

# Aggregation With Fragment Retransmission for Very High-Speed WLANs

Tianji Li, *Member, IEEE*, Qiang Ni, *Senior Member, IEEE*, David Malone, Douglas Leith, *Member, IEEE*, Yang Xiao, *Senior Member, IEEE*, and Thierry Turletti, *Senior Member, IEEE*

**Abstract**—In upcoming very high-speed wireless LANs (WLANs), the physical (PHY) layer rate may reach 600 Mbps. To achieve high efficiency at the medium access control (MAC) layer, we identify fundamental properties that must be satisfied by any CSMA-/CA-based MAC layers and develop a novel scheme called aggregation with fragment retransmission (AFR) that exhibits these properties. In the AFR scheme, multiple packets are aggregated into and transmitted in a single large frame. If errors happen during the transmission, only the corrupted fragments of the large frame are retransmitted. An analytic model is developed to evaluate the throughput and delay performance of AFR over noisy channels and to compare AFR with similar schemes in the literature. Optimal frame and fragment sizes are calculated using this model. Transmission delays are minimized by using a zero-waiting mechanism where frames are transmitted immediately once the MAC wins a transmission opportunity. We prove that zero-waiting can achieve maximum throughput. As a complement to the theoretical analysis, we investigate the impact of AFR on the performance of realistic application traffic with diverse requirements by simulations. We have implemented the AFR scheme in the NS-2 simulator and present detailed results for TCP, VoIP, and HDTV traffic.

The AFR scheme described was developed as part of the IEEE 802.11n working group work. The analysis presented here is general enough to be extended to proposed schemes in the upcoming 802.11n standard. Trends indicated in this paper should extend to any well-designed aggregation schemes.

**Index Terms**—IEEE 802.11, IEEE 802.11n, medium access control (MAC), wireless LAN (WLAN).

## I. INTRODUCTION

WIRELESS LANs (WLANs) based on IEEE 802.11 technology are becoming increasingly ubiquitous. With the aim of supporting rich multimedia applications such as HDTV (20 Mbps) and DVD (9.8 Mbps), the technology trend is toward increasingly higher bandwidths. Some recent 802.11n proposals seek to support physical (PHY) layer rates of up to 600

Mbps [4], [6], [7], [47]. However, higher PHY rates do not necessarily translate into corresponding increases in medium access control (MAC) layer throughput. Indeed, it is well known that the MAC efficiency of 802.11 typically decreases with increasing PHY rates [9], [48]. The reason is that while increasing PHY rates lead to faster transmission of the MAC frame payloads, overhead such as PHY headers and contention time typically do not decrease at the same rate and thus begin to dominate frame transmission times. This behavior is illustrated in Fig. 1(a), where it can be seen that, even under ideal case conditions, the MAC efficiency falls from 42% at a PHY rate of 54 Mbps to only 10% at 432 Mbps.

The problem here is a fundamental one for MAC design, namely that, due to cross-layer interactions, the throughput of the current 802.11 MAC does not scale well with increasing PHY rates. With continuing improvements in PHY technology and demand for higher throughput, the MAC scaling behavior is of key importance.

While the current focus of 802.11n activity is on achieving 100-Mbps throughput at the MAC layer, still higher target data rates can be expected in the future. To avoid repeated MAC re-designs, one basic question that we seek to answer is whether it is feasible to extend the 802.11 MAC to maintain high MAC efficiency regardless of PHY rates. We answer this in the affirmative. In particular, we identify fundamental properties that must be satisfied by any CSMA-/CA-based MAC layers and develop a novel scheme called aggregation with fragment retransmission (AFR) that exhibits these properties. In the AFR scheme, multiple packets are aggregated into and transmitted in a single large frame.<sup>1</sup> If errors occur during the transmission, only the corrupted fragments of the large frame are retransmitted. In this scheme, a new delimitation mechanism allows for higher throughput with less overhead compared to previous designs. We study a fragmentation technique where packets longer than a threshold are divided into fragments before being aggregated. An analytic model is developed to evaluate the throughput and delay of AFR over noisy channels and to compare AFR with competing schemes. Optimal frame and fragment sizes are calculated using this model, and an algorithm for dividing packets into near-optimal fragments is designed.

A second question we seek to answer is whether higher transmission delays are an unavoidable result of using aggregation to achieve high throughput. In particular, is additional delay necessarily introduced 1) by the need to wait until sufficient packets arrive to allow a large frame to be formed, and 2) for transmission of a large frame? We answer this question in the negative. Specifically, we propose a zero-waiting mechanism

<sup>1</sup>We define a *packet* as what the MAC receives from the upper layer, a *frame* as what the MAC transfers to the PHY layer, and a *fragment* as part(s) of a frame.

Manuscript received September 08, 2006; revised July 26, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Buddhikot. Current version published April 15, 2009. This work was supported by the Science Foundation Ireland under Grant IN3/03/I346 and Irish Research Council for Science, Engineering and Technology.

T. Li, D. Malone, and D. Leith are with the Hamilton Institute, National University of Ireland, Maynooth 10000, Ireland (e-mail: tianji.li@nuim.ie; david.malone@nuim.ie; doug.leith@nuim.ie).

Q. Ni was with Hamilton Institute, National University of Ireland, Maynooth 10000, Ireland. He is now with the School of Engineering & Design, Brunel University, Uxbridge UB8 3PH, U.K. (e-mail: qiang.ni@brunel.ac.uk).

Y. Xiao is with the Department of Computer Science, University of Alabama, Tuscaloosa, AL 35487-0290 USA (e-mail: yangxiao@ieee.org).

T. Turletti is with the PLANÈTE Project, INRIA Sophia Antipolis, 06902 Sophia Antipolis Cedex, France (e-mail: thierry.turletti@sophia.inria.fr).

Digital Object Identifier 10.1109/TNET.2009.2014654

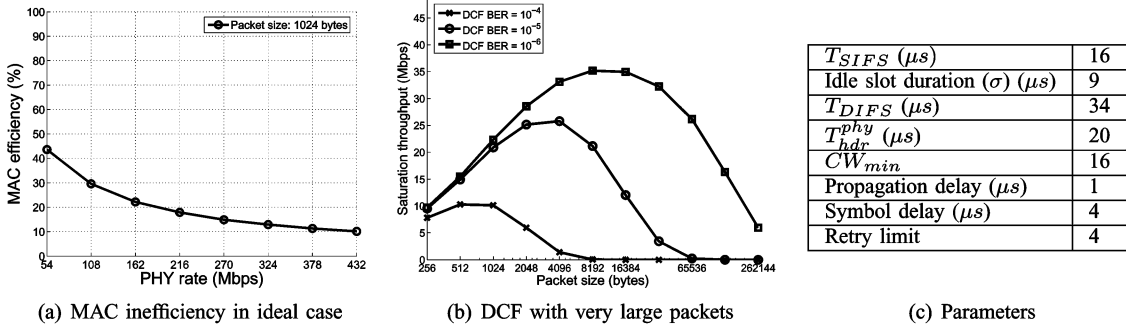


Fig. 1. (a) Legacy DCF efficiency in the ideal case with a 1024-B frame size. The  $x$ -axis represents the PHY data rate. The  $y$ -axis represents the ratio of the ideal throughput to the PHY rate. (b) Large frames transmission in DCF where PHY rate is 54 Mbps. (c) MAC and PHY parameters used.

TABLE I  
NOTATION

$n$	Number of STAs
$M$	Number of packets in a frame
$m$	Number of fragments in a frame
$m'$	Number of fragments in a packet
$T_{CW}$	Contention time
$T_{SIFS}$	Time duration of SIFS
$T_{DIFS}$	Time duration of DIFS
$T_{ack}$	Overhead for transmitting an ACK frame <sup>a</sup>
$T_{EIFS}$	Time duration of EIFS <sup>b</sup>
$T_{hdr}^{phy}$	Time duration to transmit the PHY headers of one frame
$T_{hdr}^{mac}$	Time duration to transmit the MAC headers of one frame
$T_{hdr}^{frag}$	Time duration to transmit the fragment headers of one frame
$T_p$	Time duration to transmit one packet
$T_f$	Time duration to transmit payload of one frame
$T_{oh}^p$	Overhead for transmitting one packet
$T_{oh}^f$	Overhead for transmitting payload of one frame
$\delta$	Propagation delay
$\sigma$	PHY layer time slot
$L_f$	Payload size in one frame (bytes)
$L_p$	Packet size (bytes)
$L_{frag}$	Fragment size (bytes)
$L_1$	Fragment header size (bytes)
$L_{hdr}^{mac}$	Aggregate size of all MAC headers in one frame (bytes)
$L_{hdr}^{frag}$	Aggregate size of all fragment headers in one frame (bytes)
$L_{FCS}$	FCS size (bytes)

## II. MOTIVATION

### A. DCF and Its Inefficiency

Transmission of a frame inevitably carries an overhead,<sup>2</sup> which we can consider as additional time  $T_{oh}^p$ . In 802.11, the overhead includes the time  $T_{hdr}^{phy}$  required to transmit the PHY header, the time  $T_{hdr}^{mac}$  to transmit the MAC header, the CSMA/CA backoff time  $T_{CW}$ , and the time  $T_{ack}$  to transmit a MAC ACK (Notation is listed in Table I).

In order to clarify the impact caused by the overhead, we define MAC efficiency as

$$\eta = \frac{T_p}{T_p + T_{oh}^p} \quad (1)$$

where  $T_p$  is the time required to physically transmit a packet (i.e., the frame payload) and  $T_{oh}^p = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{CW} + T_{ack}$ , as just explained above. As the PHY rate  $R$  increases, for a fixed packet size  $L_p$ , the time  $T_p = L_p/R$  to transmit the packet payload decreases. If  $T_{oh}^p$  does not also decrease, then the efficiency  $\eta \rightarrow 0$  as  $R \rightarrow \infty$ .

As the PHY rate increases, the contention time  $T_{CW}$  does not decrease toward zero due to the constraints placed on the minimum slot size by clock synchronization requirements and on DIFS by the need for backward compatibility. Similarly, the duration of the PHY header is not expected to decrease with increasing PHY rate owing to backward compatibility and PHY-layer channel equalization requirements [4]. Thus, as the PHY rate is increased, the time to transmit a frame quickly becomes dominated by the fixed overhead associated with the PHY header, contention time, etc. Much work has been done to minimize the contention time component of the overhead by regulating the randomized backoff process (e.g., [16], [49], and [33]) to reduce the number of collisions and idle slots. However, in very high-speed networks, the MAC efficiency is still intolerable even without these problems. For example, we illustrate in Fig. 1(a) the efficiency in the ideal case where the channel is perfect with neither collisions nor errors [48], hence the overhead of the backoff process is minimized. It can be seen that the efficiency decreases dramatically as the PHY rate increases. In a 216-Mbps WLAN, the efficiency is only about 20%. When the PHY rate increases to 432 Mbps, the efficiency decreases to around 10%.

<sup>2</sup>In the DCF scheme, there is only one packet in each frame, so the packet size and the payload size of one frame are the same.

where frames are transmitted immediately once the MAC wins a transmission opportunity. In a zero-waiting aggregation scheme, the frame sizes adapt automatically to the PHY rate and the channel state, thereby maximizing the MAC efficiency while minimizing the holding delay.

Thirdly, we investigate by simulations the impact of AFR on the performance of realistic applications with diverse demands. For this, we follow the 802.11n usage model [8]. We implement the AFR scheme in the network simulator NS-2 and present detailed results for TCP, VoIP, and HDTV traffic. Results suggest that AFR is a promising MAC technique for very high-speed WLANs. Moreover, AFR is particularly effective for rich multimedia services with high data rates and large packet sizes, which are expected to be key applications in future WLANs.

The remainder of the paper is organized as follows. Section II details the motivation of this work. We identify the fundamental properties that must be satisfied by any aggregation schemes in Section III, and introduce the AFR scheme in Section IV. A theoretical analysis is given in Section V, while Section VI presents detailed simulation results. Finally, we summarize our conclusions in Section VIII.

### B. Burst ACK and Block ACK

The Burst ACK (e.g., [37], [42], and [43]) and Block ACK (e.g., [3] and [48]) schemes have been proposed in the literature for improving efficiency. Burst ACK performs the backoff process once for a series of data and ACK frames (see Fig. 7 for details), while Block ACK goes one step further by using a single ACK frame for multiple data frames, thus reducing the number of ACKs and SIFS.

In both schemes, the backoff time  $T_{CW}$  is incurred once for  $M$  packet transmissions, where  $M$  is the size of a packet burst. With Burst ACK, the per-packet overhead is approximately  $T_{oh}^p = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{CW}/M + T_{ack}$ , while for Block ACK it is  $T_{oh}^p = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{CW}/M + T_{ack}/M$ . It can be seen that the contention overhead  $T_{CW}$  and MAC ACK overhead  $T_{ack}$  are amortized over multiple packets by these two schemes, therefore improving efficiency.

However, the per-packet PHY header overhead  $T_{hdr}^{phy}$  and the MAC header overhead  $T_{hdr}^{mac}$  are left untouched. According to the proposal 802.11n [4] for the future WLANs, it is likely to take at least 44  $\mu$ s to transmit a PHY header (and 48  $\mu$ s when two antenna radios are used [4]). For comparison, the transmission duration of a 1024-B frame at a PHY rate of 216 Mbs is 40  $\mu$ s, and at 432 Mbs is 20  $\mu$ s. As the PHY rate is increased, the time to transmit a frame quickly becomes dominated by PHY headers, the MAC efficiency rapidly decreases, and efforts to increase the system capacity purely by increasing the data rate are thus of limited effectiveness even when Burst ACK or Block ACK are employed.

### C. Aggregation Schemes

Aggregation schemes seek to amortize the PHY header overhead across multiple packets. This is achieved by transmitting multiple packets in a single large frame. However, there is a traditional dislike for transmitting large frames in wireless networks since in a noisy channel (e.g.,  $BER \geq 10^{-5}$ ), the throughput can fall as larger frames are used [24]. We illustrate this in Fig. 1(b). However, we note that in traditional retransmission schemes, a whole frame is retransmitted even if only one bit is lost. This raises the question of whether it is possible to retransmit only the erroneous part(s) of a frame—if properly designed, such partial retransmission could be expected to improve performance. This is a key motivation of the work presented here.

Although this idea seems simple at first glance, it is actually a radical challenge for PHY and MAC technology. From the PHY viewpoint, the traditional small-packet rule does not hold anymore. The PHY layer has to transmit very large frames and continue decoding even if the BER exceeds some previously unacceptable value. Under these conditions, the size of the largest practical frame is still unknown [4]. From the MAC viewpoint, any retransmission scheme carries an associated signaling overhead and, hence, a tradeoff exists between system efficiency and the granularity of retransmission. Moreover, since real traffic is typically bursty/on-off in nature, this raises questions as to the optimal policy for aggregating packets into frames—for example, how much time should the MAC wait for sufficient packets to arrive to form a large frame.

Our previous work on aggregation schemes resulted in a proposal for the forthcoming IEEE 802.11n standard. In [5] and [26], we propose to aggregate multiple packets into a single

large frame and, should an error occur, the damaged packets are retransmitted. The present paper substantially extends this previous work (see Section II-D). In parallel with our work, there are other activities in the 802.11n standard working group on this topic (e.g., [4], [6], and [7]). These support similar functionalities to our scheme, with a special *delimiter* for locating each fragment in a frame. Other related work includes that of Ji *et al.* [22], where an aggregation technique is used to solve an unfairness problem in WLANs. Ji *et al.* suggest removing the DIFS, SIFS, and backoffs before a series of packets, and transmitting the packets together in a large PHY layer frame. However, a small PHY header (12  $\mu$ s) is used to identify each packet within a frame. In [23], a two-level (one at MAC, another at PHY) aggregation scheme is proposed that uses a similar *delimiter* to that in the TGn Sync proposal [4].

### D. Open Questions

Although aggregation is not a new idea, many fundamental questions remain open.

- How do we aggregate packets? The frames we want are larger than typical packets. If the packets from the upper layer are large and arrive rapidly, then aggregation is simple. If not, should a timing mechanism be used to wait for sufficient packets to arrive to form a large frame? If so, how much time do we wait to maximize throughput while minimizing delay?
- What is an appropriate (re)transmission unit? Should very large packets be divided for retransmissions?
- What is the aggregation throughput and delay performance?
- How does packet aggregation impact real world traffic, e.g., voice, video and TCP traffic?

We address these open questions in this paper.

## III. FUNDAMENTAL CONSIDERATIONS

We highlight in this section the basic requirements that must be respected by any aggregation schemes that seek to maintain high MAC efficiency as PHY rates increase and introduce the zero-waiting approach to aggregation.

### A. MAC Efficiency

The basic requirement for high efficiency is to aggregate packets into large frames so as to spread the cost of fixed overhead across multiple packets. To reduce the overhead associated with transmission errors, each frame is subdivided into fragments, with packets that exceed the fragment size being divided. Fragments are the unit used in the retransmission logic, i.e., damaged fragments rather than the entire frame are retransmitted.

The time to transmit a packet is  $T_p = L_p/R$ , where  $L_p$  is the packet size and  $R$  is the PHY rate. Hence, the per-packet MAC efficiency is

$$\eta_p = \frac{T_p}{T_p + T_{oh}^p} = \frac{L_p/R}{L_p/R + T_{oh}^p}. \quad (2)$$

We can see that  $T_p = L_p/R$  scales with  $1/R$ . We show that under certain assumptions, it is indeed possible to maintain a constant MAC efficiency while  $R$  is increased. That is, we may decouple MAC efficiency from the PHY rate  $R$ . In order to maintain MAC efficiency  $\eta_p$ , we require that the per packet

overhead  $T_{oh}^p$  also scales with  $1/R$ . Considering  $T_{oh}^p$  in more detail, we can typically decompose it into the following elements (where  $r$  denotes the average number of transmissions before all fragments from this packet are transmitted successfully, and other notation is listed in Table I):

$$T_{oh}^p = \frac{(T_{hdr}^{phy} + T_{hdr}^{mac} + T_{hdr}^{frag} + T_{CW} + T_{ack}) \cdot r}{M}. \quad (3)$$

To ensure that  $T_{oh}^p$  scales with  $1/R$ , we require that:

- The number of packets  $M$  in a frame should be proportional to  $R$ , that is  $M = bR$  for some constant  $b$ . This ensures that the overhead  $T_{hdr}^{phy}$ ,  $T_{hdr}^{mac}$ ,  $T_{ack}$ , and  $T_{CW}$  translate into a per-packet overhead that scales with  $R$ .
- Since there is only one MAC header and one ACK per frame, when  $M$  is proportional to  $R$ , there is no fundamental constraint on the rate at which MAC headers and ACK frames are transmitted. The same is not true for fragment headers.
- For a given fragment size  $L_{frag}$ , the number of fragments in a frame  $m$  increases with the number of packets  $M$  in a frame, i.e.,  $m = m' M$ , where  $m'$  is the number of fragments per packet; we thus have  $m = m' b R$  when  $M = b R$ . Hence, for  $T_{hdr}^{frag}/M$  to scale with  $1/R$ , the rate at which fragment headers are transmitted must be chosen to be proportional (see *comment 1*) to  $R$ , in which case  $T_{hdr}^{frag}/M = m L_1/R = m' L_1/R$ .
- The retransmission time  $r$  is constant. For a given packet size, the number  $r$  is determined by the bit error rate (BER). The BER itself depends upon the channel signal to noise ratio and the choice of coding. A rate controller is typically used to adjust the coding and rate  $R$  to maintain the BER below a target level [35], [36], reflecting application and transport layer requirements.<sup>3</sup> In the following we assume the use of a rate controller, and thus that rate is adjusted to ensure that the average number of retransmissions  $r$  remains approximately constant. We also note that if BER is not regulated via rate control, then provided  $r$  is bounded or is a known function of rate  $R$ , the scaling analysis can be extended to include this situation.

When the per-packet overhead satisfies these conditions, the per-packet MAC efficiency is

$$\eta_p = \frac{L_p}{L_p + r(a/b + m' \cdot L_1)} \quad (4)$$

where  $L_1$  denotes the size of one fragment header and  $a = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{CW} + T_{ack}$ .

First, observe that the efficiency is nicely decoupled from the PHY rate  $R$ , i.e., the throughput scales with  $R$ . Second, as we increase the factor  $b$ , we can see that the efficiency asymptotically tends to

$$\tilde{\eta}_p = \frac{L_p}{L_p + r \cdot m' \cdot L_1} = \frac{1}{1 + d} \quad (5)$$

where  $d = (r m' L_1)/L_p$ .

That is, the efficiency is fundamentally limited by the number of fragments per packet  $m'$  and the number of retransmissions

<sup>3</sup>For example, since TCP congestion control views packet losses as an indicator of congestion, TCP throughput is strongly dependent on the link loss rate, (e.g., [14], and [15]) and too high a loss rate may then prevent high utilization of the wireless channel.

$r$ . In particular, if we use a large fragment size corresponding to a small  $m'$ , such large fragments are more likely to be corrupted; we have therefore small  $m'$  and large  $r$ . On the other hand, when a packet is divided into many small fragments, corresponding to use of a large  $m'$ , the probability of a fragment being corrupted is low, and we have large  $m'$  but small  $r$ . To achieve high efficiency, we study in Section V-D a fragmentation technique where packets with sizes exceeding a threshold are divided into fragments to deal with the tradeoff between  $m'$  and  $r$ .

*Comment 1:* At high rates in a noisy channel the question of the impact of errors in the received fragment headers arises. First, we only require that the rate used for sending the fragment headers is proportional to the data rate  $R$ . Thus, to protect the fragment headers, they may be sent at a relatively low rate (e.g., at half, or less, of the data rate), and in this way we can ensure that the majority of bit errors affect the data payload only. Second, fragment header size (8 B in AFR, see Section IV-A) is minimized to ensure low error probabilities. Third, in the frame we collect the fragment headers together with the MAC header (details in Section IV-A) so that FEC can be more easily employed to enhance robustness.

## B. Zero-Waiting

When the channel is lightly loaded to the extent that the DCF is enough, deliberate waiting only leads to higher delays. If the channel is in a heavily loaded condition where backlogged buffers mean the desired numbers of packets to form large frames are always available when transmission opportunities are won, then all waiting schemes are the same. If the channel is in an intermediate situation between these two extremes, waiting for a certain amount of time for packets to accumulate seems reasonable at first glance. Nevertheless, we argue that fundamentally there is no need to wait for packets to accumulate at the MAC layer, and it is sufficient instead to simply start a transmission whenever the MAC wins a transmission opportunity. This zero-waiting mechanism evidently performs well in both lightly and heavily loaded situations. In the intermediate state,<sup>4</sup> the frame size used adapts to the minimum required to service the offered load. Specifically, when the current level of efficiency is too low for the offered load, a queue backlog will develop, which in turn induces larger frames and increased efficiency. If the incoming traffic subsides, smaller frame sizes will be automatically selected. Evidently, such a policy minimizes holding delay at the MAC layer.

We now show that this opportunistic aggregation policy can also lead to the desired efficiency  $\tilde{\eta}_p$  where it is feasible to do so. Assuming that there are no collisions and errors in the network,<sup>5</sup> corresponding to  $r = 1$ , we can write the per frame MAC efficiency as

$$\eta_f = \frac{T_f}{T_f + a + d \cdot T_f} = \frac{1}{1 + d + a/T_f}. \quad (6)$$

Write the desired throughput as  $\tilde{S} = R/(1+d)$ . Let the mean arrival rate of the offered load be  $\nu = \alpha \tilde{S} = \alpha R/(1+d)$  b/s,

<sup>4</sup>We note that this simple zero-waiting scheme is also attractive from a practical point of view, as it has been observed that real-world traffic can exhibit complex bursty behaviors [34], [40] that make the effective design of a more complex waiting scheme difficult.

<sup>5</sup>Proof for more complicated cases is left as further work.

where  $0 \leq \alpha \leq 1$  is a real valued factor. During the time  $T_f + a + d \cdot T_f$  to transmit a frame, on average we expect  $\nu \cdot (T_f + a + d \cdot T_f)$  arrivals at the queue. Selecting the frame size to be the same as queue size  $q(k)$ , we have that

$$\begin{aligned} E[q(k+1)] &= \nu \cdot [T_f + a + d \cdot T_f] \\ &= \nu \cdot [(1+d)E[q(k)]/R + a] \\ &= \alpha \cdot E[q(k)] + \frac{\alpha \cdot a \cdot R}{1+d}. \end{aligned} \quad (7)$$

These queue dynamics can be written as

$$E[q(k+t)] = \alpha^t E[q(k)] + \sum_{i=1}^t \alpha^{i-1} \cdot \frac{\alpha \cdot a \cdot R}{1+d}.$$

Hence, provided  $\alpha < 1$ , then as  $t \rightarrow \infty$ , we have that the queue dynamics are stable. Asymptotically, we have that

$$E[L_f] = E[q] = \frac{\alpha \cdot a \cdot R}{(1-\alpha)(1+d)}. \quad (8)$$

Combining (6) and (8), we have that

$$\eta_f = \frac{\alpha}{1+d} = \alpha \cdot \tilde{\eta}_p.$$

As  $\alpha \rightarrow 1$ , we can see that the zero-waiting policy achieves the desired efficiency.

From (8), we can see two important features of zero-waiting. First, when the offered load is light (i.e.,  $\alpha$  is small), small frames will be used. As the load increases, larger frame sizes will be automatically selected. Thus, zero-waiting elegantly creates a feedback loop whereby MAC efficiency is regulated based on queue backlogs as expected. Second, for a given level of load  $\alpha$ , the frame size  $L_f$  scales with  $R$ . Therefore, with a multirate enabled wireless card, the frame size also adapts automatically to the changing PHY rate  $R$ .

#### IV. THE AFR SCHEME

In this section, we describe in detail the AFR scheme based on the insight provided by the foregoing analysis.

##### A. AFR Implementation

Clearly, new data and ACK frame formats are a primary concern in developing a practical AFR scheme. Difficulties for new formats include 1) respecting the constraints on overhead noted previously and 2) ensuring that, in an erroneous transmission, the receiver is able to retrieve the correctly transmitted fragments—this is not straightforward because the sizes of the corrupted fragments may be unknown to the receiver. In our scheme, a MAC frame consists of a frame header and a frame body [see Fig. 2(a)]. In the new MAC header, all the fields of the DCF MAC header remain unchanged, and we add three fields—*fragment size*, *fragment number*, and a *spare* field. The *fragment size* represents the size of fragment used in the MAC frames. The *fragment number* represents the number of fragments in the current MAC frame. The *spare* field is left for future extension and maintaining alignment. The frame body consists of fragment headers, fragment bodies and the corresponding frame check sequences (FCS) [see Fig. 2(b) and (c)].

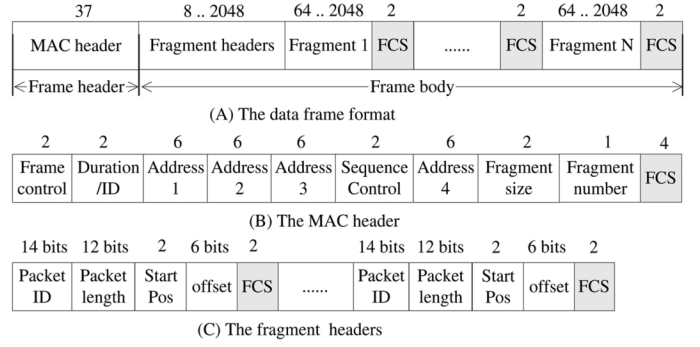


Fig. 2. Data format in the AFR scheme.

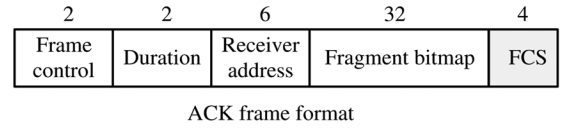


Fig. 3. ACK format in the AFR scheme.

TABLE II  
AN EXAMPLE USAGE OF THE AFR FRAME FORMATS

	packet ID	packet length	StartPos	offset
fragment 1	1	1025	0	0
fragment 2	1	1025	512	1
fragment 3	1	1025	1024	2
fragment 4	2	40	1025	0

The fragment header section of the frame body has a variable size. It includes from 1 to 256 fragment headers, each of which is protected by a FCS. The length of each fragment header is constant (8 B) and known to both the sender and the receiver. For the receiver, it knows from where the first fragment header starts and what the fragment header size is, thus it can locate all the fragments in the frame even if some of them are corrupted during the transmission.

Each fragment header is composed of six fields: packet ID (*pID*), packet length (*pLEN*), *startPos*, *offset*, *spare*, and *FCS*. *pID* and *pLEN* represent the corresponding ID and length of the packet *P* to which this fragment belongs. *startPos* is used to indicate the position of the fragment body in this frame, and *offset* is used to record the position of this fragment in packet *P*.

The new ACK format is simple; we add a 32-B bitmap in the legacy ACK format. Each bit of the bitmap is used to indicate the correctness of a fragment (see Fig. 3).

To clarify the usage of the new formats, we give an example. Suppose there are two packets (*pkt*<sub>1</sub> and *pkt*<sub>2</sub>) with lengths of  $L_{p1} = 1025$  B and  $L_{p2} = 40$  B. The frame length is  $L_f = 2048$  B and the fragment length is  $L_{frag} = 512$  B.<sup>6</sup> Then, AFR divides *pkt*<sub>1</sub> and *pkt*<sub>2</sub> into 3 and 1 fragments, respectively, and puts them into the sending queue. A frame with *fragment size* of 512 B and *fragment number* of 4 is constructed. The corresponding fragment headers are shown in Table II. After receiving the frame, the receiver operates as shown in Algorithm 1 to recover the fragments.

<sup>6</sup>To show that AFR can support arbitrary sizes of fragments, we do not restrict ourselves in this example to the fragmentation algorithm introduced in Section V-D.

**Algorithm 1:** Pseudo Code of the receiver's running logic

---

```

1: if MAC header is correct then
2:   for  $i = 0$  to  $fragment\ number - 1$  do
3:     if Fragment  $i$ 's header is correct then
4:       if  $packet\ length < fragment\ size$  then
5:         fragment  $i$ 's length =  $pLEN$ ;
6:       else if  $offset = \lfloor pLEN/fragment\ size \rfloor$  then
7:         fragment  $i$ 's length =  $pLEN - offset * fragment\ size$ ;
8:       else
9:         fragment  $i$ 's length =  $fragment\ size$ ;
10:      end if
11:      fragment start position =  $startPos$  in the fragment
        header.
12:      check the correctness of the fragment body using the
        FCS of it.
13:    end if
14:    record correctness (including fragment header and
      fragment body) of the fragments in a data structure called
      the ACK bitmap.
15:  end for
16:  construct ACK frame using the ACK bitmap and send
    it back.
17:  update the receiving queue according to the ACK bitmap.
18:  check the receiving queue and transfer all correctly received
    packets upwards, and remove them from the receiving
    queue.
19: else
20:  discard this frame and defer an EIFS before next
    transmission.
21: end if

```

---

**B. Comments**

1) *Frame/Fragment Size*: Selection of the maximum frame size and of the near-optimal fragment size is discussed in Sections V-C and D.

2) *Fairness*: AFR strictly follows the basic principle of the CSMA/CA. Therefore, the same fairness characteristics hold as in the legacy DCF. Techniques to improve DCF's fairness are all suitable for AFR. Interested readers can refer to [13], [38], and [20].

3) *Multiple Destinations*: Thus far, we have focussed only on aggregation between a single source–destination pair. This facilitates a clear understanding of the pros and cons of the aggregation itself. In order to support one-to-many aggregation, a broadcast/multicast MAC address should be used and all stations that hear the transmission then check a new receiver-list

field in the MAC header that specifies the destination address for each fragment. That is, the only modification in terms of frame format is adding the receiver-list field. However, one-to-many aggregation introduces a number of new issues that we will mention. Resolving these issues is beyond the scope of this paper.

First, one-to-many aggregation requires consideration of new ACKing techniques to avoid collisions between ACK transmissions by the multiple receivers. This might be achieved by sequential transmission of ACKs or perhaps by use of advanced PHY layer techniques (coding, multiple antennas) to enable decoding of ACKs that are sent simultaneously (e.g., [18] and [41]). The resulting performance requires detailed study, and these techniques are not proposed for future 802.11n standards [4].

Second, multiple antenna systems are widely considered to be of vital importance for achieving very high transmission rates [4]. The design of one-to-many aggregation for multiple antenna systems remains an open question that is likely to require tightly coupled cross-layer PHY/MAC design and operation.

Third, the channel quality may differ between neighbors, and it might therefore be necessary to use multiple subphysical headers. These new headers clearly would cause extra overhead. Further, rate adaptation, which has become an indispensable functionality of 802.11-based networks, requires further work in the context of one-to-many aggregation.

4) *Multirate*: In the current WLANs, a commonly used technique to resist channel noise is to lower the PHY rate after measuring a high packet (or bit) error rate, and when the channel state improves, the PHY layer increases its rate accordingly. There are two issues to be addressed if multirate is to be supported in AFR: i) Should we change the frame size with the PHY rate? ii) Should we support one-to-many aggregation where receivers have different channel states? The first issue has been discussed at the end of Section III-B, and the second one is just mentioned.

**V. THEORETICAL ANALYSIS**

Building on previous modeling work [12], [28], [30], [32], [45], we develop a model in this section and use it to analyze the saturation throughput and delay of the AFR scheme over noisy channels.

**A. Model**

We assume that readers are familiar with the Bianchi model [12] and explain only the differences between our model and that of Bianchi. We say a station is saturated if, whenever the MAC layer needs a frame to transmit, it can always fill a long enough frame without waiting. The saturation throughput  $S$  is defined as the expected payload size of a successfully transmitted frame  $E[L_f]$  in an expected slot duration  $E[T]$ , i.e.,  $S = \frac{E[L_f]}{E[T]}$ . We first compute the expected state duration  $E[T]$ . Altogether, there are three kinds of events in the AFR scheme (notation is listed in Table I).

- *Idle duration  $T_I$* : When all STAs are counting down, no station transmits a frame and we have

$$T_I = \sigma. \quad (9)$$

- *Success/Error duration*  $T_3$ : When a frame is successfully transmitted or it is corrupted due to channel noise,<sup>7</sup> the slot duration is the sum of a frame, a SIFS, and an ACK duration

$$T_3 = T_{hdr}^{phy} + T_f + T_{ack}. \quad (10)$$

- *Collision duration*  $T_C$ : When two or more stations transmit at the same time, a collision occurs. In this case, the sender waits for an EIFS before the next transmission, and so

$$T_C = T_{hdr}^{phy} + T_f + T_{EIFS}. \quad (11)$$

The expected state duration is  $E[T] = P_I T_I + P_3 T_3 + P_C T_C$ , where  $P_I$ ,  $P_3$ , and  $P_C$  are the probabilities of *Idle*, *Success/Error*, and *Collision* events, respectively. Let  $\tau$  denote the STA transmission probability and  $n$  the number of STAs in the system. We have that

$$P_I = (1 - \tau)^n \quad (12)$$

$$P_3 = \binom{n}{1} \tau (1 - \tau)^{n-1} \quad (13)$$

$$P_C = 1 - P_I - P_3. \quad (14)$$

Letting  $p_f$  denote the probability of doubling the contention window after a transmission,  $\tau$  can be expressed as a function of  $p_f$  using a Markov chain similar to that of Bianchi's. In more detail, Bianchi's model assumes there are no errors in the channel, so  $p_f = p_c = 1 - (1 - \tau)^{n-1}$  where  $p_c$  is the STA collision probability. However, we are interested in noisy channels. In this case, if the contention window is reset after an erroneous transmission, then  $p_f = p_c$ ; if the contention window is doubled, then  $p_f = p_c + p_e - p_c \cdot p_e$ , where  $p_e$  stands for the frame error rate. In the AFR scheme, the receiver sends back the ACK frame in both successful and erroneous cases, thus  $p_f = p_c$  and the Bianchi's formula could in fact be applied without change. We note that Bianchi assumes a frame can be retransmitted infinite times, which is inconsistent with the 802.11 specification [1]. Wu *et al.* relax this assumption [45], and thus we use [45, eqs. (8) and (9)] for greater accuracy.

Solving for  $\tau$ , we can obtain the saturation throughput  $S_{AFR}$  of the AFR scheme from

$$S_{AFR} = \frac{P_3 \cdot E[L]}{P_I T_I + P_3 T_3 + P_C T_C}. \quad (15)$$

Note that  $E[L]$  is not the frame payload size but rather the expected number of successfully transmitted bits; recall that the AFR scheme allows successfully transmitted fragments to be received even if some fragments within a frame are corrupted. We calculate  $E[L]$  as follows. Let  $i$  denote the number of erroneous fragments, and  $m$  denote the number of fragments in a frame. Assuming independent and identically distributed errors

$$E[L] = \sum_{i=0}^m \binom{m}{i} \cdot (p_e^{frag})^i \cdot (1 - p_e^{frag})^{m-i} \cdot (L_f - i \cdot L_{frag}), \quad (16)$$

<sup>7</sup>Recall that, in the AFR scheme, we consider frames that are partially corrupted by channel noise as successful transmissions.

and the fragment error rate  $p_e^{frag}$  is

$$p_e^{frag} = 1 - (1 - p_b)^{L_{frag} + L_{FCS}}, \quad (17)$$

where  $L_{frag}$  and  $L_f$  are the lengths of a fragment and the length of payload of a full frame, respectively, and  $p_b$  is the BER.

Let  $\Delta = \binom{m}{i} \cdot (p_e^{frag})^i \cdot (1 - p_e^{frag})^{m-i}$ . We have that

$$E[L] = \sum_{i=0}^m [\Delta \cdot (L_f - i \cdot L_{frag})] = L_f \cdot (1 - p_e^{frag}) \quad (18)$$

We thus have that

$$S_{AFR} = \frac{P_3 \cdot L_f \cdot (1 - p_e^{frag})}{P_I T_I + P_3 T_3 + P_C T_C}. \quad (19)$$

This model is validated against NS-2 simulations. Both simulation and model results are shown in Fig. 4(a). As we can see from the results, the analysis and simulation results match well.

### B. Improvements Over DCF

For comparing the AFR and DCF performance, a model for the latter is required. We use the DCF model that has been developed and validated in our previous work [32]. It can be seen from Fig. 4(b) that AFR fundamentally changes the throughput scaling behavior in a noisy channel. Specifically, the DCF throughput exhibits a maximum value as the frame size is varied, with the maximum depending on the BER. This arises because while increasing the frame size tends to increase the throughput, the probability of a frame being corrupted by noise also increases, thereby tending to decrease throughput. The interaction of these two effects therefore leads to the existence of optimal sizes of frames that depend on the BER. In contrast, the AFR throughput *increases monotonically* with frame sizes even when the channel is noisy. The resulting gain in throughput compared to DCF is dramatic. For example, DCF achieves almost zero throughput for a frame size of 8192 B in a channel with BER of  $10^{-4}$ , while AFR achieves around 30 Mbps throughput under the same conditions.

Fig. 4(c) plots the MAC efficiency  $\left(\frac{\text{Throughput}}{\text{PHYRate}} \cdot 100\%\right)$  of the DCF and AFR schemes as PHY rate is increased and a chosen frame size of 65 536 B is used.<sup>8</sup> It can be seen that whereas the DCF efficiency rapidly decreases with increasing PHY rate (falling from 42% at 54 Mbps to less than 10% at 432 Mbps), the AFR efficiency is approximately constant with increasing PHY rate as discussed above. Observe that the efficiency falls with increasing BER as expected, but the efficiency remains relatively high even under noisy conditions, e.g., achieving approximately 70% MAC efficiency for a BER of  $10^{-5}$  and 60% efficiency for a BER of  $10^{-4}$ .

### C. Maximum Frame Size

It can be seen in Fig. 4(b) that the AFR throughput asymptotically approaches a maximum value as the frame size is increased. We can determine this asymptotic value analytically

<sup>8</sup>Note that 65 536 is a power of two (i.e.,  $65\,536 = 2^{16}$ ). The reason of selecting power of two for simulating is that these sizes are more practical for binary implementation.

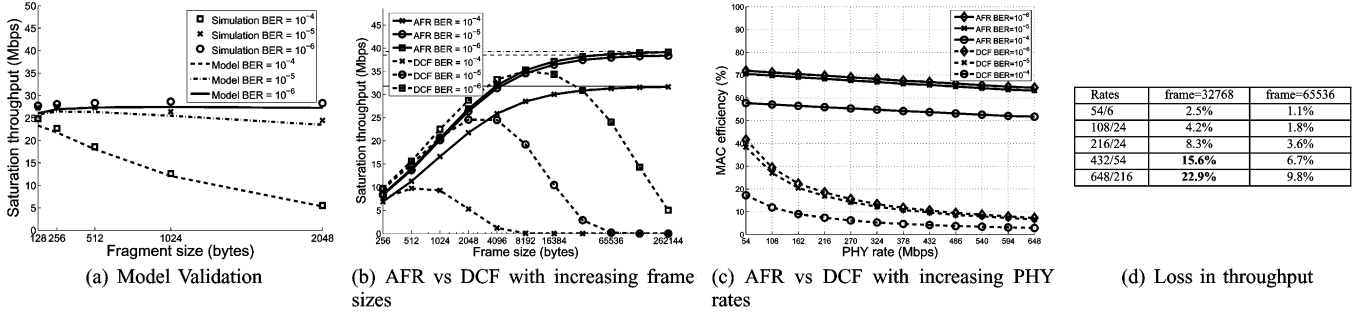


Fig. 4. (a) AFR: model versus simulations. (b) The influence of frame sizes. (c) AFR versus DCF with increasing PHY rates. (d) In the first column, the PHY rates are on the left of the slash, the basic rates are on the right. The unit of the rates is Mbps. The values in the second and the third columns are differences between the throughput under the rates in the first column and the maximum throughput. Other parameters are listed in Fig. 1(c) and Table III.

TABLE III  
PARAMETERS USED IN THE THEORETICAL MODEL AND ITS VALIDATION

	Fig. 4(a)	Fig. 4(b)	Fig. 4(c)	Fig. 5	Fig. 6(a)	Fig. 6(b)	Fig. 8(a)	Fig. 8(b) & 9
Number of STAs ( $n$ )	10	10	10	10	10	10	10	10
Application rate (Mbps)	54	54	$=R$	54	432	432	432	$=R$
Data rate (Mbps) ( $R$ )	54	54	varied	54	432	432	432	varied
Basic rate (Mbps)	6	6	$=R$	6	54	54	54	$=R$
AFR sending queue (pkts) <sup>a</sup>	200	N/A	N/A	N/A	N/A	N/A	N/A	N/A
AFR IFQ (pkts) <sup>b</sup>	200	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Packet (B) <sup>c</sup>	2048	$=L_f$	1024	$=L_{frag}$	2048	2048	2048	1024
Frame (B) ( $L_f$ )	2048	256, ..., 262144	65536	8192	65536	4096	8192	varied
AFR fragment (B) ( $L_{frag}$ )	128, ..., 2048	256	256	32, ..., 8192	256	256	256	256

<sup>a</sup>AFR sending queue is the queue at MAC layer to temporarily store the packets from the AFR IFQ in AFR's simulations.

<sup>b</sup>AFR IFQ is the queue between the MAC and its upper in AFR's simulations.

<sup>c</sup>Here, B refers to bytes.

as follows. As the frame size  $L_f \rightarrow \infty$ , we have that (since  $T_3 = T_C$ )

$$\begin{aligned}
 S_{AFR} &\rightarrow \frac{P_3 \cdot (1 - p_e^{frag})}{(P_3 + P_C) \cdot T_3 / L_f} \approx \frac{P_3 \cdot (1 - p_e^{frag})}{(1 - P_I) \cdot T_f / L_f} \\
 &= \frac{R \cdot P_3 \cdot (1 - p_e^{frag})}{(1 - P_I)}.
 \end{aligned} \quad (20)$$

Using this equation, the asymptotic values are 39.30, 38.55, and 31.78 Mbps for  $BER = 10^{-6}$ ,  $BER = 10^{-5}$  and  $BER = 10^{-4}$ , respectively. These values are marked by horizontal lines in Fig. 4(b).

In practice, of course, arbitrarily large frame sizes are often not feasible. The upper limit on frame sizes depends on the PHY's abilities and is also constrained by interface memory and the size of the STA's sending buffer. Fortunately, it can be seen in Fig. 4(b) that the gap between the maximum and actual throughput narrows rapidly with increasing frame sizes. The table in Fig. 4(d) gives the loss in throughput (compared to the maximum achievable throughput) versus the frame size for a range of data-rates. If we consider operation at 90% or higher of the maximum achievable throughput to be our target, it can be seen that a maximum frame size of 32 768 B is acceptable for data rates of up to 216 Mbps over a wide range of channel conditions, while a maximum frame size of 65 536 B is acceptable for data rates of up to 648 Mbps. We note that 65 536 B is also the maximum size proposed in TGN's 802.11n proposal [4].

#### D. Optimal Fragment Size

Fragmentation plays a central role in aggregation schemes, with fragments being the unit used for retransmission. When a very small fragment size is used, only corrupted bits are retransmitted, but since each fragment has a fixed size header, the overhead is relatively large. When a large fragment size is used, the overhead created by the fragment header is small, but many bits will be unnecessarily retransmitted since a single damaged bit in a fragment will lead to the entire fragment being retransmitted. For a given BER, there therefore exists an optimal fragment size that balances the tradeoff between the fragment header overhead and excessive retransmissions. Fig. 5(a) plots throughput versus fragment sizes from which the existence of an optimal fragment size that maximizes throughput is evident. Observe that the optimal fragment size depends on the BER, as expected (128, 512, and 1024 B for  $BER = 10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$ , respectively).

In practice, we are interested in determining a simple scheme that approximates the optimal fragment sizes' performance. It can be seen from Fig. 5(a) that the throughput peak is relatively flat and broad, and thus we expect that the throughput reduction resulting from an approximate scheme can be kept relatively small. Fig. 5(b) plots the reduction in throughput, compared to that achieved with the optimal fragment sizes, of using a sub-optimal fragment size. From this plot, we can see that if we can tolerate a throughput loss of up to 10%, then fragment sizes of 128 and 256 B are near-optimal across a wide range of BERs. Corresponding data for a range of PHY rates are summarized in Table IV. It can be seen that fragment sizes of 128 and 256 B are always able to achieve within 10% of the maximum possible throughput. We have obtained similar results under a wide



TABLE IV  
DIFFERENCES TO MAXIMUM THROUGHPUT IN DIFFERENT PHY LAYERS

Rates <sup>a</sup>	64 <sup>b</sup>	128	256	512
54/6	2.5%, 10.4%, <b>14.5%</b>	0.0%, 2.9%, 6.2%	6.6%, 0.0%, 2.3%	<b>28.2%</b> , 0.0%, 0.0%
108/24	1.8%, 9.4%, <b>13.2%</b>	0.0%, 2.7%, 5.7%	6.9%, 0.0%, 0.2%	<b>28.4%</b> , 0.0%, 0.0%
216/24	0.1%, 8.3%, <b>11.6%</b>	0.0%, 2.6%, 5.2%	6.9%, 0.0%, 1.6%	<b>28.8%</b> , 0.0%, 0.0%
432/54	0.0%, 7.0%, 9.9%	0.0%, 1.9%, 4.1%	7.7%, 0.0%, 1.3%	<b>30.2%</b> , 0.1%, 0.0%
648/216	0.0%, 5.5%, 8.7%	0.0%, 0.1%, 3.3%	8.8%, 0.0%, 1.6%	<b>31.2%</b> , 0.0%, 0.0%

<sup>a</sup>The PHY rates are on the left of the slash, and the basic rates are on the right. The unit of the rates is Mbps.

<sup>b</sup>The results are frames with 64-B fragments, under BER  $10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$ , respectively.

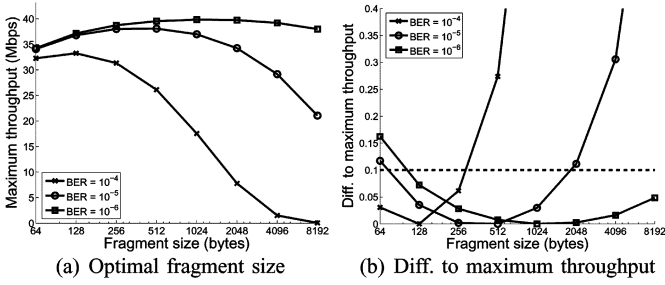


Fig. 5. The  $x$ -axis is fragment size, the  $y$ -axis of (a) is the absolute (i.e., always positive) difference between the throughput using the fragment size marked on the  $x$ -axis and the throughput when using the optimal fragment size. Other parameters are listed in Fig. 1(c) and Table III.

range of conditions including different numbers of stations, but these are not included here due to their similarity to the results in Table IV.

Based on these results, we propose a simple fragmentation algorithm: Namely, for a packet  $P$  with a size of  $L_p$ , find the  $m'$  which satisfies

$$(m' - 1) \cdot 256 + 1 < L_p \leq m' \cdot 256$$

where  $m' = 1, 2, \dots, 256$ . We divide  $P$  into  $m'$  fragments, each of which has a size in the range of  $\left(\frac{L_p}{m'}, \frac{L_p}{m'} + 1, \dots, \frac{L_p}{m'} + (m' - 1)\right)$ . In this way, the sizes of all fragments fall between 128 and 256 B. More importantly, the resulting sizes are almost the same. For example, a 257-B packet is divided into one 128-B and one 129-B fragment, rather than one 256-B and one 1-B fragment.

### E. RTS/CTS

Using large frames increases the duration of colliding transmissions, including collisions induced by hidden terminals. While consideration of hidden terminals is out of the scope of this paper, we consider the overhead associated with collisions. One technique for mitigating the duration of collisions is to probe the channel first using small packets so that losses only happen on small probing packets, thereby improving channel utilization. In Fig. 6, we use RTS/CTS as example probing packets to illustrate their impact on the AFR scheme. In Fig. 6(a), we vary the number of transmitting stations (the probability of a colliding transmission increases with the number of active stations) and use a fixed frame size of 65 536 B and fragment size of 256 B in line with Section V-C and D, respectively. We observe that enabling RTS/CTS consistently

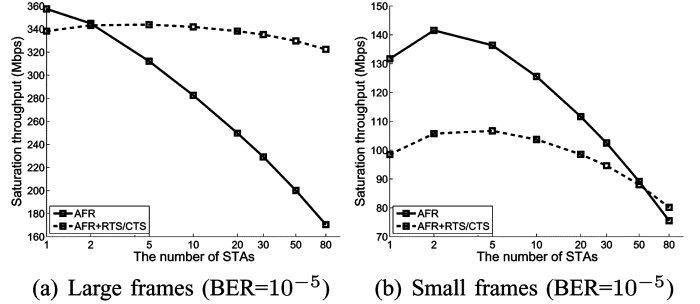


Fig. 6. AFR versus AFR + RTS/CTS. The frame sizes are 65 536 and 4096 B in (a) and (b), respectively. In both figures, packet/fragment sizes are 2048/256 B, and the PHY data/basic rates are 432/54 Mbps. Other parameters are listed in Fig. 1(c) and Table III.

results in significantly higher throughput when there is more than one station.

Nevertheless, using RTS/CTS adds a fixed extra overhead to each successful transmission, which can have negative impact on performance. There is, therefore, a tradeoff between reducing the duration of colliding slots and increasing the fixed overhead on successful transmissions. This can be seen in Fig. 6(b), where we compare AFR with and without RTS/CTS when the frame size is 4096 B. It can be seen that RTS/CTS starts to have positive impact only when there are more than 50 stations. In reality, an adaptive RTS/CTS mechanism is thus needed where RTS/CTS is enabled/disabled depending on frame sizes used and the channel load. We do not consider this in detail in the present paper due to space constraints.

### F. Comparison With Similar Schemes

In this section, we compare the throughput performance of AFR with four other schemes proposed in the literature: Burst ACK [37], [42], [43], Block ACK [3], [48], Packet Concatenation (PAC) [22], and Aggregation [23].

These schemes can be classified into two categories: 1) Burst ACK and Block ACK; 2) PAC, Aggregation, and AFR. The schemes in the first category transmit multiple frames at each transmission opportunity. The schemes in the second category transmit only one frame and use packet aggregation. AFR is the only scheme to use both fragmentation and aggregation. In the Burst ACK and Block ACK schemes, collisions lead to the whole Burst/Block being lost while errors lead to retransmission only of the corrupted packet. The PAC scheme is similar to our AFR scheme, except that before each packet in a frame there is a subphysical-header, which is of a 12  $\mu$ s duration with an IEEE 802.11a PHY. The Aggregation scheme in [23] uses a special *delimiter* before each packet in a frame. As shown in

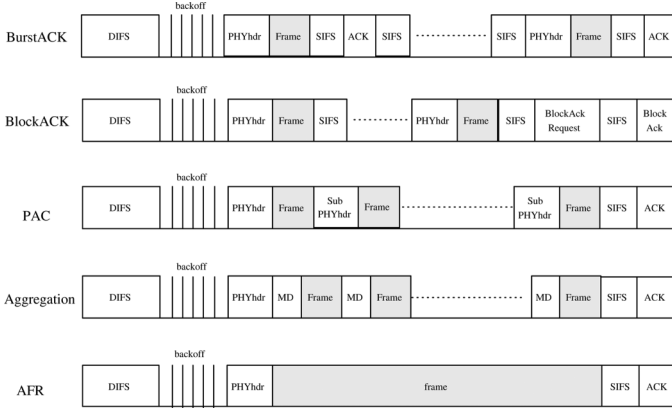


Fig. 7. Five schemes compared in this paper. 1) Burst ACK. 2) Block ACK. 3) Packet Concatenation from [22]. 4) Aggregation from [23]. 5) AFR.

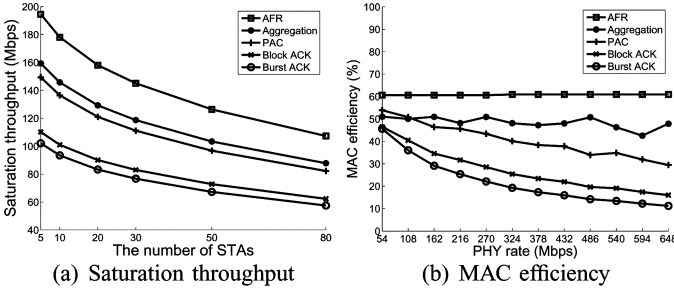


Fig. 8. AFR versus similar schemes.  $BER = 10^{-5}$  in both figures. In (a), the PHY data rate is 432 Mbps and the basic rate is 54 Mbps. In (b), frames are so selected to maintain a constant AFR efficiency. The other parameters are listed in Fig. 1(c) and Table III.

[29], delimitation techniques need support from the PHY layer. In particular, zeros should be inserted to ensure the particularity of the *delimiter*. The number of zeros inserted depends on the sizes of the *delimiter* and the packet. For an 8-b *delimiter* as in [23],  $L_p/(2^{\zeta+1} - 2)$  zeros are required, where  $L_p$  is the packet size and  $\zeta = 5$  [29].

Note that apart from AFR, none of these schemes satisfy all of the scalability conditions derived in Section III. Specifically, we have the following.

- **Burst ACK and Block ACK:** A PHY header is transmitted before each packet. The PHY header duration has a minimum value as discussed previously. Hence, the per-packet overhead does not decrease with increasing PHY rate.
- **PAC:** A subphysical header is transmitted before each packet, and similar comments apply.
- **Aggregation:** Fragmentation is not addressed in this scheme.

Results are shown in Fig. 8. It can be seen that the schemes employing aggregation (the second category) consistently outperform the Burst and Block ACK schemes. It can also be seen that the PAC scheme has the lowest throughput among schemes in the second category. This is due to the long duration of the subphysical header. AFR achieves the highest throughput regardless of the number of stations.

We further compare AFR with these alternative schemes using MAC efficiency in Fig. 8(b). Here, all schemes use the same frame sizes, which are selected to ensure a constant MAC

efficiency for AFR. We can see that AFR consistently achieves the highest efficiency.

### G. Delay Analysis

Our model can be extended to estimate the MAC layer delay, i.e., the mean time between a packet reaching the head of the MAC interface queue and being successfully transmitted. Let  $S^{frame}$  be the system throughput in frames per second rather than bits per second. That is, the MAC layer can transport  $S^{frame}$  frames in one second. Thus, the delay to successfully transmit one frame is  $1/S^{frame}$ , where

$$S^{frame} = \frac{E[\text{number of frames}]}{E[T]}. \quad (21)$$

In the AFR scheme, a packet is fragmented and may be only partially transmitted in one transmission. Thus, we need to know the mean delay before all fragments of a packet are successfully transmitted. Each fragment will be successfully transmitted in  $\leq r'$  successful frame transmissions with probability

$$(1 - p_e^{frag}) + (p_e^{frag})(1 - p_e^{frag}) + \dots + (p_e^{frag})^{r'-1}(1 - p_e^{frag}) = 1 - (p_e^{frag})^{r'}. \quad (22)$$

Suppose that a packet arrives and is divided into  $m'$  fragments. The probability of successfully transmitting  $m'$  fragments in  $\leq r'$  attempts is  $(1 - (p_e^{frag})^{r'})^{m'}$ . Further, assuming that errors are independent, the probability of transmitting a packet in exactly  $r'$  attempts is  $(1 - (p_e^{frag})^{r'})^{m'} - (1 - (p_e^{frag})^{r'-1})^{m'}$ . So the expected number of retransmission attempts can be written as

$$r = \sum_{r'=1}^{\infty} r' \left[ (1 - (p_e^{frag})^{r'})^{m'} - (1 - (p_e^{frag})^{r'-1})^{m'} \right]. \quad (23)$$

Here, the sum may be truncated to account for the finite number of retransmission attempts. Therefore we have that the per packet MAC delay  $D_{AFR}^{mac}$  is

$$D_{AFR}^{mac} = r \cdot \frac{P_I T_I + P_3 T_3 + P_C T_C}{P_3}. \quad (24)$$

For a fixed PHY rate, we expect the MAC delay to increase with the frame size due to the larger transmission time  $T_f$  for a frame. However, this is not the case when we choose the frame size to be a function of the PHY rate. In particular, by scaling the frame size in proportion to the PHY rate, not only do we maintain MAC efficiency, but we also maintain an approximately constant frame transmission time, in which case the MAC delay is invariant with PHY rate. This is illustrated in Fig. 9(a), which plots the MAC delay with increasing PHY rate. The corresponding frame sizes as a function of the PHY rates are shown in Fig. 9(b). Note that while the MAC efficiency and the MAC delay are constant, the actual throughput increases from  $54 \times 60\% = 32$  Mbps to  $648 \times 60\% = 388.8$  Mbps.

As noted previously, the level of MAC efficiency depends on the scaling factor  $b$  relating frame sizes to PHY rates. As we increase  $b$ , the efficiency rises. However, owing to the associated increase in frame transmission time, the MAC delay will also increase with  $b$ . A design decision therefore has to be made as to the desired tradeoff between MAC efficiency and delay.

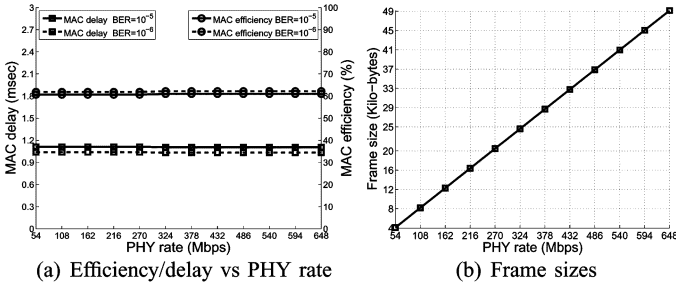


Fig. 9. Delay performance: In (a), we vary the frame sizes while increasing the PHY rates so that the MAC efficiency and MAC layer delay maintain roughly constant, and the corresponding frame sizes are shown in (b). The other parameters are listed in Fig. 1(c) and Table III.

## VI. SIMULATIONS

As a complement to the theoretical analysis in Section V, we have implemented the AFR scheme in the network simulator NS-2 [10], [11]. The network topology that we used is a peer-to-peer one where STA  $i$  sends packets to STA  $i+1$ . We report here the simulation results for three types of traffic (TCP, HDTV, and VoIP), all of which follow the requirements of the 802.11n usage model [8]. See our technical report for other details about the simulation [27].

### A. Metrics

We use the following metrics. Let  $c$  denote the number of packets (packet size is  $L_p$  B) successfully received by all of the STAs, and  $t$  denote the simulation duration. Let  $t_i^s$  be the time when the  $i$ th packet is put in the interface queue (IFQ) between MAC and its upper layer at the sender. Let  $t_i^e$  denote the time when the  $i$ th packet is transferred to its upper layer by the receiver.

- Throughput ( $= c * L_p * 8 / t$  Mbps): Throughput represents the maximum rate at which the MAC layer can forward packets from senders to receivers. Since in a WLAN, all the STAs share a common medium, this throughput is achieved by the whole system rather than by a single STA.
- Peak delay ( $= \max\{d_1^{\max}, d_2^{\max}, \dots, d_n^{\max}\}$ , where  $d_i^{\max}$  denotes the maximum delay among all the packets successfully received by STA  $i$ ): Peak delay is the maximum delay experienced by a successfully transmitted packet. This metric is used for HDTV.
- Percentage delay: The metric we use for VoIP is the percentage delay at the application level. It is defined as the percentage of packets whose delay is greater than a delay upper limit (e.g., at the application layer, the system should have less than 1% of packets whose delays are greater than 30 ms). At the MAC layer, we use a similar threshold, i.e., less than 1% of packets may have delay greater than 15 ms.

### B. TCP Traffic

TCP currently carries the great majority [50] of network traffic and it is therefore important to investigate the support of the AFR scheme for TCP traffic. Important features of TCP include the fact that traffic is i) elastic, and so achieved throughput is related to network capacity, and ii) two-way, and while TCP data packets are typically large, TCP ACKs are small packets, so it may be difficult to aggregate enough of them to form a large frame.

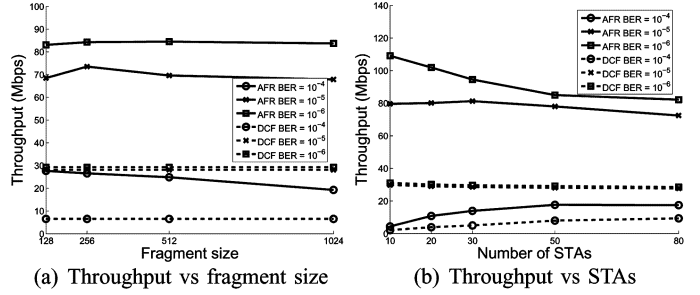


Fig. 10. Simulation results for TCP traffic. The parameters are listed in Fig. 1(c) and Table V.

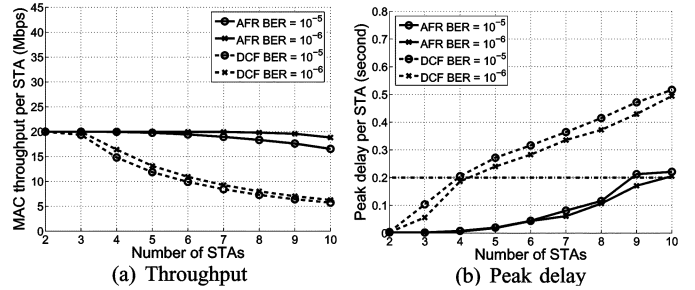


Fig. 11. Simulation results for HDTV traffic. The parameters are listed in Fig. 1(c) and Table V.

First, we evaluate AFR performance in a heavily-loaded WLAN with 50 STAs. Each STA performs a large FTP download. The data packet length is 984 B, which yields an IP packet size of 1024 B when TCP and IP headers are added, TCP SACK functionality is used as this is prevalent in real networks. From Fig. 10(a), we can see that AFR achieves considerable throughput gains (by a factor of between 2 and 3 depending on channel conditions) over DCF. As discussed previously, AFR performance is relatively insensitive to the choice of fragment size in the range 128–256 B, although as might be expected the choice of fragment size becomes more important at higher BERs.

Second, we evaluate AFR performance as the number of STAs is varied from 10 to 80. Fig. 10(b) shows both the AFR and DCF throughput. AFR achieves between 2.5 and 3 times the throughput of DCF over this range of network conditions.

### C. HDTV

According to the requirement of the IEEE 802.11n proposal [8], HDTV should be supported in future WLANs. HDTV has a constant packet size of 1500 B, a sending rate of 19.2–24 Mbps, and a 200-ms peak delay requirement.

We investigate AFR HDTV performance with a 432-Mbps PHY data rate. Fig. 11 shows the throughput and delay performance of the AFR and DCF schemes as the number of STAs (and so HDTV flows) is varied. The peak delay constraint of 200 ms is marked on Fig. 11(b). It can be seen that DCF can support only two simultaneous HDTV streams before the delay requirement is violated and the per-flow throughput rapidly falls below the offered load. In contrast, AFR can support up to nine and 10 streams for  $\text{BER} = 10^{-5}$  and  $\text{BER} = 10^{-6}$ , respectively. That is, the HDTV capacity is increased by a factor of 5 over DCF.

TABLE V  
PARAMETERS USED IN THE NS-2 SIMULATIONS

	Fig. 10	Fig. 11	Table VI
Number of STAs ( $n$ )	(a)50 (b)varied	varied	varied
Application rate (Mbps)	N/A	20	0.096
Data rate (Mbps) ( $R$ )	432	432	54
Basic rate (Mbps)	54	54	6
AFR sending queue (packets) <sup>a</sup>	10	10	10
AFR IFQ (packets) <sup>b</sup>	10	10	10
DCF IFQ (packets) <sup>c</sup>	20	20	20
Packet (bytes)	1024	1500	120
DCF frame (bytes)	1024	1500	120
AFR frame (bytes)	8192	9000	1200
AFR fragment (bytes)	(a)varied (b)512	750	120

TABLE VI  
SIMULATION RESULTS FOR VOIP TRAFFIC. THE FIRST ROW REPRESENTS THE NUMBER OF STAS. THE OTHER ROWS REPRESENT THE PERCENTAGE OF PACKETS WITH DELAY MORE THAN 15 ms WITH THE BOLD FIGURES SHOWING THE PERCENTAGE GREATER THAN 1%. THE PARAMETERS ARE LISTED IN FIG. 1(c) AND TABLE V

	10	30	50	80	90
AFR ( $BER = 10^{-4}$ )	0.0%	0.0%	0.0%	<b>4.4%</b>	<b>15.4%</b>
AFR ( $BER = 10^{-5}$ )	0.0%	0.0%	0.0%	<b>1.1%</b>	<b>9.4%</b>
AFR ( $BER = 10^{-6}$ )	0.0%	0.0%	0.0%	0.9%	<b>3.9%</b>
DCF ( $BER = 10^{-4}$ )	0.0%	0.0%	0.0%	<b>24.9%</b>	<b>85.7%</b>
DCF ( $BER = 10^{-5}$ )	0.0%	0.0%	0.0%	<b>10.1%</b>	<b>75.2%</b>
DCF ( $BER = 10^{-6}$ )	0.0%	0.0%	0.0%	<b>9.2%</b>	<b>34.8%</b>

#### D. VoIP

The third application that we consider is VoIP, which is basically an on/off UDP stream with a peak rate (96 Kbps) and a small packet size (120 B), according to the IEEE 802.11n requirements [8]. VoIP is a challenging application for aggregation schemes because of its on/off nature and small packet sizes. Thus, there may not be enough packets for AFR to aggregate, and the DCF and AFR schemes might be expected to achieve more or less the same performance.

We consider a WLAN with pure VoIP traffic. We use Brady's model [44] of VoIP traffic in which the mean ON and OFF periods are 1500 ms. Our performance requirement is to have less than 1% of packets with delays larger than 15 ms. Table VI shows the percentage of packets with delay exceeding 15 ms for a range of network conditions and numbers of voice calls. It can be seen that AFR's delay percentages are substantially less than the DCF's under all conditions, demonstrating the effectiveness of the AFR scheme, even for traffic with very small packet sizes.

#### VII. SCOPE OF THE PAPER

In this paper, we restrict consideration to independent and identically distributed (i.i.d.) channel noise. Although we recognize that such a memoryless model is unable to capture fading characteristics in wireless channels, we comment that the PHY characteristics of IEEE 802.11n are still unknown at this time, making the selection of a more accurate channel model problematic. We note that, provided the channel coherence time is long enough to support large frame transmissions, it is relatively straightforward to modify our analysis to encompass more complex channels. Moreover, it can be argued that i.i.d. noise is, in fact, a worst case for aggregation schemes since the bit errors tend to cluster together into bursts in fading environments [17]

(see also the measurement of the bit error distribution from an IEEE 802.11a test-bed [31]). An uneven error distribution typically benefits aggregation schemes since fewer retransmissions are required compared to i.i.d. noise with the same mean BER [9]. For instance, if there are 10 corrupted bits in one frame that contains 10 fragments, and each fragment has exactly one corrupted bit, then all the fragments have to be retransmitted. If all 10 corrupted bits occur in burst and gather into, say, five fragments, it is obvious that fewer retransmissions are needed.

In this paper, we focus on the fundamental issues affecting the performance of aggregation schemes in 802.11 WLANs. Thus, several other techniques for further optimizing CSMA/CA performance are not addressed here. These include optimization of the CSMA/CA contention window, which has been the subject of much attention in the literature (see [16], [20], [33], [49], and references therein for further details). Two-way aggregation is also possible, in which large frames are piggybacked in the ACK frames [4], [25], [46].

#### VIII. CONCLUSION

To achieve high efficiency for next-generation very high-speed WLANs, we develop a novel scheme called AFR, in which multiple packets are aggregated into and transmitted in a single large frame. Only the corrupted fragments are retransmitted instead of retransmitting the whole frame in case of errors. Transmission delays are minimized by using a zero-waiting mechanism where frames are transmitted immediately once the MAC wins a transmission opportunity. Analytical and simulation analysis models are developed to evaluate the throughput and delay performance of AFR over a noisy channel and to compare AFR with similar schemes in the literature. The analysis presented here is general enough to be extended to the proposed scheme in the upcoming 802.11n standard. Trends indicated in this paper should extend to any well-designed aggregation schemes.

#### ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their valuable comments.

#### REFERENCES

- [1] *Part 11: Wireless LAN MAC and Physical Layer Specifications, Reference Number ISO/IEC 8802-11:1999(E)*, IEEE Std 802.11, 1999.
- [2] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, IEEE Std. 802.11a, 1999.
- [3] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements*, IEEE 802.11e/D8.0, 2004.
- [4] S. A. Muftaba *et al.*, "TGn sync proposal technical specification," 2005 [Online]. Available: [www.tgnsync.org](http://www.tgnsync.org)
- [5] Q. Ni, T. Li, T. Turletti, and Y. Xiao, "AFR partial MAC proposal for IEEE 802.11n," 2004, IEEE 802.11-04-0950-00-000n.
- [6] J. Ketchum *et al.*, "System description and operating principles for high throughput enhancements to 802.11," 2004, IEEE 802.11-04/0870r0.
- [7] M. Singh *et al.*, "System description and operating principles for high throughput enhancements to 802.11," 2004, IEEE 802.11-04-0886-00-000n.
- [8] A. P. Stephens *et al.*, "IEEE.11 wireless LANs: Usage models," 2004, IEEE 802.11-03/802r23.
- [9] P. Brenner, "Magis Networks White Paper," 2004 [Online]. Available: [www.magisnetworks.com](http://www.magisnetworks.com)
- [10] NS [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [11] T. Li, "AFR implementation," Aug. 2006 [Online]. Available: [http://www.hamilton.ie/tianji\\_li/afr.html](http://www.hamilton.ie/tianji_li/afr.html)

- [12] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 607–614, Mar. 2000.
- [13] G. R. Cantieni, Q. N. C. Barakat, and T. Turletti, "Performance analysis under finite load and improvements for multirate 802.11," *Elsevier Comp. Commun. J.*, vol. 28, no. 10, pp. 1095–1109, Jun. 2005.
- [14] N. Celandroni, "Comparison of FEC types with regard to the efficiency of TCP connections over AWGN satellite channels," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1735–1745, Jul. 2006.
- [15] N. Celandroni, F. Davoli, E. Ferro, and A. Gotta, "Long-lived TCP connections via satellite: Cross-layer bandwidth allocation, pricing, and adaptive control," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 1019–1030, Oct. 2006.
- [16] J. Choi, J. Yoo, S. Choi, and C. Kim, "EBA: An enhancement of the IEEE 802.11 DCF via distributed reservation," *IEEE Trans. Mobile Comput.*, vol. 4, no. 4, pp. 378–390, Jul. 2005.
- [17] R. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
- [18] S. Ghez, S. Verdu, and S. C. Schwartz, "Stability properties of slotted ALOHA with multipacket reception capability," *IEEE Trans. Autom. Control*, vol. 33, no. 7, pp. 640–649, Jul. 1988.
- [19] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM*, 2003, pp. 836–843.
- [20] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," in *Proc. ACM SIGCOMM*, 2005, pp. 121–132.
- [21] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experiment Design, Measurement, Simulation and Modeling*. New York: Wiley, 1991.
- [22] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia, "Exploiting medium access diversity in rate adaptive wireless LANs," in *Proc. ACM MOBICOM*, 2004, pp. 345–359.
- [23] S. Kim, Y. Kim, S. Choi, K. Jang, and J. Chang, "A high-throughput MAC strategy for next-generation WLANs," in *Proc. IEEE WOWMOM*, 2005, pp. 220–230.
- [24] P. Lettieri and M. B. Srivastava, "Adaptive frame length control for improving wireless link throughput, range, and energy efficiency," in *Proc. IEEE INFOCOM*, 1998, pp. 564–571.
- [25] C. Liu and A. Stephens, "An analytic model for infrastructure WLAN capacity with bidirectional frame aggregation," in *Proc. IEEE WCNC*, 2005, pp. 113–119.
- [26] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "A new MAC scheme for very high-speed WLANs," in *Proc. IEEE WOWMOM*, 2006, pp. 171–180.
- [27] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "Aggregation with fragment retransmission for very high-speed WLANs," Hamilton Institute, NUIM, Ireland, 2006.
- [28] T. Li, Q. Ni, T. Turletti, and Y. Xiao, "Performance analysis of the IEEE 802.11e block ACK scheme in a noisy channel," in *Proc. IEEE Broadnets*, 2005, pp. 511–517.
- [29] J. S. Ma, "On the impact of HDLC zero insertion and deletion on link utilization and reliability," *IEEE Trans. Commun.*, vol. COM-30, pp. 375–381, Feb. 1982.
- [30] D. Malone, K. Duffy, and D. J. Leith, "Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 159–172, Feb. 2007.
- [31] A. Miu, H. Balakrishnan, and C. E. Koksal, "Improving loss resilience with multi-radio diversity in wireless networks," in *Proc. ACM Mobicom*, 2005, pp. 16–30.
- [32] Q. Ni, T. Li, T. Turletti, and Y. Xiao, "Saturation throughput analysis of error-prone 802.11 wireless networks," *Wireless Commun. Mobile Comput.*, vol. 5, no. 8, pp. 945–956, Dec. 2005.
- [33] Q. Ni, I. Aad, C. Barakat, and T. Turletti, "Modelling and analysis of slow CW decrease for IEEE 802.11 WLAN," in *Proc. PIMRC*, 2003, pp. 1717–1721.
- [34] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, Jun. 1995.
- [35] Q. Pang, V. Leung, and S. C. Liew, "A rate adaptation algorithm for IEEE 802.11 WLANs based on MAC-layer loss differentiation," in *Proc. IEEE Broadnets*, 2005, pp. 659–667.
- [36] F. Peng, J. Zhang, and W. E. Ryan, "Adaptive modulation and coding for IEEE 802.11n," in *Proc. IEEE WCNC*, Mar. 2007, pp. 656–661.
- [37] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proc. ACM MOBICOM*, 2002, pp. 24–35.
- [38] G. Tan and J. Gutttag, "Time-based fairness improves performance in multi-rate wireless LANs," in *Proc. USENIX*, 2004, pp. 269–282.
- [39] G. Tan and J. Gutttag, "The 802.11 MAC protocol leads to inefficient equilibria," in *Proc. IEEE INFOCOM*, 2005, pp. 1–11.
- [40] O. Tickoo and B. Sikdar, "On the impact of IEEE 802.11 MAC on traffic characteristics," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 189–203, Feb. 2003.
- [41] L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: From signal processing to optimal medium access control," *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 108–112, Nov. 2001.
- [42] J. Tourrilhes, "Packet frame grouping: Improving IP multimedia performance over CSMA/CA," in *Proc. ICUPC*, 1998, pp. 1345–1349.
- [43] V. Vitsas, P. Chatzimisios, A. C. Boucouvalas, P. Raptis, K. Paparrizos, and D. Kleptouris, "Enhancing performance of the IEEE 802.11 distributed coordination function via packet bursting," in *Proc. GLOBECOM*, 2004, pp. 245–252.
- [44] W. Wang, S. Liew, and V. O. K. Li, "Solutions to performance problems in VoIP over a 802.11 wireless LAN," *IEEE Trans. Veh. Technol.*, vol. 54, no. 1, pp. 366–384, Jan. 2005.
- [45] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement," in *Proc. IEEE INFOCOM*, 2002, pp. 599–607.
- [46] Y. Xiao, "IEEE 802.11 performance enhancement via concatenation and piggyback mechanisms," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2182–2192, Sep. 2005.
- [47] Y. Xiao, "IEEE 802.11n: Enhancements for higher throughput in wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 82–91, Dec. 2005.
- [48] Y. Xiao and J. Rosdahl, "Performance analysis and enhancement for the current and future IEEE 802.11 MAC protocols," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 7, no. 2, pp. 6–19, Apr. 2003.
- [49] X. Yang and N. Vaidya, "A wireless MAC protocol using implicit pipelining," *IEEE Trans. Mobile Comput.*, vol. 5, no. 3, pp. 258–273, Mar. 2006.
- [50] Z. Zhao, S. Darbha, and A. L. N. Reddy, "A method for estimating the proportion of nonresponsive traffic at a router," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 708–718, Aug. 2004.



**Tianji Li** (M'08) received the M.S. degree in networking and distributed computation in 2004 from Ecole Doctorale STIC, Université de Nice–Sophia Antipolis, France, and the Ph.D. degree in 2008 from the Hamilton Institute, National University of Ireland at Maynooth, where he is currently a Research Fellow.

He is interested in improving performance for computer and telecommunication networks.



**Qiang Ni** (SM'08) received the Ph.D. degree in 1999 from HuaZhong University of Science and Technology (HUST), China.

He subsequently spent two years as a Post-Doctoral Fellow at the Wireless Communication Research Laboratory, HUST. He is currently a Lecturer in the School of Engineering & Design, Brunel University, Uxbridge, U.K. Prior to that, he was a Senior Research Scientist at Hamilton Institute, National University of Ireland at Maynooth. He worked with INRIA France as a Researcher from 2001 to 2004. Since 2002, he has been active as an IEEE 802.11 Wireless Standard Working Group voting member and a contributor for the IEEE wireless standards. His research interests are wireless networking and mobile communications. He has published over 40 refereed papers in the above fields.



**David Malone** received the B.A.(mod), M.Sc., and Ph.D. degrees in mathematics from Trinity College Dublin, Ireland in 1996, 1997, and 2000, respectively.

During his time as a postgraduate, he became a Member of the FreeBSD development team. He is a Research Fellow at Hamilton Institute, National University of Ireland at Maynooth, working on wireless networking. His interests include wavelets, mathematics of networks, IPv6, and systems administration.

Dr. Malone is a co-author of O'Reilly's "IPv6 Network Administration."



**Douglas Leith** (M'02) graduated from the University of Glasgow, U.K., in 1986 and received the Ph.D. degree from the University of Glasgow in 1989.

In 2001, he moved to the National University of Ireland, Maynooth, to assume the position of SFI Principal Investigator and to establish the Hamilton Institute ([www.hamilton.ie](http://www.hamilton.ie)), where he is Director. His current research interests include the analysis and design of network congestion control and distributed resource allocation in wireless networks.



**Yang Xiao** (SM'04) received the B.S. and M.S. degrees from Jilin University, China, in 1989 and 1991, respectively, and the M.S. and Ph.D. degrees in computer science and engineering from Wright State University, Dayton, OH, in 2000 and 2001, respectively.

He worked in the industry 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, TN, in 2002. He is currently with the Department of Computer Science at The University

of Alabama, Tuscaloosa. He was a voting member of IEEE 802.11 Working

Group from 2001 to 2004. He is a member of the American Telemedicine Association. His research areas are security, telemedicine, sensor networks, and wireless networks. His research has been supported by the U.S. National Science Foundation (NSF) and U.S. Army Research, Fleet & Industrial Supply Center San Diego (FISCSD), and the University of Alabama's Research Grants Committee.

Dr. Xiao currently serves as Editor-in-Chief of *International Journal of Security and Networks (IJSN)*, *International Journal of Sensor Networks (IJSNet)*, and *International Journal of Telemedicine and Applications (IJTA)*. He serves as a referee/reviewer for many funding agencies, as well as a panelist for the U.S. National Science Foundation (NSF) and a member of Canada Foundation for Innovation (CFI)'s Telecommunications Expert Committee. He serves on TPC for more than 100 conferences such as INFOCOM, ICDCS, MOBIHOC, ICC, GLOBECOM, WCNC, etc. He serves as an Associate Editor for several journals, e.g., IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He has published more than 300 papers in major journals, refereed conference proceedings, and published book chapters related to these research areas.



**Thierry Turetletti** (SM'07) received the M.S. and the Ph.D. degrees in computer science from the University of Nice—Sophia Antipolis, France, in 1990 and 1995, respectively.

He is a Senior Researcher in the Planète Project-Team at INRIA Sophia Antipolis—Méditerranée, France. During his Ph.D. studies in the High Speed Networking (RODEO) project-team at INRIA, he designed one of the first videoconferencing tools for the Internet. From 1995 to 1996, he was a Post-Doctoral Fellow in the Telemedia, Net-

works and Systems (TNS) group at the MIT Laboratory for Computer Science, Cambridge. His current research interests include networking experimental platforms and simulators, mechanisms to support episodic connectivity, and adaptive cross-layer mechanisms for multimedia transmission over wireless networks.

Dr. Turetletti serves on the editorial boards of the journals *Wireless Communications*, *Mobile Computing (WCMC)*, *Wireless Networks (WINET)*, and *Advances in Multimedia*.