Aggregation with Fragment Retransmission for Very High-Speed WLANs

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

Abstract

The notorious overhead occurs at the medium access control (MAC) layer prevents the Wireless LANs (WLANs) from achieving desirable performance. This problem becomes even severe in the upcoming very high-speed WLANs, in which the physical layer (PHY) rate may exceed 216Mbps. To alleviate overhead, we propose a new MAC layer scheme — Aggregation with Fragment Retransmission (AFR). In this scheme, multiple packets rather than one are aggregated into and transmitted in a single large frame. If some are lost during the transmission, only the corrupted parts of the large frame will be retransmitted. This aggregation with partial retransmission technique allows for more efficient use of the wireless medium, since one frame transmission is expected to cause much less overhead than multiple packets transmissions do if properly designed. A theoretical analysis is used to evaluate AFR's performance. Extensive simulations are then carried out to validate the model and to study the behavior of AFR. Results confirm our expectation. Moreover, AFR is particularly effective for rich multimedia services with high data rates and large packet sizes, which is the key applications in the future WLANs.

Index Terms

Medium access control (MAC), Wireless LAN, very high-speed WLAN, IEEE 802.11, IEEE 802.11n.

I. Introduction

Following the wide-spread deployment, people are seeking to deliver rich multimedia applications such as high-definition television (HDTV, 20Mbps), DVD (9.8Mbps), etc. in the upcoming Wireless LANs (WLANs) [8]. To support these applications, the PHY rate in such networks is expected to exceed 216Mbps, some 802.11n proposals

The work of the first four authors was supported by Science Foundation Ireland Grant 03/IN3/I396.

Tianji Li is with the Hamilton Institute, National University of Ireland, Maynooth, Ireland (Email: Tianji.Li@nuim.ie).

Qiang Ni is with the Electronic & Computer Engineering, School of Engineering & Design, Brunel University, West London, UK (Email: Qiang.Ni@brunel.ac.uk). Part of his work was done while he was with the Hamilton Institute.

David Malone and Douglas Leith are with the Hamilton Institute, National University of Ireland, Maynooth, Ireland (Email: david.malone@nuim.ie, doug.leith@nuim.ie).

Yang Xiao is with the Computer Science Division, University of Memphis, USA (Email: yangxiao@ieee.org).

Thierry Turletti is with the PLANETE Project, INRIA Sophia Antipolis, France (E-mail: thierry.turletti@sophia.inria.fr).

claim to support up to 600Mbps ([4], [6], [7]). The MAC layer, however, greatly restrains the performance improvement due to its overhead (e.g., [9], [15], [34]). This paper addresses a new solution for this problem.

The overhead of it refers to backoff, distributed interframe space (DIFS), acknowledgment (ACK), short interframe space (SIFS) and PHY layer header. Due to its randomized characteristic, backoff leads to collisions and idle slots, which degrades throughput easily and attracts much attention [12], [15], [35]. However, even without the channel waste caused by the randomization backoff, the overhead is still huge, and becomes intolerable in very high-speed WLANs as shown in Fig. 1(a).

Therefore, we propose an Aggregation with Fragment Retransmission (AFR) scheme in this paper. The idea is to send, in only one frame, multiple of packets which are used to be transmitted in a Burst/Block as in Burst ACK ([31]) or in Block ACK ([3]). If errors occur during the transmission, we retransmit only the corrupted parts of this frame. We study in this paper the pros and cons of all the aspects of this aggregation plus partial retransmission. The main novelties lie in the following. We design a new frame format for supporting all the functionalities of AFR. This format allows for higher throughput with less overhead compared to previous proposals, and more importantly it supports one-to-many aggregation naturally. We propose a zero-waiting mechanism, which enables the aggregation technique a self-adaptive ability to the channel state. In addition, we study a fragmentation technique, in which packets longer than a threshold are divided into fragments before being aggregated. This technique is necessary for supporting jumbo frame transmissions.

A theoretical model is designed to evaluate AFR's performance and to compare AFR with its competing schemes. This model also gives a guideline for finding optimal frame and fragment sizes. We implemented the AFR scheme in the *NS-2* simulator. This implementation enable us to validate the theoretical model, and to simulate applications with diverse requirements according to 802.11n's requirements [8]. In particular, besides traditional CBR and TCP traffic, we test an application (20Mbps HDTV) with very large frames and a very high bandwidth request and an application (0.096Mbps VoIP) with very small frames and a very low rate. Results confirm AFR is a promising MAC technique for very high-speed WLANs.

The remainder of the paper is organized as follows. In Section III, we review the legacy schemes and introduce the motivation of this work. Section IV presents the AFR scheme in detail. A theoretical model is described in Section V to quantify the proposed scheme. Section VI describes the implementation details for AFR and the corresponding results. Finally Section VII concludes this paper. All the notation used in this paper is listed in Appendix.

II. Definitions

Before introducing our work, we define some concepts that will be used throughout this paper.

- We define a *packet* as what MAC receives from the upper layer, a *frame* as what MAC transfers to the PHY layer, and a *fragment* as a part of a *frame*.

- We define a *collision* as the event where at least two stations (STAs) start transmission at the same time. In this case the receivers can not decode any frames correctly.
- We define an *error* as the event satisfying the following two conditions at the same time. First, there is one and only one STA transmitting but the channel is so noisy that the receiver can not decode the whole frame successfully; Second, although PHY has detected errors, it still completes the reception and transfers the received frame to MAC. According to this definition, an *error* in this paper is a MAC layer concept[1].

III. MOTIVATION

A. DCF and Its Inefficiency

In the legacy DCF, a STA transmits a frame once it has observed an idle medium for a DIFS plus a backoff duration (the very first frame defers only for DIFS). If this frame is received without any errors, then the receiver sends back an ACK after a SIFS period. All the other STAs that also successfully receive this frame defer until the receiver completes sending the ACK. After the ACK, the receiver and all the other STAs defer a DIFS before backing off again for the next round of transmission.

Collisions and errors make the MAC layer protocol more complicated. In the case of collisions or errors, receivers and all the other STAs do not send back ACKs. The receivers defer their own transmission for an EIFS duration ($T_{EIFS} = T_{SIFS} + T_{PHYhdr} + T_{ACK} + T_{DIFS}$). The senders wait the potential ACKs for an ACK timeout duration, then defer a new backoff period before attempting the retransmission.

n	Number of STAs
$T_{\overline{CW}}$	Average backoff duration
T_{SIFS}	Time duration of SIFS
T_{DIFS}	Time duration of DIFS
T_{EIFS}	Time duration of EIFS
T_{data}	Time duration to transmit a frame in DCF
T_{ack}	Time duration to transmit an ACK frame
T_{PHYhdr}	Time duration for PHY header
δ	Propagation delay
σ	PHY layer time slot
L_f	MAC layer frame size in AFR (bytes)
L_p	Packet size in both DCF and AFR (bytes)
L_{data}	MAC layer frame size in DCF (bytes)

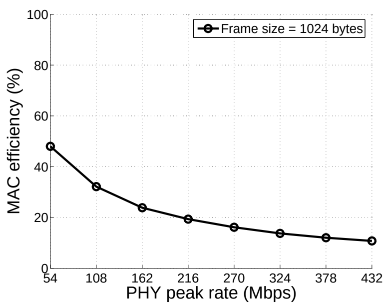
TABLE I
NOTATION USED IN THIS PAPER

The length of the backoff is the product of one slot duration σ and a random number uniformly chosen from the range of $[0, CW]$, where CW is the current contention window size. CW is doubled after each corrupted (collided or erroneous) transmission until the maximum contention window size CW_{max} is reached. After each successful

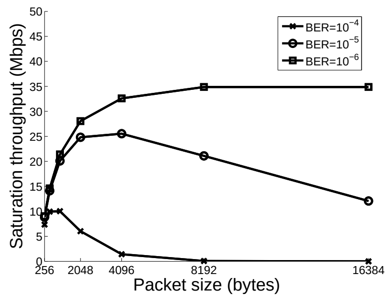
[1]In reality, errors may be also due to collisions if PHY is able to receive the transmission from multi-users simultaneously or in the presence of hidden terminals. Then an *error* can be defined as the event where although the receiver's PHY can complete reception, the frame received by MAC contains errors. A *collision* can be defined as the event where the receiver can detect that signals are coming but the reception is always interrupted.

transmission, CW is reset to the minimum contention window CW_{min}, thus $CW_{min} \leq CW \leq CW_{max}$. For full details of the DCF protocol see [1].

It has been demonstrated that DCF is not effective due to its overhead. We illustrate this in Fig. 1(a) by using the ideal case throughput [34]. In the ideal case, the channel is assumed to be perfect, i.e., neither errors nor collisions would occur. As can be seen from the results, the MAC efficiency decreases dramatically as the PHY data rate increases. In a 216Mbps WLAN, the MAC efficiency is only about 20%. When PHY data rate increases to 432Mbps, the efficiency decreases to around 10%. Apparently, the efforts to increase the system capacity are mostly wasted. Furthermore, even if the PHY data rate is infinitely high, the MAC throughput is still bounded [34].



(a) MAC inefficiency (1024 bytes)



(b) Throughput

T_{SIFS} (μs)	16
Idle slot duration (σ) (μs)	9
T_{DIFS} (μs)	34
T_{PHYhdr} (μs)	20
CW_{min}	16
Propagation delay (μs)	1
Symbol delay (μs)	4
Retry limit	4

(c) Parameters

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

B. Burst ACK and Block ACK

To reduce the overhead, Burst ACK (e.g., [31] [29] and [28]) and Block ACK (e.g., [3], [34]) have been proposed. The former solution reduces the times of backoff and DIFS by performing backoff once for a series of data and ACK frames (Fig. 7); the latter one goes one step further by backing off for a train of data frames plus only one ACK (Fig. 7), thus reducing the number of ACKs and SIFS. A comparison of these two schemes with the aggregation-based ones is shown in Fig. 8.

In these two schemes, the times of backoff, SIFS, DIFS and ACKs are reduced, but before each frame a PHY header is still needed. In the future WLANs, the PHY layer speed will exceed 216Mbps. If the duration of the PHY headers is still the same as the one used in IEEE 802.11a (i.e., $20\mu s$), then it is half of the transmission duration of a 1024-byte frame ($40\mu s$). According to the proposal 802.11n [4] for the future WLANs, it will take at least $44\mu s$ to transmit a PHY header, which is even longer than a frame transmission duration. Therefore, it is highly desired to curb the use of PHY headers.

C. Aggregation Schemes

One solution for decreasing the use of PHY headers is to aggregate and transmit the packets, which are transmitted in a Burst or a Block, into a single large frame. Traditionally, there is a dislike of large frames in wireless networks since small frames are usually more efficient [22]. We show this characteristic in Fig. 1(b). A large frame is very effective in a clear channel with low Bit Error Rate (BER, e.g, $BER \leq 10^{-6}$). But in a noisy channel (e.g., $BER \geq 10^{-5}$), the performance degrades dramatically. But, the precondition of this conclusion is that the traditional retransmission discards a whole frame even enough there is only one bit lost. Is it possible to retransmit only the erroneous part(s) of a frame? If possible, this partial retransmission would achieve better performance. This is the key motivation of this work.

Although this idea seems simple at first glance, it is actually a radical challenge for the PHY and MAC techniques. From the PHY viewpoint, the traditional small-packet rule does not hold any more. PHY has to transmit very large frames, and has to 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].

There are some ongoing activities in the 802.11n standard working group (e.g., [4], [5], [6], [7]), among which the TGn Sync [4] attracts much attention. Two features of the TGn Sync distinguish it from the others. First, a header compression technique is proposed. Second, a special *delimiter* is used to locate each fragments in a frame. Our previous proposal [5] was done in parallel with the TGn Sync. This paper extends our previous work substantially as we will summarize shortly.

In the academia, Ji et. al. [20] used an aggregation technique to solve the unfairness problem in WLANs. They suggest to remove the DIFS, SIFS and backoffs before a series of packets, and to transmit them together in a large PHY layer frame. However, a small PHY header ($12\mu s$) is used to identify each packet in each frame. In the upcoming very high-speed WLANs, however, the PHY header will be $44\mu s$ for one antenna, and $48\mu s$ for two antennas [4]. It will be the major source of overhead, thus should be removed if possible. In [16], a two-level (one at MAC, another at PHY) aggregation is proposed by using a similar *delimiter* to the one in the TGn Sync. The main disadvantage of the *delimiter* method is that the following start positions and lengths are unknown for the receiver if one preceding *delimiter* is corrupted. We will show this in Fig. 8.

D. Open Questions

Although aggregation is not a new idea, questions about it are still open:

- Firstly, where do large frames come from? The frames we want are typically larger than a packet. If the packets from the upper layer are big and come fast, then the aggregation is simple. If not, should some timing mechanisms be used to wait for enough packets? If so, how much time do we wait? Will this waiting causes delay problems?
- Secondly, suppose there are enough packets to aggregate into a large frame, how do we arrange the aggregated frame? What is a suitable frame format? The requirements for this format are at least twofold. First, the overhead caused should be as low as possible. It will make no sense if the overhead exceeds what caused by

PHY headers. Second, it is of crucial importance that the receiver can recognize the transmitted information despite errors.

- Thirdly, how do we choose the proper frame and fragment size(s)? If some fragments are lost during some previous transmission, how many times should they be retried?

IV. The AFR Scheme

From this section, we describe in detail our design and analysis for the AFR scheme. In this section, a frame format which causes fewer overhead than the previous schemes is introduced and an example is given to clarify the usage of it; Zero-waiting for aggregation is then descried to eliminate the delay worries; New queue management and retransmission logic are then discussed.

A. Scheme Description

The basic idea of the AFR scheme is to aggregate packets from the upper layer into large frames. Packets that exceed the fragmentation threshold are segmented into fragments. Then the MAC layer transmits the large frames containing multiple fragments and retransmits only fragments with errors identified using their Frame Check Sequence (FCSs). An example of the AFR scheme is shown in Fig. 2. In particular, at the sender, every outgoing packet is segmented according to a fragmentation threshold which will be discussed in Section V-C. Before transmission, all the fragments are marked as ’undelivered’ and kept temporarily in a MAC layer sending-queue (Sq). The MAC layer constructs a frame in the following way: It searches the Sq from head to tail for fragments marked as ’undelivered’ and aggregates them into the sending frame until either no ’undelivered’ fragments available or the frame size is sufficiently large (The optimal frame size is discussed in Section V-B). Then, the MAC layer transmits this frame (Fig. 3) according to the normal CSMA/CA procedure described in Section III-A.

Upon receiving a frame successfully, the receiver first checks the FCS of each fragment, constructs an ACK frame accordingly, and then sends back the ACK frame (see Fig. 4) in which the lost fragments are indicated in a bitmap field. The receiver keeps all the received fragments in a receiving-queue (*Rq*). All the packets that have been received successfully are to be transferred to the upper layer and be removed.

On receiving the ACK frame, the sender’s MAC checks the ACK bitmap field and updates the Sq accordingly by marking correctly received fragments as ’delivered’. Then it removes the successfully received packets from the Sq. Next, as long as the Sq is not null, MAC will construct and send out another frame immediately without waiting for more packets even though they are not long enough for a large frame. Please refer to Section IV-B.2 for the reason.

In the case that collisions happen, the AFR scheme runs in the same way as in the DCF scheme.

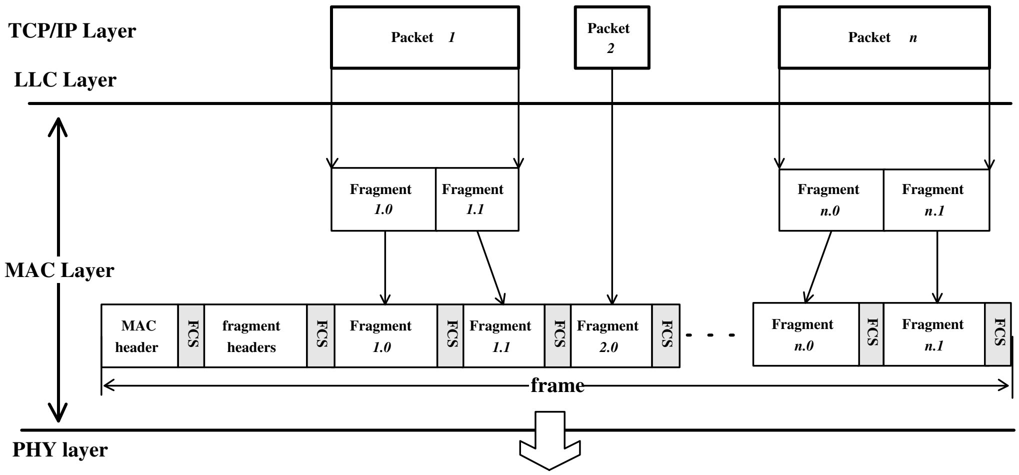


Fig. 2. The Aggregation with Fragment Retransmission (AFR) scheme.

There are two possibilities if transmission errors occur. First, the data frames may be corrupted while the ACK is successfully received. In contrast to DCF, AFR uses an ACK to notify the sender of which fragments have been lost. Therefore this is treated by AFR like a successful transmission. Second, the ACK frames may be lost. In this case AFR behaves in the same way as DCF, i.e., it behaves as if there has been a collision.

B. Design Issues

1) Frame Formats: Clearly, new data and ACK formats are the first concern for the AFR scheme. The difficulties of new formats are as follows: First, in an erroneous transmission, the receiver should be able to retrieve the correctly transmitted fragments. This is not easy because the sizes of the corrupted fragments may be unknown to the receiver. Second, tradeoff must be made between performance and overhead. Adding many fields in a frame will definitely support all the expected functionalities, but using reasonably few bits is important for system performance.

In our scheme, a MAC frame consists of a frame header and a frame body (Fig. 3(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. Second, the frame body consists of fragment-headers, fragment bodies and the corresponding FCSs (Fig. 3(b) and (c)).

The fragment-headers 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 bytes) and known to both the sender and the receiver. For the receiver, it knows where the *1-st* fragment header starts from 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*.

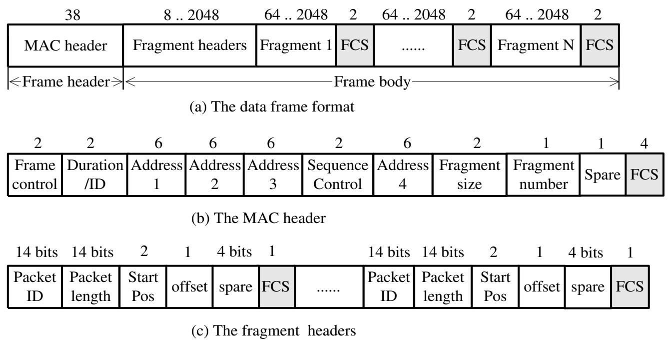


Fig. 3. Data format in the AFR scheme.

The new ACK format is simple, we add a 32-byte bitmap in the legacy ACK format. Each bit of the bitmap is used to indicate the correctness of a fragment (Fig. 4).

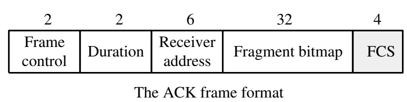


Fig. 4. ACK format in the AFR scheme.

To clarify the usage of the new formats, we give an example below. Suppose there are two packets (pkt_1 and pkt_2) with lengths of $l_1 = 1025$ bytes and $l_2 = 40$ bytes, the frame length is $l^f = 2048$ bytes and the fragment length is $l^{frag} = 512$ bytes[2]. Then AFR divides pkt_1 and pkt_2 into 3 and 1 fragments respectively and put them into the Sq. Then a frame with *fragment size* 512 bytes and *fragment number* 4 is constructed. The corresponding fragment headers are shown in Table. II.

	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

TABLE II

AN EXAMPLE USAGE OF THE AFR FRAME FORMATS.

After receiving the frame, the receiver operates in a way as shown in *Algorithm* 1 to recover the fragments.

2) Zero-Waiting: Large frame sizes are used in the AFR scheme, thus if the packets from the upper layer have small sizes, then a proper waiting mechanism should be designed. In this paper, we suggest an adaptive waiting

[2]To show that AFR can support arbitrary sizes of fragmentation, we do not use the optimal fragment sizes here.

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

```
if MAC header is correct then
  for i = 0 to fragment number - 1 do
    if Fragment i's header is correct then
      if packet length < fragment size then
        fragment i's length = pLEN;
      else if offset = ⌊pLEN/fragment size⌋ then
        fragment i's length = pLEN - offset * fragment size;
      else
        fragment i's length = fragment size;
      end if
      fragment start position = startPos in the fragment header.
      check the correctness of the fragment body using the FCS of it.
    end if
    record correctness (including fragment header and fragment body) of the fragments in a data structure called the ACK bitmap.
  end for
  construct ACK frame using the ACK bitmap and send it back.
  update the Rq according to the ACK bitmap.
  check the Rq and transfer all correctly received packets upwards, and remove them from the Rq.
else
  discard this frame and defer an EIFS before next transmission.
end if
```

mechanism, in which MAC never deliberately waits for packets to aggregate, a transmission is started whenever MAC wins the channel contention. The reasons for this zero-waiting are twofold:

First, aggregation is natural in heavily loaded networks. Because transmissions collide frequently in this case, and a frame would likely be retried several times before being received successfully. Every time a frame is retransmitted, MAC has a chance to search for more packets to fill this frame if it is not already long enough.

Second, in a lightly loaded networks, if the channel is noisy, aggregation will also happen automatically after a failure transmission attempt, remember we search for available packets to aggregate before each transmission; If the channel is error free, AFR degenerates to the legacy DCF scheme using zero-waiting. Since there is no much traffic to be transported, even though DCF is not desirable in terms of efficiency, it can still drain the system quickly.

In both cases, the zero-waiting enables the AFR to adapt to the channel conditions and traffic load automatically. Note that a similar method is used in a real test-bed [27].

3) Queue Management: We add two finite queues in the AFR scheme: the Sq and the Rq. Both of them are First In First Out (FIFO) queues. At the sender, the Sq can keep up to $limit_{Sq}$ packets, MAC never fetches new packets from its upper layer while the Sq is full. Thus, the actual frame size may be smaller than the desired one when there are many partially corrupted packets in the Sq. At the receiver, there is not an upper limit for the Rq. But, our scheme implicitly ensures that the Sq and the Rq always have the same size at the time when a data or an ACK is received. Of course, the contents of them are different while a transmission is in process. This character is very important for the retransmission mechanisum which is detailed in the Section IV-B.4.

4) Retransmission: In the DCF scheme, all the frames that have experienced more transmit attempts than the retransmission-threshold, $ReTXthreshold$, should be discarded and the contention window will be reset.

However, in the AFR scheme, one frame may contain multiple fragments from different packets. We maintain a retransmission counter for each frame, which is incremented after each transmission attempt. If $ReTXthreshold$ is exceeded, then a packet from the Sq will be discarded. But which one? It must be the head of the Sq ($head_{Sq}$) in our scheme. The reason is that we construct MAC frame by a strict order in which packets are received from the upper layer. According to the rule, the first fragment in a transmitting frame must be a part of the $head_{Sq}$. At the same time, it must be the one that has been retried the most times. Therefore, after re-trying a frame more than $ReTXthreshold$ times, the $head_{Sq}$ is removed from the Sq.

After removing a packet from the Sq, the sender needs to indicate this to the receiver. This still relies on the strict order of the Sq, Rq and the way we construct MAC frames. Each time when the receiver gets a frame f, if the first fragment in f does not belong to the first packet in the Rq ($head_{Rq}$), then the $head_{Sq}$ must have been removed from the Sq. Then, two steps should be performed by the receiver. First, *Algorithm* 1 is executed and the first fragment's ID pID_{coming} is recorded; Second, the receiver checks the Rq from head to tail, compares pID_{coming} to the packets' ID in its Rq, and removes all the packets whose ID is not equal to pID_{coming} until meets the packet with $pID = pID_{coming}$. Note that this solution is also suitable after the packet ID wraps to zero after reaching its maximum value.

C. Comments

1) Optimality: Solutions for optimal frame and fragment sizes are discussed in Section V-B and V-C. The fragmentation algorithm based on the optimal fragment sizes is then discussed.

2) Fairness: AFR strictly follows the basic principle of the CSMA/CA, therefore the same fairness characteristics holds as in the legacy DCF. Techniques to improve DCF's fairness are all suitable for AFR. Interested readers should refer to [20], [18], [29] or the TXOP mechanism in IEEE 802.11e [3].

3) Multi-destinations: Thus far, we focus only on the aggregation between one source-destination pair. The reason is that we can have a clear understanding of the pros and cons of the aggregation itself. However, our frame format can be easily extended to support multi-destinations. We can add a destination address field in each fragment header, and remove the destination address field from the MAC header.

Adding another field in the fragment header will result in more overhead, which is unavoidable. But compared to the solution in the literature [16], our scheme would have low overhead due to the following reasons. The authors of [16] propose to use a *physical delimiter*, which is transmitted at 6Mbps. First, this *delimiter* technique requires that extra zeros are added at PHY layer [25], see Section V-D for an example of this overhead. Second, transmitting the *delimiter* at 6Mbps leads to a constant $8\mu s$ overhead. In our scheme, however, both MAC and fragment headers are transmitted at the current data rate which may be more than 400 Mbps.

V. Theoretical Analysis

This section introduces our theoretical analysis for AFR, which is used to analyse AFR itself, to find the optimal frame and fragment sizes, and to compare AFR with its competing schemes.

Based on previous work [11], [33], [26] and [24], we have designed a model to analyse the saturation throughput of the AFR scheme. Here, we assume the readers are familiar with the Bianchi's model, and explain only the differences of our model to his.

We say a MAC is saturation if whenever the MAC layer needs a frame to transmit, it can always fill a long enough frame without waiting [11]. The saturation throughput S_{AFR} is defined as the expected payload size of the successfully transmitted frame in an expected slot duration.

$$S_{AFR} = \frac{E[L_{pld}]}{E[T]}. \tag{1}$$

We first compute the expected slot duration $E[T]$. Altogether, there are three kinds of duration in the AFR scheme. First, if none of the n STAs transmit any frames, they all wait for an idle duration T_I. Second, in both successful and error cases, the slot durations are the same which is the sum of a frame, a SIFS and an ACK duration. We use T_3 to denote this. Third, let T_C denote the duration for a collision. in which case the receiver waits for an EIFS before the next transmission. Therefore:

$$T_I = \sigma \tag{2}$$

$$T_3 = T_{PHYhdr} + T_f + T_{SIFS} + T_{PHYhdr} + T_{ack} + T_{DIFS} \tag{3}$$

$$T_C = T_{PHYhdr} + T_f + T_{EIFS} \tag{4}$$

Then, we derive the corresponding possibilities for these durations. Let τ and n denote a STA's transmission probability in a slot and the number of STAs in the system respectively.

Firstly, in an idle slot, a single STA does not attempt transmission with probability $(1 - \tau)$, so all the n STAs in the system keep silent with probability P_I:

$$P_I = (1 - \tau)^n \tag{5}$$

Secondly, let P_3 denote the probability of an successful or erroneous transmission. In this case, there is one and only one STA is in transmission, thus the probability is:

$$P_3 = \binom{n}{1} \cdot (\tau(1 - \tau)^{n-1}) \tag{6}$$

Thirdly, since these three events (idle, success_and_error, collision) are mutually exclusive [21], collision probability of the whole system can be defined as:

$$P_C = 1 - P_I - P_3 \tag{7}$$

So far we have known all the variables except τ in Equations 5, 6 and 7. Let p_f denote the probability of doubling contention window after a transmission. Then τ can be expressed as a function of p_f, and we can find another function of τ for p_f. Both of them are obtained from a Markov chain that is similar to the one in Bianchi's paper [11].

Let us consider the first formula for p_f and τ. Bianchi's paper assumes there are no errors in the channel, so $p_f = p_c = 1 - (1-\tau)^{n-1}$ where p_c is one STA's collision probability. 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 packet error rate. In the AFR scheme, the receiver sends back the ACK frame in both successful and erroneous cases, thus $p_f = p_c$.

Now, we introduce the second formula for p_f and τ. The transmission probability τ in a slot time should be the sum of all the probabilities of the contention window decreases to zero at all the backoff stages. I.e., $\tau = \sum_{i=0}^{b} b_{i,0}$ where b is the maximum backoff stage as defined by $CW_{max} = 2^b \cdot CW_{min}$, and $b_{i,0}$ is the probability of the contention window decreases to zero at the stage i. Bianchi's paper assumes that a frame can be retransmitted infinite times, which is inconsistent with the 802.11 specification [1]. Wu et. al. loose this assumption in their work [33]. We borrow Equations (8) and (9) from [33] to solve $b_{i,0}$.

With these two formulas, a closed form for p_f and τ is formed and both of them can be solved. Therefore, we find the last variable τ required in Equations 5, 6 and 7.

As a result, the saturation throughput S_{AFR} of the AFR scheme is:

$$S_{AFR} = \frac{P_3 \cdot E[L_f]}{P_I T_I + P_3 T_3 + P_C T_C} \tag{8}$$

Let i denote the number of erroneous fragments, and m denote the number of fragments in a frame. Assuming an independent and identical distribution (see Section VI-A for the explanations of using this distribution) of errors, the expected size $E[L_f]$ can be expressed as:

$$E[L_f] = \sum_{i=0}^{m} \binom{m}{i} \cdot (p_e^{frag})^i \cdot (1 - p_e^{frag})^{m-i} \cdot (L_f - i \cdot L_{frag}) \tag{9}$$

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

$$p_e^{frag} = 1 - (1 - p_b)^{L_{frag} + L_{FCS}} \tag{10}$$

where L_{frag} and L_f are the length of a fragment and the length of a frame respectively, and p_b is the BER[3].

This model is validated against *NS-2* simulations. Both simulation and model results are shown in Fig. 5(a). As we can see from the results, the analysis and simulations match very well.

A. Improvement over DCF

To compare AFR with DCF, a model for the latter is required. We use the DCF-MODEL that has been developed and validated in our previous work [26].

$$S_{DCF} = \frac{P_S \cdot L_{data}}{P_I T_I + P_S T_S + P_E T_E + P_C T_C}. \tag{11}$$

AFR achieves fundamental improvement comparing to the legacy DCF scheme. The results are shown in Fig. 5(b). In this figure, the x-axis represents the frame sizes in log scale, while the y-axis is the throughput. The PHY rate is 54Mbps, and the basic rate is 6Mbps. Moreover, both schemes are saturated in the sense that no matter what the frame sizes are, large frames are always available. In the AFR scheme, fragment sizes are chosen in a way that maximizes the throughput. As we can see from the results that DCF behaves poorly for large frames. But, AFR prevents the throughput from dropping as the frame sizes increase. The improvement are rather promising. For example, DCF achieves zero for 8192 bytes in a 10^{-4} channel while AFR achieves around 30Mbps throughput for a 54Mbps PHY layer. Then a natural question is what the best frame size is. We answer this in the following section.

B. Optimal frame size

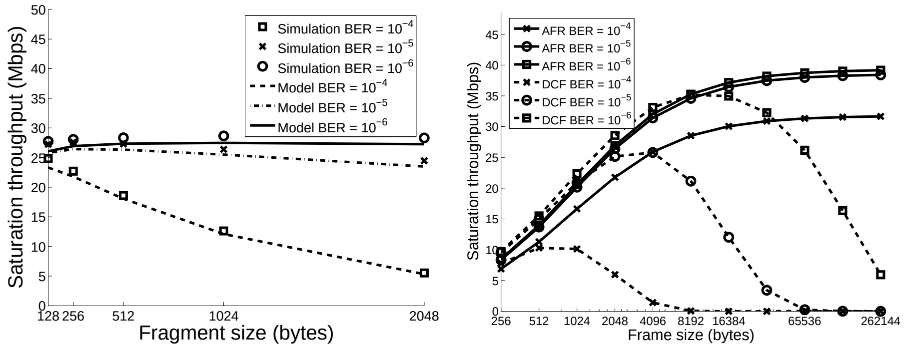
First, the frame sizes depend on the PHY's abilities and the traffic characteristics (especially sending rates and packet sizes). If the PHY layer can support arbitrary frame sizes, and applications can provide arbitrary amount of packets, then the optimal frame size will be constrained by the length of the sending queue and the delay requirements of the applications.

Second, ignoring the constraints just mentioned, the longer the frame size the better. That is, the MAC layer does not pose any constraints on performance. To show this, let us look at Fig. 5(b) again. As we can see, under all channel states, the throughput increases with frame sizes in the AFR scheme. This is because we amortise the duration of the PHY header across more fragments, while the per-fragment error probability remains constant.

Third, a shorter frame than the optimal one is preferred in practice since some of the constraints may not be true. Still using the proposed model, we show that a long-enough frame can also near optimal throughput. As summarised in Table 5(c) in which the tolerable throughput loss is 10% comparing to the optimum, a 32768-byte

[3]In reality, p_b can be measured by PHY. If p_b is difficult to obtained in a real device, the measurement of p_e can be implemented alternatively because it is simpler.

frame fails for 432Mbps and 648Mbps PHY rates, respectively. While a 65536-byte frame always maintains less than 10% differences comparing to the optimum. Therefore, 65536 bytes can be used in practice to approximate the optimal frame size, and this is also the longest size proposed in TGn's 802.11n proposal [4].



(a) Model Validation (b) Influence of the frame size

Rates	32768	65536
54/6	2.5%	1.1%
108/24	4.2%	1.8%
216/24	8.3%	3.6%
432/54	**15.6%**	6.7%
648/216	**22.9%**	9.8%

(c) Throughput differences

Fig. 5. (a) AFR: model vs. simulations. (b) The influence of frame size. PHY date rate is 54Mbps, ACK rate is 6Mbps, and the number of STA is 10. (c) 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 optimal throughput. Other parameters are listed in Fig. 1(c) and Table III.

	Fig. 5(a)	Fig. 5(b)	Fig. 6	Fig. 8	Fig. 9
Number of STAs (n)	10	10	10	10	10
Application rate (Mbps)	54	54	54	432	54
Data rate (Mbps) (R)	54	54	54	432	54
Basic rate (Mbps)	6	6	6	54	6
AFR Sq (packets)[a]	200	N/A	N/A	N/A	N/A
AFR IFQ (packets)[b]	200	N/A	N/A	N/A	N/A
Packet (bytes) (L_p)	2048	$= L_f$	$= L_{frag}$	2048	$= L_f$
Frame (bytes) (L_f)	2048	256, $\cdots$, 65536*4=262144	8192	8192	256, $\cdots$, 65536
AFR fragment (bytes)(L_{frag})	128, $\cdots$, 2048	256	32, $\cdots$, 8192	256	256

TABLE III

THE PARAMETERS USED IN THE THEORETICAL MODEL AND ITS VALIDATION.

[a] AFR Sq is the queue at MAC layer for temporarily store the packets from the AFR IFQ in AFR's simulations.
[b] AFR IFQ is the queue between MAC and its upper in AFR's simulations.

For large frame sizes, fragmentation plays a critical role for efficiency. The fragment sizes used in Fig. 5(b) are the optimal ones. The method of determining the optimal fragment size is explained in Section V-C.

C. Optimal fragment size

As the third application of the proposed model, we use it to analyse the optimal fragment size. In the AFR scheme, we propose to divide large frames into fragments to improve the efficiency. To make the best use of this technique, it is desirable to use an optimal fragment size. The optimum may have different definitions, a suitable one here is the fragment size which maximizes the throughput.

Let us look at the impact of fragment sizes as illustrated in Fig. 6. Here, we use a constant frame size (8192 bytes) and different fragment sizes. The x-axis represents the fragment sizes, the y-axis represents the absolute (i.e., always positive) differences between the throughput using the current fragment size and the throughput using the optimal fragment size. The PHY date rate is 54Mbps, the ACK rate is 6Mbps, and the number of STA is 10. From the results we can see that: First, the optimal fragment sizes are not constant in different channel conditions. Second, if we allow for 10% performance loss, then 128 and 256 can be used as near-optimal fragment sizes. Question is if these two values are also suitable for other conditions. To answer this, results under varied PHY layers are summarised in Table IV. Interestingly, these two optimal values are not sensitive to the PHY rates since they always lead to less than 10% throughput losses. Other results under different number of STAs, and different frame sizes are also obtained, but will not be plotted here due to their similarity to the results in Table IV. As a whole, 128 and 256 are the optimal fragment sizes.

Based on these two optimal fragment sizes. A simple fragmentation algorithm can be descried as: for a packet P with a size of L_p, find the m which satisfies that

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

where $m = 1, 2, ..., 256$. Then we divide P into m fragments, each of which has a size in the range of $(\frac{L_p}{m}, \frac{L_p}{m}+1, ..., \frac{L_p}{m}+(m-1))$. In this way, the sizes of all fragments will fall between 128 and 256 bytes, which are the best ones for the throughput.

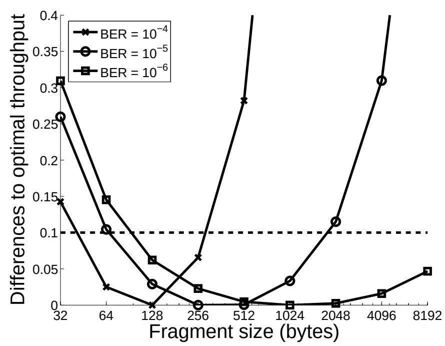


Fig. 6. The x-axis represents the fragment sizes, the y-axis represents the absolute (i.e., always positive) differences between the throughput using the current fragment sizes and the throughput using the optimal fragment sizes. Other parameters are listed in Fig. 1(c) and Table III.

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

TABLE IV

DIFFERENCES TO OPTIMAL THROUGHPUT IN DIFFERENT PHY LAYERS.

[a]The PHY rates are on the left of the slash, the basic rates are on the right. The unit of the rates is Mbps.
[b]The results are frames with 64-byte fragments, under BER 10^{-4}, 10^{-5}, 10^{-6} respectively.

D. Comparison with Similar Schemes

In this section, we use the theoretical model to compare AFR with four similar schemes: Burst ACK ([28] [29] [31]), Block ACK ([3] [34]), Packet Concatenation (PAC) [20] and *Aggregation* [16].

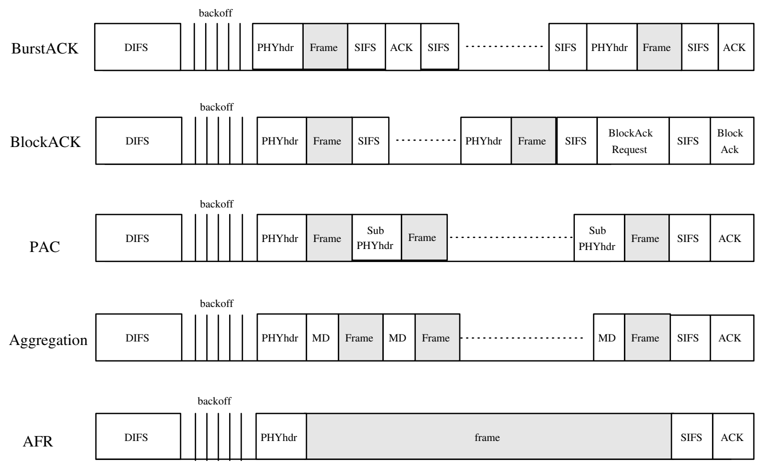


Fig. 7. Five schemes compared in this paper. 1) Burst ACK. 2) Block ACK. 3) Packet Concatenation from [20]. 4) Aggregation from [16]. 5) AFR.

The reason that we can use the same model for all these schemes is that this model is designed for CSMA/CA based MAC protocols. As long as a scheme is based on CSMA/CA, the two dimensional Markov Chain can be used, Equation 1 is also valid. What we need to change are time durations (Equations 2, 3, 4) and the corresponding probabilities (Equations 5, 6, 7).

The five 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 once obtain the channel; The schemes in the second category transmit **only one** frame, and they all use aggregation. Only in AFR, fragmentation plus aggregation is employed.

For Burst ACK and Block ACK. If collisions happen, then the whole Burst/Block is lost; Errors distribute according to the independent and identical distribution in the Burst/Block.

The PAC scheme is similar to our AFR scheme, except before each packet in a frame there is a sub-physical-header, which is transmitted in $12\mu s$ for IEEE 802.11a.

The *Aggregation* scheme in [16] uses a special *delimiter* before each packet in a frame. As shown in [25], 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 should be inserted depends on the sizes of the *delimiter* and the packet. For a 8-bit *delimiter* as in [16], $\frac{L}{2^{n+1}-2}$ zeros are needed, where L is the packet size, and $n = 5$ [25].

The results are shown in Fig. 8. First, the schemes based on aggregation (the second category) outperform those non-aggregation ones. Second, the PAC scheme has the least throughput in the second category due to its slow sub-physical-header. Third, AFR achieves the best since it combines the benefits of both aggregation and fragmentation.

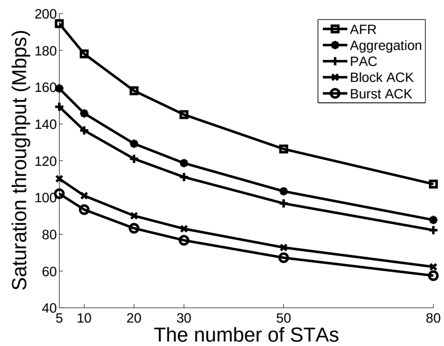


Fig. 8. Saturation throughput of the five schemes compared in this paper. The PHY data rate is 432 Mbps, basic rate is 54 Mbps. The other parameters are listed in Fig. 1(c) and Table III.

E. Delay Analysis

Following the analysis above, the saturation delay can also be derived. Delay is the time period during which an upper layer packet is successfully transmitted by MAC. Since the queue is never null under the saturated assumption, the delay here denotes only the MAC layer CSMA/CA delay which consists of idle slots, collisions, errors and transmission delays.

Let S^{frame} be the system throughput in frame-per-second rather than bit-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

$$S^{frame} = \frac{E[number\ of\ frames]}{E[T]} \tag{12}$$

Let D_{AFR} and D_{DCF} be the MAC delay of AFR and DCF respectively. For the DCF delay, as just explained,

$$D_{DCF} = \frac{P_I T_I + P_S T_S + P_C T_C + P_E T_E}{P_S} \tag{13}$$

In the AFR scheme, however, a packet is partially transmitted in one transmission. Thus, we need to know the transmission times in which all the fragments of this packet are transmitted. Each fragment will be successfully transmitted in $\leq a$ attempts with probability

$$(1 - p_e^{frag}) + (p_e^{frag})(1 - p_e^{frag}) + \ldots + (p_e^{frag})^{a-1}(1 - p_e^{frag}) = 1 - (p_e^{frag})^a. \tag{14}$$

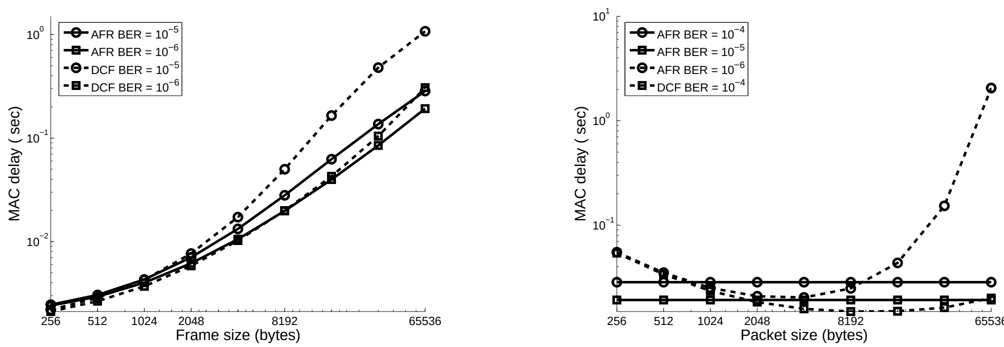
Suppose that a packet of length L_p arrives and is cut into M fragments. The probability of successfully transmitting M fragments in $\leq a$ attempts is $(1 - (p_e^{frag})^a)^M$. Further, assuming that errors are independent, the probability of transmitting a packet in exactly a attempts is $(1 - (p_e^{frag})^a)^M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

$$A = \sum_{a=1}^{\infty} a \left[(1 - (p_e^{frag})^a)^M - (1 - (p_e^{frag})^{a-1})^M \right]. \tag{15}$$

Here, the sum may be truncated to account for finite retransmission attempts. Therefore,

$$D_{AFR} = A \cdot \frac{P_I T_I + P_3 T_3 + P_C T_C}{P_3} \tag{16}$$

A delay comparison using Equations 13 and 16 is given in Fig. 9.



(a) Frame (b) Packet

Fig. 9. MAC delay: AFR vs DCF. The PHY data rate is 54 Mbps, basic rate is 6 Mbps. The other parameters are listed in Fig. 1(c) and Table III.

VI. SIMULATIONS

A. The simulation setup

In this section, we introduce our implementation of AFR in the network simulator *NS-2* [10].

In the *NS-2* version 2.27 [10], the PHY headers are transmitted at the PHY data rate. However, the IEEE 802.11a standard [2] and the IEEE 802.11n proposals [4], [7] specify that PHY header should be transmitted within a constant duration ($20\mu s$ in 802.11a) no matter what the PHY data rate is. We change the *NS-2* code accordingly.

Our network topology is a single-hop WLAN in which all the STAs are put on a line and the transmission power is high enough to cover all the other STAs, so that there are no hidden terminals in the network. Such a topology is a typical one proposed for future WLANs [8], the multi-hop extension of the AFR scheme will be our future work.

To guarantee fairness in the simulations, we use the Jain's fairness index I [19] which is a real value between 0 and 1. In particular, Jain's fairness index I is defined as:

$$I = \frac{(\sum_{i=1}^{n} S_i)^2}{n \cdot \sum_{i=1}^{n} S_i^2}, \tag{17}$$

where n stands for the number of STAs and S_i is the throughput of STA i. When every STA achieves exactly the same throughput, I is equal to 1. If only one STA happens to dominate the channel entirely, I approaches $1/n$. In our simulations, we run each test for a long enough duration to obtain a fairness index $I > 0.95$.

We summarize the assumptions of the simulations in the following:

- There are no hidden terminals.
- Channel model: We use the discrete-time, memory-less Gaussian channel as an example. In such a channel, the bit errors are assumed to occur independently and identically distributed over a frame [13]. Let L_f and p_b denote the frame size and the BER respectively, then the frame error rate p_e can be derived as:

$$p_e = 1 - (1 - p_b)^{L_f}. \tag{18}$$

  where p_b is assumed to be known by MAC. Although the memory-less Gaussian model is unable to capture the fading characteristics of the wireless channel, it is widely used to model wireless channels due to its simplicity[4].
- RTS/CTS: We do not use RTS/CTS in our simulations. Basically the RTS/CTS technique does not change the running logic of either AFR or DCF, i.e., what we are interested in is how the AFR scheme will improve the performance of the basic CSMA/CA scheme. Besides, both RTS and CTS frames need a PHY header, which causes large overhead in very high-speed WLANs. Thus, RTS/CTS are unlikely to be a good option in single-hop WLANs.

B. Metrics

In this section, we define the metrics that will be used in the simulations. Let c denote the number of packets (packet size is L_p bytes) successfully received by all the STAs and t denote the simulation duration. Let t_i^s denote the time at which 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 packet at which 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 without packet losses. Since in a WLAN, all the STAs share a common medium, this throughput is what achieved by the whole system rather than by a single STA.
- Average delay (= $(\sum_{i=1}^{m}(t_i^e - t_i^s))/m$): Average delay represents the mean duration between the time a packet arrives at the IFQ and the time it is transferred to the receiver's upper layer successfully.

[4]In a fading channel, the bit errors tend to cluster together into bursts [14]. In the gap between two consecutive bursts, error probability decreases to almost zero. In a burst, however, the errors occur with high probability. This characteristic can be described by a correlation factor, a value that ranges from 0 to 1. With a correlation factor close to 0, the channel becomes a Gaussian one. On the other hand, when the correlation factor approaches 1, all errors occur consecutively. The throughput in the fading channel is expected to be higher than in the Gaussian channel, because fewer retransmission are required [9].

- Peak delay (= $max\{d_1^{max}, d_2^{max}, \cdots, d_n^{max}\}$, where d_i^{max} denotes the maximum average delay among all the packets successfully received by STA i): Peak delay is the maximum delay experienced by a successfully transmitted packet in one simulation. This metric is used for HDTV.
- Percentage delay: A suitable metric for VoIP should be the percentage delay at the application level. It can be defined as the percentage of packets whose delay are greater than a delay upper limit (e.g, at the application layer, the system should tolerate less than 1% of packets whose delays are greater than 30 ms. This is the criterion proposed in IEEE 802.11n's requirement [8]). At the MAC layer, we use a similar percentage, i.e., there should be less than 1% of packets whose delays are greater than 15 ms in the system.

C. CBR traffic

Constant Bit Rate (CBR) traffic is a simple application which generates constant-size packets with a fixed rate. There are not any application level ACKs for lost packets or other control mechanisms for retransmission. Thus we use it as a basic test for evaluating the functionalities of the new MAC layer.

As a first example, we compare AFR and DCF while increasing the PHY data rates. As illustrated in Fig. 10(a), the DCF scheme's efficiency is always bounded by that of the ideal case as discussed in Section III-A. But the AFR scheme exceeds this limit easily and improves MAC efficiency to around 60% and 35% for 54Mbps and 432Mbps PHY rates, respectively. Compared to DCF, the improvement[5] of AFR ranges from 50% up to 200% (Fig.10(b)).

	Fig. 10	Fig. 11	Fig. 12	Fig. 13	Fig. 14	Fig. 15	Fig. **??**
Number of STAs (n)	10	varied	10	50	(a)50 (b)varied	varied	varied
Application rate (Mbps)	R / n	54	54	54	N/A	20	0.096
Data rate (Mbps) (R)	varied	432	432	432	432	432	432
Basic rate (Mbps)	R / 9	54	54	54	54	54	54
AFR Sq (packets)[a]	10	10	10	10	10	10	10
AFR IFQ (packets)[b]	10	10	10	10	10	10	10
DCF IFQ (packets)[c]	20	20	20	20	20	20	20
Packet (bytes)	1024	1024	(a)1024 (b)8192	512	1024	1500	120
DCF frame (bytes)	1024	1024	1024	512	1024	1500	120
AFR frame (bytes)	8192	8192	8192	varied	8192	9000	1200
AFR fragment (bytes)	512	256	varied	512	(a)varied (b)512	750	120

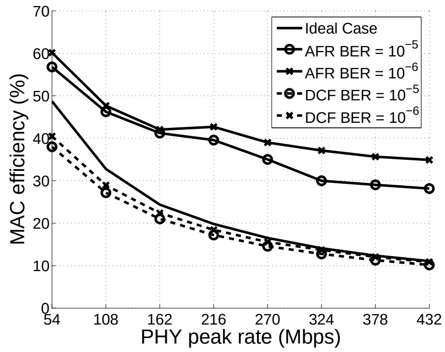
TABLE V

THE PARAMETERS USED IN THE *NS-2* SIMULATIONS.

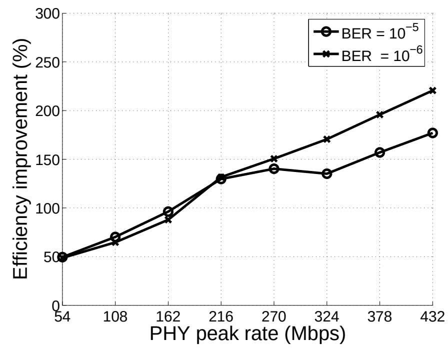
[a]AFR Sq is the queue at MAC layer for temporarily storing the packets from the AFR IFQ.
[b]AFR IFQ is the queue between MAC and its upper in AFR's simulations.
[c]DCF IFQ is the queue between MAC and its upper in DCF's simulations.

In this example, we use a constant PHY header duration ($20\mu s$) from the IEEE 802.11a for all the PHY data rates. This value will be increased for higher speed WLANs [4] since the decoding time of WLANs with higher speed PHY layer will be longer than that of the 802.11a. Given a longer PHY header, the efficiency improvement will be greater than what is shown in this example.

[5]Let S_{AFR} and S_{DCF} be the throughput of AFR and DCF, then the improvement is: $(S_{AFR} - S_{DCF})/S_{DCF}$



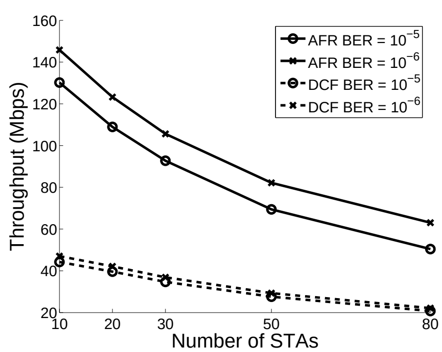
(a) MAC efficiency



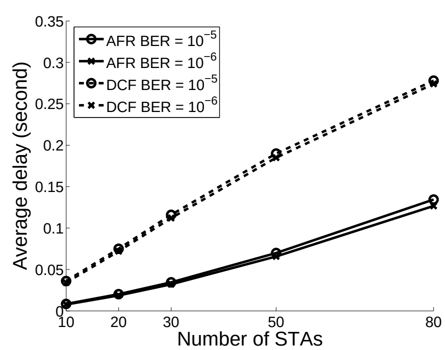
(b) Improvement over DCF

Fig. 10. Simulation results for CBR traffic and different PHY data rates. The x-axis of both (a) and (b) represents the PHY data rate. The y-axis of (a) represents the ratio of the MAC throughput to the PHY data rate. Let S_{AFR} and S_{DCF} be the throughput of AFR and DCF, then The y-axis of (b) represents $(S_{AFR} - S_{DCF})/S_{DCF}$. The parameters are listed in Fig. 1(c) and Table V.

The second example shows how the network's load influences the performance. In this example, the number of STAs is increased and all the other parameters are kept unchanged. As shown in Fig. 11(a), AFR always outperforms DCF. But, the gap between them becomes narrower when the network is heavily loaded. This is due to the fact that we use a constant CW_{min} and CW_{max} for all the simulations, so in a highly populated network collisions happen so often that AFR is not sufficient to alleviate its impact. Another observation is that AFR achieves lower average delay than DCF while it still maintains higher throughput, see Fig. 11(b). In this simulation, we retransmit a frame 4 times if collisions or errors happen, and we limit the sending queue size to be 20 packets. As a result, the measured delay does not increased exponentially with the number of STAs. This is a promising result for the AFR scheme. It could be very useful for multimedia applications whose delay requirements are usually strict. We will show two examples in Section VI-E and VI-F to further explore this characteristic.



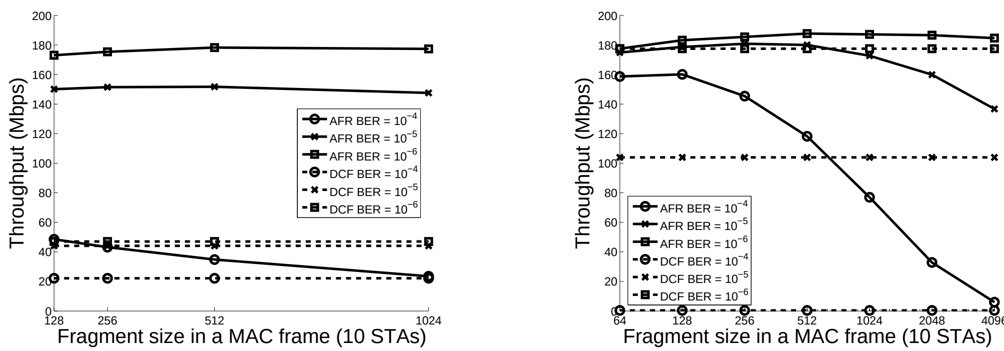
(a) Throughput



(b) Average delay

Fig. 11. Simulation results for CBR traffic with different number of STAs. The parameters are listed in Fig. 1(c) and Table V.

In the third example, we investigate the impact of the fragment size on the system performance. First, for ordinary packet sizes such as 1024 bytes, as shown in Fig. 12(a), 128-byte fragments lead to 30Mbps higher throughput than 1024-byte ones in a very noisy channel (e.g., $BER = 10^{-4}$). The fragment size has negligible impact on throughput in clear channels in which $BER = 10^{-5}$ and $BER = 10^{-6}$, because most of the frames are transmitted successfully, so fragmentation only adds some unnecessary overhead. Second, if *jumbo frames* are to be supported as in Gigabit Ethernet, the packet size would be very large. For example, Intel Pro 1000 Ethernet Adapter even supports a huge packet size of 16110 bytes[6]. Here, we simulate a 8192-byte packet size as shown in Fig. 12(b). In this example, fragmentation is critical important for the cases where $BER = 10^{-4}$ and $BER = 10^{-5}$. In particular, when $BER = 10^{-4}$, AFR with 512-byte fragments achieves more than 100Mbps than AFR with 4096-byte fragments. Moreover, DCF in the $BER = 10^{-4}$ case can barely transmit anything (throughput is almost zero).



(a) Packet size = 1024 bytes (b) Packet size = 8192 bytes

Fig. 12. Simulation results for CBR traffic with different fragment sizes. The parameters are listed in Fig. 1(c) and Table V.

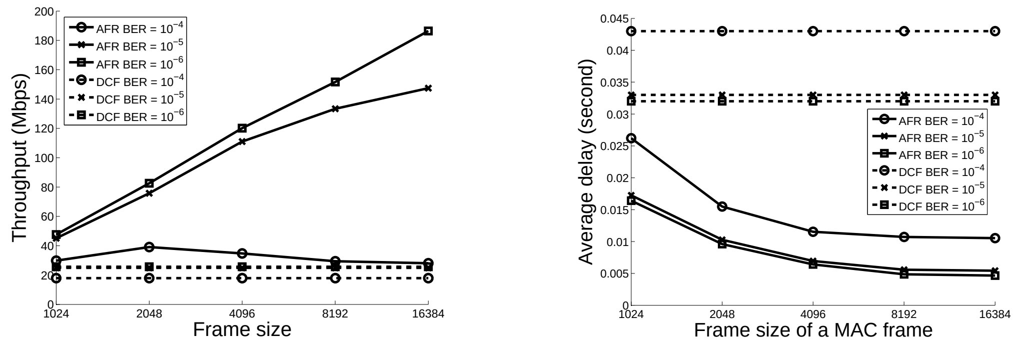
Comparing Fig. 12(a) to Fig. 12(b), we draw the conclusion that fragmentation is of great importance in noisy channels, especially for *jumbo frames*.

We test another situation to see the impact of frame size as shown in Fig. 13. In this example, packets are not fragmented. Clearly, in a noisy channel ($BER = 10^{-4}$), AFR achieves higher efficiency than DCF, but only for smaller frame sizes such as 1024 and 2048 bytes, as shown in Fig. 13(a). This is because the Sq length is short (it contains maximum 10 packets), and the channel is very noisy, thus the actual frame size is likely to be less than the desired one. In a less noisy channel (e.g., $BER = 10^{-5}$ and $BER = 10^{-6}$), AFR keeps increasing the performance with increased frame size. The ability to keep high performance for large frames is an important attribute of the AFR scheme. In Fig. 13(b), the average delay of AFR is less than that of DCF. The larger the frame size the lower the average delay.

D. TCP traffic

TCP is such a popular and successful transport layer protocol that the ability of a new MAC scheme to support it is obviously a must. Therefore, we now investigate the support of the AFR scheme for TCP traffic.

[6]http://darkwing.uoregon.edu/~joe/jumbo-clean-gear.html



(a) Throughput (b) Average delay

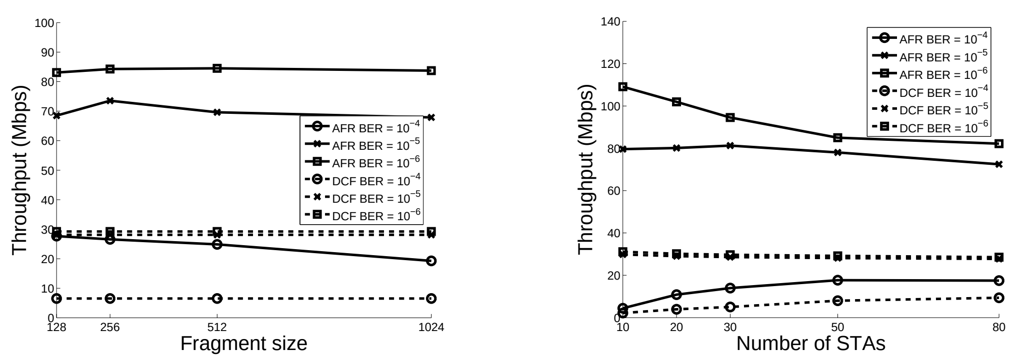
Fig. 13. Simulation results for CBR traffic with different frame size. The parameters are listed in Fig. 1(c) and Table V.

The most important difference between TCP and CBR is that there are ACKs in TCP, which are small packets from the viewpoint of MAC. For the AFR scheme, small packets will decrease effectiveness since it is hard to aggregate enough of them for a large frame. The good news is that TCP is an aggressive protocol which increases its sending rate after successful transmission, thus the channel tends to be heavily loaded.

All the results we report here are for long-lived TCP SACK with FTP as the application. The application layer packet length is 984 bytes. By adding 40 bytes TCP and IP headers, the MAC frame size is 1024 bytes.

First, we test a WLAN with 50 STAs. From Fig. 14(a) we can see that AFR achieves considerable gains over DCF in all channel conditions. For a channel with $BER = 10^{-5}$ and $BER = 10^{-6}$, AFR outperforms DCF significantly but fragment sizes have unnoticeable impacts. The impact of the fragment size becomes considerable when BER is 10^{-4}. This behavior of the TCP traffic is just like that of the CBR traffic in Fig. 12(a).

Second, we increase the number of STAs from 10 to 80 (Fig. 14(b)). AFR still achieves higher throughput than DCF in all channels. Meanwhile, thanks to the TCP's self-adaptive ability, throughput of the TCP traffic is less sensitive than that of the CBR traffic as shown in Fig. 11(a).



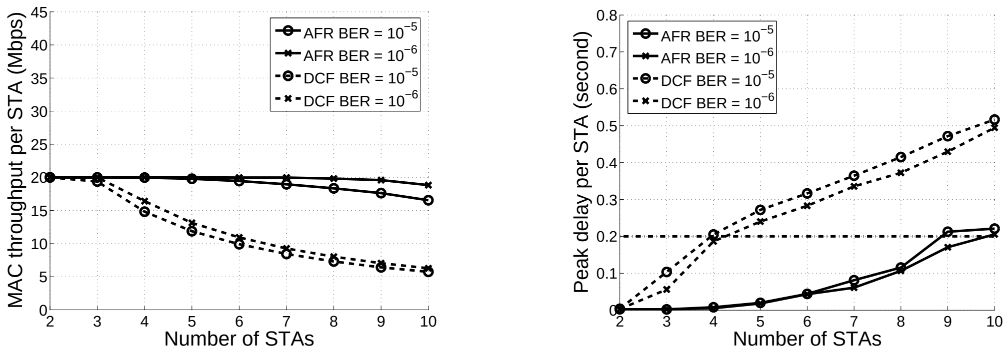
(a) Throughput (50 STAs) (b) Throughput (80 STAs)

Fig. 14. Simulation results for TCP traffic. The parameters are listed in Fig. 1(c) and Table V.

E. HDTV

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

In this example, we use a 432Mbps PHY data rate, and a 9000-byte frame size for the AFR scheme. As we increase the number of STAs in the network, we check if the requirements of HDTV are still satisfied and illustrate the results in Fig. 15. In such a network, DCF can only support 2 simultaneous HDTV streams, but AFR can support 6 and 9 streams for $BER = 10^{-5}$ and $BER = 10^{-6}$ respectively, which means more than 400% improvement. This again demonstrates the advantage of AFR for high sending rate applications in very high-speed WLANs.



(a) Throughput (b) Peak delay

Fig. 15. Simulation results for HDTV traffic. The parameters are listed in Fig. 1(c) and Table V.

F. VoIP

The last application that we consider is VoIP, which is basically an UDP stream with a varying and low-speed rate (96Kbps) and a small packet size (120 bytes) according to the IEEE 802.11n requirements [8]. VoIP is a challenging application for CSMA/CA based WLANs because it has a limited bandwidth requirement and small packet sizes. Thus there may not be enough packets for AFR to aggregate, thus DCF and AFR are expected to achieve more or less the same performance. In this example, we show that a WLAN with pure VoIP traffic.

To characterize the variety of the sending rate, we use the Brady's model [32] in which both ON and OFF period of the traffic are 1500 ms. To compare the results, we use a criterion in which the network can tolerate less than 1% of packets with delays larger than 15 ms. As shown in Table VI, DCF fails to meet this requirement starting from 100 STAs while $BER = 10^{-4}$, and AFR's loss percentages are always much less than DCF's. The reason for this behaviour is mainly due to the self-adaptive ability of the zero-waiting mechanism. This simulation demonstrates that AFR is still suitable for traffic with low rate and small packet size such as VoIP.

	10	50	80	90	100
AFR ($BER = 10^{-4}$)	0.0%	0.0%	0.0%	0.0%	0.22%
AFR ($BER = 10^{-5}$)	0.0%	0.0%	0.0%	0.0%	0.0008%
AFR ($BER = 10^{-6}$)	0.0%	0.0%	0.0%	0.0%	0.0%
DCF ($BER = 10^{-4}$)	0.0%	0.0%	0.0%	0.0%	**3.085%**
DCF ($BER = 10^{-5}$)	0.0%	0.0%	0.0%	0.0%	0.430%
DCF ($BER = 10^{-6}$)	0.0%	0.0%	0.0%	0.0%	0.322%

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. THE PARAMETERS ARE LISTED IN FIG. 1(C) AND TABLE V.

G. *Comments*

The conclusions that may be drawn from our simulations are summarized as follows:

- First, the simulation results confirm the conclusion drawn from the theoretical analysis in Section V, that is in the AFR scheme, the longer the frame size the better.
- Second, the simulations further confirm that fragmentation is necessary, show that this is particularly true in noisy channels for *jumbo frames*.
- Third, even though new queues are added in AFR, its delay performance is much better than that of DCF due to the zero-waiting technique.
- Fourth, complicated applications such as TCP, despite the presence of small ACK packets, do not make the AFR scheme ineffective.
- Fifth, AFR is particularly effective for rich media applications such as HDTV.
- Last, AFR achieves better performance than DCF for applications with low-rate and small-sizes such as VoIP.

VII. CONCLUSION

The basic impetus of this work is to enhance the MAC layer for very high-speed WLANs. To this end, we have designed and implemented a new MAC scheme — the AFR scheme. The rationale of AFR is to aggregate as many as possible packets from the upper layer into large frames. The large frames are then divided into fragments before transmission. If errors occur, only fragments that are acknowledged with errors will be retransmitted. To support the functionalities envisaged, new MAC frame formats and the corresponding dynamic logic including timing, queueing, and retransmission mechanisms are designed and implemented in the *NS-2* simulator.

A theoretical model has been designed to evaluate the saturation throughput of the AFR scheme. This model is used to compute the optimal frame and fragment sizes, and to compare AFR with the related schemes.

Extensive simulations have been carried out for different scenarios. From the results we have drawn the following conclusions: First, the AFR scheme is very effective for WLANs with very high-speed PHY layers. Second, its behavior for applications with high sending rates, or large packet sizes, or both, is very promising. Third, for low sending rate and small packet size applications, the performance of AFR is still better than that of DCF.

The objective of this paper is to show the potential and efficiency of the aggregation idea, thus several possible optimization techniques for the CSMA/CA are not addressed. Combined with these, an integrated solution may be more effective. These techniques include:

- Backoff optimization for WLANs: to curb the inefficiency caused by the exponential backoff, a lot of work has been done (e.g., [12], [15], [35]). Recently, non-exponential backoff is also proposed [18].
- Aggregation can also be combined with Block ACK of 802.11e [3] to further improve efficiency, i.e., only one ACK is used for a train of large frames instead of one.
- Two-way aggregation is another method, in which large frames piggyback in the ACK frames [4] [23].

REFERENCES

[1] IEEE std 802.11-1999, 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, Sept. 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, February 2004.

[4] S. A. Mujtaba, et. al., “TGn Sync Proposal Technical Specification”, www.tgnsync.org, IEEE 802.11-04/889r6, May. 2005.

[5] Q. Ni, T. Li, T. Turletti and Y. Xiao, “AFR partial MAC proposal for IEEE 802.11n”, IEEE 802.11-04-0950-00-000n, Aug. 2004.

[6] J. Ketchum, et. al., “System Description and Operating Principles for High Throughput Enhancements to 802.11”, IEEE 802.11-04/0870r0, Aug. 2004.

[7] M. Singh, B. Edwards, et. al., “System Description and Operating Principles for High Throughput Enhancements to 802.11”, IEEE 802.11-04-0886-00-000n, Aug. 2004.

[8] A. P. Stephens, et. al., “IEEE P802.11 Wireless LANs: Usage Models”, IEEE 802.11-03/802r23, May. 2004.

[9] Magis Networks White Paper, ”IEEE 802.11 e/a Throughput Analysis”, 2004, www.magisnetworks.com.

[10] NS, http://www.isi.edu/nsnam/ns/

[11] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function”, IEEE Journal on Selected Areas in Communications, Vol. 18, Number 3, March 2000.

[12] J. Choi, J. Yoo, S. Choi and C. Kim, “EBA: An Enhancement of the IEEE 802.11 DCF via Distributed Reservation”, IEEE Transactions on Mobile Computing, Apr. 2004.

[13] T. Cover, J. Thomas, “Elements of Information Theory”, John Wiley & Sons, 1991.

[14] R. Gallager, “Information Theory and Reliable Communication”, John Wiley & Sons, 1968.

[15] Y. Kwon, Y. Fang and H. Latchman, “A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs”, IEEE INFOCOM 2003.

[16] S. Kim, Y. Kim, S. Choi, K. Jang and J. Chang, “A High-Throughput MAC Strategy for Next-Generation WLANs”, IEEE WoWMoM 2005.

[17] M. Heusse, F. Rousseau, G. Berger-Sabbatel and A. Duda, “Performance Anomaly of 802.11b”, INFOCOM 2003.

[18] M. Heusse, F. Rousseau, R. Guillier and A. Duba, “Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs”, SIGCOMM 2005.

[19] R. Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experiment Design, Measurement, Simulation and Modeling”, John Wiley and Sons, Inc, 1991.

[20] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia, "Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs", MOBICOM 2004.

[21] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, 1975.

[22] P. Lettieri and M. B. Srivastava, "Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency", INFOCOM 1998.

[23] C. Liu and A. Stephens, "An Analytic Model for Infrastructure WLAN Capacity with Bidirectional Frame Aggregation", IEEE WCNC 2005.

[24] T. Li, Q. Ni, T. Turletti, and Y. Xiao, "Performance Analysis of the IEEE 802.11e Block ACK Scheme in a Noisy Channel", IEEE Broadnets 2005.

[25] J. S. Ma, "On the Impact of HDLC Zero Insertion and Deletion on Link Utilization and Reliability", IEEE Transactions on Communications, 1982.

[26] Q. Ni, T. Li, T. Turletti and Y. Xiao, "Saturation Throughput Analysis of Error-Prone 802.11 Wireless Networks", Wiley Journal of Wireless Communications and Mobile Computing (JWCMC), Vol. 5, No. 8, Dec 2005, pp. 945-956.

[27] D. Niculescu, S. Ganguly, K. Kim and R. Izmailov, "Performance of VoIP in a 802.11-based Wireless Mesh Network", INFOCOM 2006.

[28] V. Vitsas, et. al., "Enhancing performance of the IEEE 802.11 Distributed Coordination Function via Packet Bursting", GLOBECOM 2004.

[29] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly, "Opportunistic Media Access for Multirate Ad hoc networks", MOBICOM 2002.

[30] G. Tan and J. Guttag, "The 802.11 MAC Protocol Leads to Inefficient Equilibris", INFOCOM 2005.

[31] J. Tourrilhes, "Packet Frame Grouping: Improving IP multimedia performance over CSMA/CA", ICUPC 1998.

[32] W. Wang, S. Liew, and V. O. K. Li, "Solutions to Performance Problems in VoIP over a 802.11 Wireless LAN", IEEE Transactions On Vehicular Technology, Jan. 2005.

[33] 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", IEEE INFOCOM 2002.

[34] Y. Xiao and J. Rosdahl, "Performance analysis and enhancement for the current and future IEEE 802.11 MAC protocols", ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), special issue on Wireless Home Networks, Vol. 7, No. 2, Apr. 2003, pp. 6-19.

[35] X. Yang and N. Vaidya, "A Wireless MAC Protocol Using Implicit Pipelining", to appear in the IEEE Transactions on Mobile Computing.

APPENDIX I: THE MARKOV CHAIN

In [11], Bianchi first introduced a bi-dimensional stochastic process $\{s(t), b(t)\}$ to model the backoff behavior of the legacy DCF. Process $b(t)$ represents the backoff counter, and it is decremented at the beginning of each slot. For an idle slot, the time scale of $b(t)$ corresponds to a real slot time. In a collision slot, however, $b(t)$ is frozen for the duration of this transmission. Whenever $b(t)$ reaches zero the STA transmits and starts another round of backoff regardless of the outcome of the transmission. The new backoff starts from a value selected randomly from 0 to contention window CW. The CW shall be reset after a successful transmission and be doubled up to a maximum value CW_{max} for corrupted cases. This implies that $b(t)$ depends on the transmission history, therefore is a non-Markovian process. To overcome this, another process $s(t)$ is defined to track the contention window size.

This bi-dimensional stochastic process is a Markov chain under the following two assumptions. First, the transmission probability τ is constant in every slot time. Second, at each transmission attempt, regardless of the number of retransmission, each frame is lost with an independent constant probability p_f.

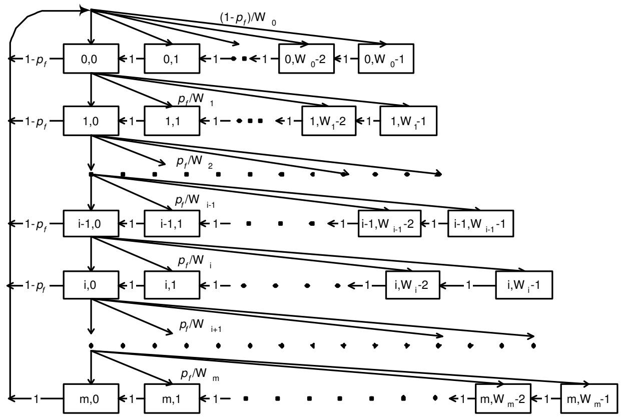


Fig. 16. The Markov chain used in this paper

Under these assumptions, the bi-dimensional stochastic process $\{s(t), b(t)\}$ forms a Markov chain as shown in Fig.16. In this chain, all the states are ergodic because they are aperiodic, recurrent and non-null, thus a stationary solution exists [21]. Given the stationary distribution, we can solve τ and p_f with this Markov chain as follows.

Let us consider the first formula for p_f and τ. In the Markov chain above, p_f stands for the probability that the contention window is doubled because of either collisions or errors. Bianchi's paper assumes there are no errors in the channel, so $p_f = p_c = 1 - (1 - \tau)^{n-1}$ where n stands for the number of STAs in the system. 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 packet error rate. In the AFR scheme, the receiver sends back the ACK frame in both successful and erroneous cases, thus $p_f = p_c$.

Now, we introduce the second formula for p_f and τ. The transmission probability τ in a slot time should be the sum of all the probabilities of the contention window decreases to zero at all the backoff stages. I.e., $\tau = \sum_{i=0}^{m} b_{i,0}$ where *m* is the maximum backoff stage as defined by $CW_{max} = 2^m \cdot CW_{min}$, and $b_{i,0}$ is the probability of the contention window decreases to zero at the stage i. Bianchi's paper assumes that a frame can be retransmitted infinite times, which is inconsistent with the 802.11 specification [1]. Wu et al. loose this assumption in their work [33]. We use formulas (8) and (9) in [33] to solve $b_{i,0}$.

Finally, with these two formulas, a closed form solution for p_f and τ is formed and both of them can be solved. Therefore, we find the last variable τ required in (Equations 5, 6 and 7).