and $E(B)$ as the total number of idle slots and the total number of busy slots that the frame encounters during backoff stages, respectively. Let D denote the random variable representing the frame delay. Let T_o denote the time that a station has to wait when its frame transmission collides before sensing the channel again. Let $T_{\text{ACK_timeout}}$ and $T_{\text{CTS_timeout}}$ denote the duration of the ACK timeout and the duration of the CTS

timeout, respectively. Note that $E(N_{\text{retry}})$ is one less than the number of transmissions. The average slot lengths are δ , $[p_s T_s + (p_b - p_s) T_c]/p_b$, $(T_c + T_o)$, and T_s for an idle slot at states $\{j, k\}$ ($k > 0$), a busy slot at states $\{j, k\}$ ($k > 0$), a failed transmission slot for this station at states $\{j, 0\}$, and a successful transmission at states $\{j, 0\}$, respectively. We have $T_o^{\text{basic}} = T_{\text{SIFS}} + T_{\text{ACK_timeout}}$, $T_o^{\text{RTS/CTS}} = T_{\text{SIFS}} + T_{\text{CTS_timeout}}$, and

$$E(D) = E(X)\delta + \frac{E(B)[p_s T_s + (p_b - p_s) T_c]}{p_b} + E(N_{\text{retry}})(T_c + T_o) + T_s$$

$$E(D)_{\text{CM}} = \frac{E(D)}{N}$$

$$E(D)_{\text{PM}} = \frac{E(D)}{2}.$$

B. Availability of CM and PM

In the previous two subsections, we assume that there are always available frames for both the CM and the PM. We will lose this assumption as follows. Let α_{CM} and α_{PM} denote probabilities that frames are available for the CM and the PM, respectively. For the CM with the basic access mode, we have $T_{s \text{ CM}}^{\text{mixed}} = (1 - \alpha_{\text{CM}})T_s + \alpha_{\text{CM}}T_s^{\text{CM}}$, $T_{c \text{ CM}}^{\text{mixed}} = (1 - \alpha_{\text{CM}})T_c + \alpha_{\text{CM}}T_c^{\text{CM}}$, and

$$S_{\text{CM}}^{\text{mixed}} = \frac{(1 - \alpha_{\text{CM}})p_s T_{E(L)} + \alpha_{\text{CM}} N p_s T_{E(L)}}{(1 - p_b)\delta + p_s T_{s \text{ CM}}^{\text{mixed}} + [p_b - p_s]T_{c \text{ CM}}^{\text{mixed}}}.$$

For the CM with the RTS/CTS mode, we have $T_{c \text{ CM}}^{\text{mixed RTS}} = T_c^{\text{RTS}}$, $T_{s \text{ CM}}^{\text{mixed RTS}} = (1 - \alpha_{\text{CM}})T_s^{\text{RTS}} + \alpha_{\text{CM}}T_s^{\text{CM RTS}}$, and

$$S_{\text{CM}}^{\text{mixed RTS}} = \frac{(1 - \alpha_{\text{CM}})p_s T_{E(L)} + \alpha_{\text{CM}} N p_s T_{E(L)}}{(1 - p_b)\delta + p_s T_{s \text{ CM}}^{\text{mixed RTS}} + [p_b - p_s]T_{c \text{ CM}}^{\text{mixed RTS}}}.$$

For the PM with the basic access mode, we have $T_{c \text{ PM}}^{\text{mixed}} = T_c^{\text{PM}}$, $T_{s \text{ PM}}^{\text{mixed}} = (1 - \alpha_{\text{PM}})T_s + \alpha_{\text{PM}}T_s^{\text{PM}}$, and

$$S_{\text{PM}}^{\text{mixed}} = \frac{(1 - \alpha_{\text{PM}})p_s T_{E(L)} + \alpha_{\text{PM}} p_s (T_{E(L1)} + T_{E(L2)})}{(1 - p_b)\delta + p_s T_{s \text{ PM}}^{\text{mixed}} + [p_b - p_s]T_{c \text{ PM}}^{\text{mixed}}}.$$

For the PM with the RTS/CTS mode, we have $T_{c \text{ PM}}^{\text{mixed RTS}} = T_c^{\text{RTS}}$, $T_{s \text{ PM}}^{\text{mixed RTS}} = (1 - \alpha_{\text{PM}})T_s^{\text{RTS}} + \alpha_{\text{PM}}T_s^{\text{PM RTS}}$, and

$$S_{\text{PM}}^{\text{mixed RTS}} = \frac{(1 - \alpha_{\text{PM}})p_s T_{E(L)} + \alpha_{\text{PM}} N p_s T_{E(L)}}{(1 - p_b)\delta + p_s T_{s \text{ PM}}^{\text{mixed RTS}} + [p_b - p_s]T_{c \text{ PM}}^{\text{mixed RTS}}}.$$

VII. SIMULATION VALIDATIONS

In this subsection, we conduct simulations to validate analytic results. IEEE 802.11a simulation models had been developed based on IEEE Standard 802.11a [3] and OPNET wireless LAN simulation model 8.0A (for IEEE 802.11b DCF). We

TABLE I
SIMULATION VERSUS ANALYTICAL THROUGHPUT
RESULTS (in Mb/s)

			Payload (byte)		
			100	500	1000
Best Case	CM	A	6.1657	21.0804	30.1035
		S	6.1595	21.0741	30.0854
		E	-0.1%	-0.03%	-0.06%
	PM	A	6.5440	21.9479	30.9777
		S	6.5401	21.9216	30.9498
		E	-0.06%	-0.12%	-0.09%
Saturation	-	-	Number of Stations		
			10	30	45
			A	0.1578	0.1442
	CM	S	0.1597	0.1456	0.1393
		E	1.22%	0.96%	0.45%
		PM	A	0.1505	0.1443
	S		0.1521	0.1455	0.1421
	E		1.04%	0.83%	0.53%

(Legend: A: Analytical; S: Simulation; E: Error)

adapt our simulation model with similar assumptions as those in the analytical model. The data rate is 54 Mb/s. For the CM, we let $N = 2$, where N is the number of concatenated frames. All the simulation results have over 95% confidential intervals. The relative error is calculated by $[(\text{Simulation Result} - \text{Analytical Result})/\text{Simulation Result}]$.

Table I shows the simulation results versus numerical results of the CM and the PM at both the best-case scenario and the saturation scenario. As illustrated in the table, analytical results and simulation results match pretty well. At the saturation scenario, the payload size is 100 B. Compared with the best-case scenario, relative errors in the saturation scenario are relative larger.

VIII. NUMERICAL RESULTS

In this section, numerical results for the CM and the PM are presented as well as availability study of CM and PM. Parameters of IEEE 802.11a are listed as follows [3]: $T_{\text{slot}} = 9 \mu\text{s}$; $T_{\text{SIFS}} = 16 \mu\text{s}$; $\text{CW}_{\text{min}} = 15 + 1$; $T_p = 16 \mu\text{s}$; $T_{\text{PHY}} = 4 \mu\text{s}$; $T_{\text{DIFS}} = 34 \mu\text{s}$; $T_{\text{SYM}} = 4 \mu\text{s}$; $L_{\text{H_DATA}} = 28$; and $L_{\text{ACK}} = 14$. We let $L_{\text{retry}} = 7$. Data rate is 54 Mb/s. The number of concatenated frames N is 2. For the PM, we assume that the size of the first data frame is the same as the size of the piggybacked frame.

A. CM at the Best Case Scenario

Fig. 5(a) shows MT and TUL with/without CM. As illustrated in the figure, MT_{CM} and TUL_{CM} are much larger than MT and TUL, respectively. For example, MT_{CM} (payload size = 400 B) is almost the same as MT (payload size = 800 B), and TUL_{CM} (payload size = 400 B) is exactly the same as TUL (payload size = 800 B). When the payload size is small, MT_{CM} is even larger than TUL. Fig. 5(b) shows MD and DLL with/without CM. As illustrated in the figure, MD_{CM} and DLL_{CM} are much smaller than MD and DLL, respectively. For example, MD_{CM} (payload size = 800 B) is almost the same as MD (payload size = 400 B), and DLL_{CM} is exactly half of DLL. When the payload size is small, MD_{CM} is even smaller than DLL. Much better performance is expected for larger N .

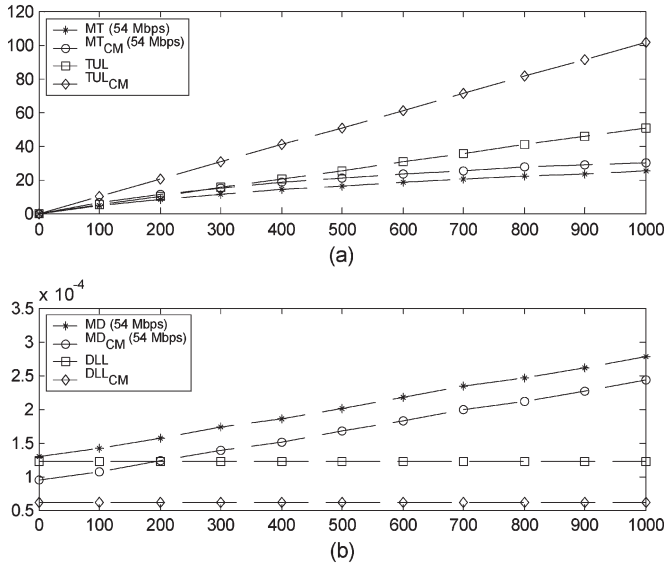


Fig. 5. MT (in Mb/s), TUL (in Mb/s), MD (in microseconds), and DLL (in microseconds). (a) MT and TUL versus payload size (in bytes). (b) MT and DLL versus payload size (in bytes).

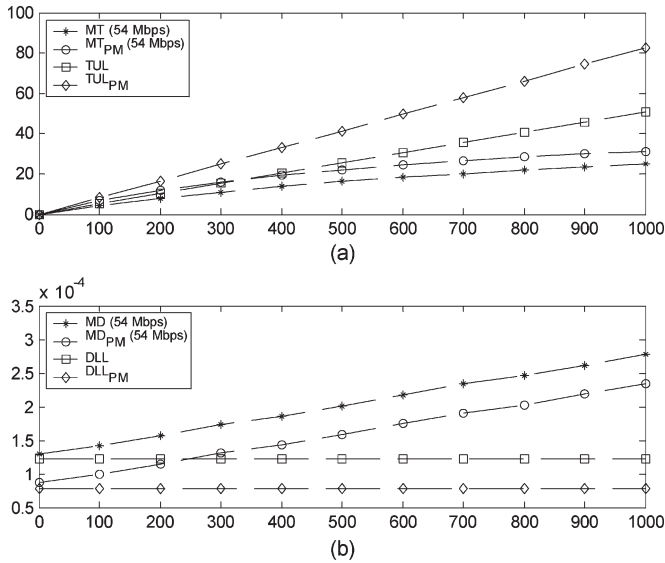


Fig. 6. MT (in Mb/s), TUL (in Mb/s), MD (in microseconds), and DLL (in microseconds). (a) MT and TUL versus payload size (in bytes). (b) MT and DLL versus payload size (in bytes).

B. PM at the Best Case Scenario

Fig. 6(a) shows MT and TUL with/without PM. As illustrated in the figure, MT_{PM} and TUL_{PM} are much larger than MT and TUL, respectively. When the payload size is small, MT_{PM} is even larger than TUL. Fig. 6(b) shows MD and DLL with/without PM. As illustrated in the figure, MD_{PM} and DLL_{PM} are much smaller than MD and DLL, respectively. When the payload size is small, MD_{PM} is even smaller than DLL.

C. CM at the Saturation Scenario

In Fig. 7, the payload size is 100 B. Fig. 7(a) [Fig. 7(b)] compares throughputs [delays] with/without CM over the number of active stations. Fig. 7 shows that the throughput (delays) decreases (increases) as the number of stations increases.

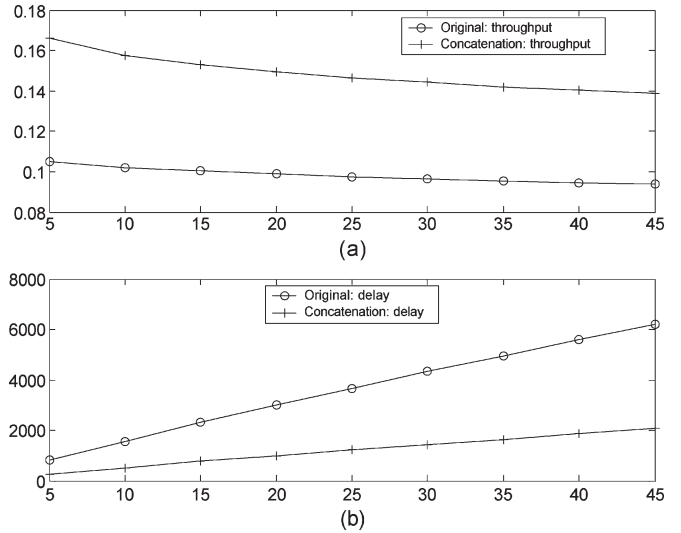


Fig. 7. Compare throughputs and delay ($L = 100$ B). (a) Saturation throughput versus number of stations. (b) Saturation delay (in microseconds) versus number of stations.

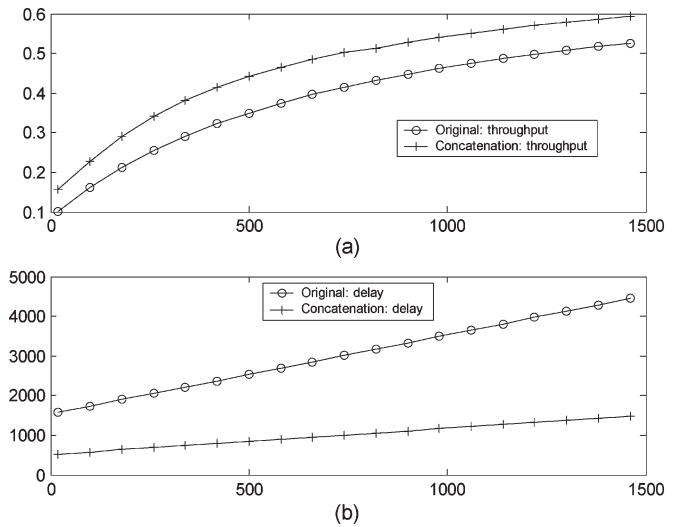


Fig. 8. Compare throughputs and delays ($N = 10$). (a) Saturation throughput versus payload (in bytes). (b) Saturation delay (in microseconds) versus payload (in bytes).

Fig. 7 shows that CM has a much better throughput and a much lower delay. Fig. 7(b) also shows that the reduced-delay difference becomes larger as the number of stations increases.

In Fig. 8, the number of active stations is 10. Fig. 8(a) [Fig. 8(b)] compares throughputs [delays] with/without CM over the payload size. Fig. 8 shows that CM has a much higher throughput and a much lower delay. Both the throughput and the delay increase as the payload size increases. The reduced-delay difference is relatively constant with respect to the payload size. The figure also indicates that from the collision aspect, concatenation threshold could be as large as 1500 B without affecting throughput and delay.

D. PM at the Saturation Scenario

In Fig. 9, the payload size is 100 B. Fig. 9(a) [Fig. 9(b)] compares throughputs [delays] with/without PM over the

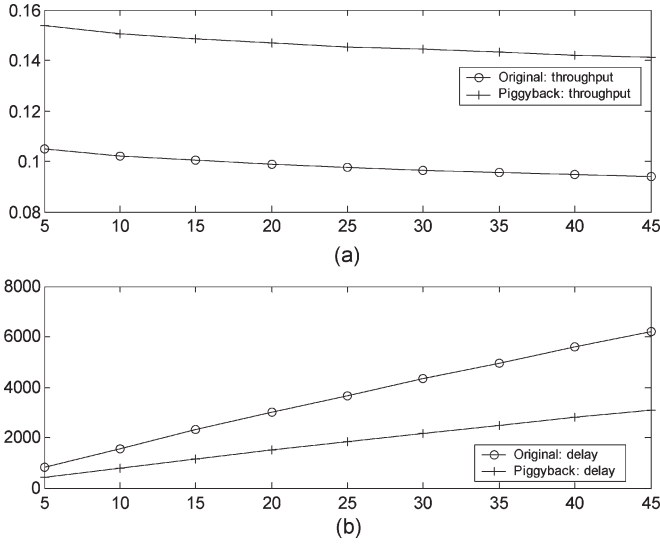


Fig. 9. Compare throughputs and delay ($L = 100$ B). (a) Saturation throughput versus number of stations. (b) Saturation delay (in microseconds) versus number of stations.

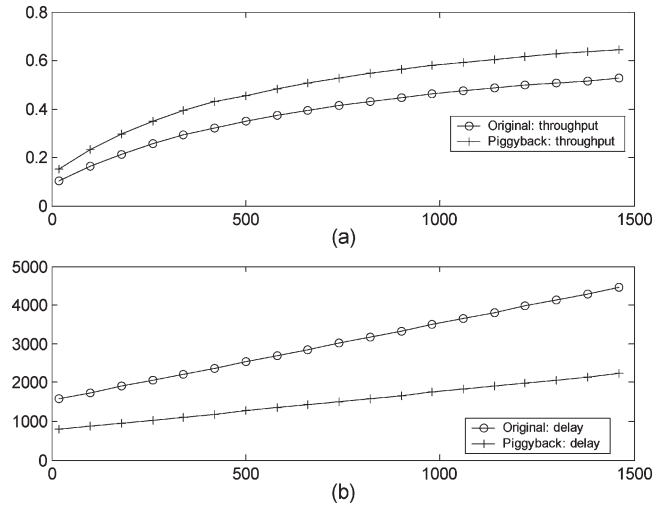


Fig. 10. Compare throughputs and delays ($N = 10$). (a) Saturation throughput versus payload (in bytes). (b) Saturation delay (in microseconds) versus payload (in bytes).

number of active stations. Fig. 9 shows that PM has a much better throughput and a much lower delay. It also shows that the throughput (delay) decreases (increases) as the number of stations increases. Fig. 9(b) also shows that the reduced-delay difference becomes larger as the number of stations increases.

In Fig. 10, the number of active stations is 10. Fig. 10(a) [Fig. 10(b)] compares throughputs [delays] with/without PM over the payload size. Fig. 10 shows that PM has a much better throughput and a lower delay. The figure also shows that both the throughput and the delay increase as the payload size increases. Fig. 10(b) also shows that the reduced-delay difference increases a little as the payload size increases.

E. Availability of CM and PM

In the previous four subsections, we assume that there are always available frames for both CM and PM. In this subsection,

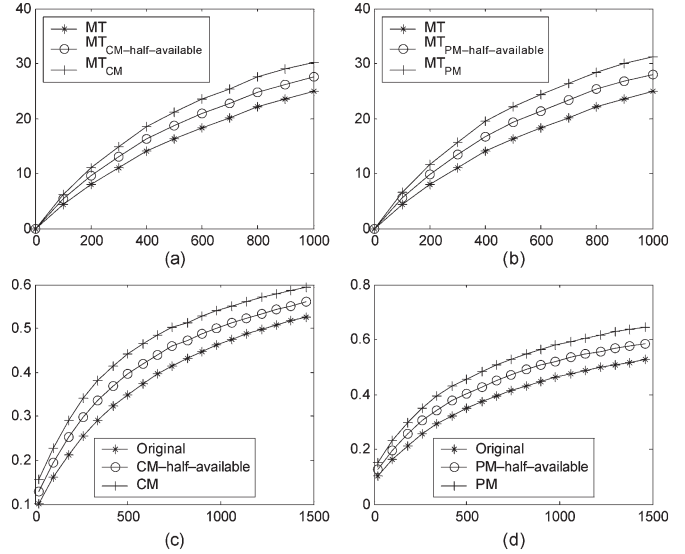


Fig. 11. Throughputs with availability (in Mb/s). (a) Concatenation: MT versus payload (in bytes). (b) Piggyback: MT versus payload (in bytes). (c) CM: throughput versus payload (in bytes). (d) PM: throughput versus payload (in bytes).

tion, we study a mixed situation when frames are available in one time and are not in another time. Fig. 11(a) and (b) shows throughputs of CM and PM, respectively, at the best case scenario when $\alpha_{CM} = 0.5$ and $\alpha_{PM} = 0.5$. The figures indicate that throughputs for both schemes are between the original throughput and the throughput when frames are always available. Fig. 11(c) and (d) shows throughputs of CM and PM, respectively, at the saturation scenario when $\alpha_{CM} = 0.5$ and $\alpha_{PM} = 0.5$. The figures indicate that throughputs for both schemes are between the original throughput and the throughput when frames are always available.

IX. CONCLUSION

Two novel mechanisms were proposed in this paper to reduce overhead and to improve the IEEE 802.11 medium access control (MAC) performance. Performance was evaluated at both the best case scenario and the saturation scenario. Better performance results were produced as expectations since both proposed mechanisms do not change system behaviors such as backoff.

For the concatenation mechanism (CM), the virtual frame format is designed in such a way that the processing time of combining and decomposing frames is insignificant since concatenated frames are transmitted one by one separately and received one by one separately. A more efficient virtual frame format can be easily designed, though it sacrifices complexity and processing time. At the best case scenario, the throughput upper limit (TUL) increases N times, and the delay lower limit (DLL) decreases N times, where N is the number of concatenated frames; the effect on the maximum throughput (MT) and the minimum delay (MD) is that the overhead time is decreased almost by N times. The saturation throughput has been improved greatly. The saturation delay has also been greatly improved especially when the number of active stations

is large. The reduced-delay difference is relatively constant with the respect to the payload size.

For the piggyback mechanism (PM), performance has been greatly improved at both the best-case scenario and the saturation scenario. The reduced-delay difference increases as the number of active stations increases but increases only a little as the payload size increases.

Availability of the proposed mechanisms was studied. If there are no available frames, normal distributed coordination function (DCF) procedures will be used by default, and the system performance will be the same as before. Studies show that the performance has been greatly improved in a more realistic traffic scenario at which frames are available in one time and are not in another time.

Studies show that from the collision aspect, concatenation threshold could be as large as 1500 B without affecting throughput and delay. We recommend that concatenation threshold could be chosen as $\min(\alpha, \beta, \theta)$, where $\alpha = 1500$ B is the Ethernet frame size, β is the fragmentation threshold, and θ is the measured average frame size among relatively large frames during a measurement interval such as a beacon interval.

The proposed mechanisms can be optional mechanisms for the existing IEEE 802.11 MAC with very minor revisions. Furthermore, they are also very useful for the IEEE 802.11n, established in September 2003, emphasizing higher throughput in wireless local area networks (LANs).

REFERENCES

- [1] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1999.
- [2] IEEE 802.11b WG, Part 11: High-Speed Physical Layer Extension in the 2.4 GHz Band, 1999.
- [3] IEEE 802.11a WG, Part 11: High-Speed Physical Layer in the 5 GHz Band, 1999.
- [4] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 355–357, Aug. 2002.
- [5] —, "Performance analysis and enhancement for the current and future IEEE 802.11 MAC protocols," *ACM SIGMOBILE Mobile Comput. Commun. Rev. (MC2R)*, vol. 7, no. 2, pp. 6–19, Apr. 2003.
- [6] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
- [7] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: Throughput and delay analysis," *Comput. Commun.*, vol. 25, no. 3, pp. 313–321, 2002.
- [8] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LANs: Analysis and enhancement," in *Proc. IEEE Inf. Commun. (INFOCOM)*, New York, 2002, pp. 599–607.

- [9] Y. Xiao, "A simple and effective priority scheme for IEEE 802.11," *IEEE Commun. Lett.*, vol. 7, no. 2, pp. 70–72, Feb. 2003.
- [10] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–790, Dec. 2000.
- [11] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 9, pp. 1774–1786, Sep. 2000.
- [12] B. Bing and R. Subramanianb, "A novel technique for quantitative performance evaluation of wireless LANs," *Comput. Commun.*, vol. 21, no. 9, pp. 833–838, Jul. 1998.
- [13] K. C. Huang and K.-C. Chen, "Interference analysis of nonpersistent CSMA with hidden terminals in multicell wireless data networks," in *Proc. IEEE Personal Indoor Mobile Radio Communications (PIMRC)*, Toronto, ON, Canada, Sep. 1995, pp. 907–911.
- [14] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," *Wireless Netw.*, vol. 3, no. 3, pp. 217–234, 1997.
- [15] Y. C. Tay and K. C. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," *Wireless Netw.*, vol. 7, no. 2, pp. 159–171, 2001.
- [16] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1506–1515, Jul. 2005.



Yang Xiao (SM'04) worked at Micro Linear as a Medium Access Control (MAC) Architect involving the IEEE 802.11 standard enhancement work before he joined the Department of Computer Science at the University of Memphis, Memphis, TN, in 2002. He was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004. He currently serves as Editor-in-Chief for the *International Journal of Security and Networks (IJSN)* and the *International Journal of Sensor Networks (IJSNet)*. He serves as an Associate Editor or on editorial boards for the

following refereed journals: *International Journal of Communication Systems* (Wiley); *Wireless Communications and Mobile Computing (WCMC)*; *EURASIP Journal on Wireless Communications and Networking (WCN)*; and *International Journal of Wireless and Mobile Computing (IJWMC)*. He has served as a (lead) journal Guest Editor for the IJSN special issue on security issues in sensor networks in 2005, a (lead) journal Guest Editor for the EURASIP WCN special issue on wireless network security in 2005, a (sole) journal Guest Editor for the *Computer Communications* (Elsevier) special issue on energy-efficient scheduling and MAC for sensor networks, wireless personal area networks (WPANs), wireless local area networks (WLANs), and wireless metropolitan access networks (WMANs) in 2005, a (lead) journal Guest Editor for the WCMC (Wiley) special issue on mobility, paging, and quality of service management for future wireless networks in 2004, and as a (lead) journal Guest Editor for the IJWMC special issue on MAC for WLANs, WPANs, ad hoc networks, and sensor networks in 2004. He serves as a Referee/Reviewer for many funding agencies, as well as a Panelist for the U.S. National Science Foundation. His research areas include wireless networks and network security.

Mr. Xiao is a Senior Member of the IEEE Computer Society and the IEEE Communications Society.