A New MAC Scheme for Very High-Speed WLANs

Tianji Li[*], Qiang Ni[†], David Malone, Douglas Leith
Hamilton Institute, National University of Ireland at Maynooth, Ireland
{tianji.li, david.malone, doug.leith}@nuim.ie

Yang Xiao
University of Memphis, USA
yangxiao@ieee.org

Thierry Turletti
Planète Group, INRIA, France
turletti@sophia.inria.fr

Abstract

We consider the medium access control (MAC) layer for very high-speed Wireless LANs, which is designed to support rich multimedia applications such as high-definition television. In such networks, the physical (PHY) layer data rate is proposed to exceed 216Mbps. The legacy MAC layer, however, greatly restricts the performance improvement due to its overhead. It has been shown that MAC utilizes less than 20% of the transportation ability provided by the PHY layer. To mitigate this inefficiency, we propose an Aggregation with Fragment Retransmission (AFR) scheme, which supports transmissions of very large frames and partial retransmissions in the case of errors. Aggregation allows for increased performance despite per-transmission overhead while partial retransmission alleviates the risk of losing the entire frame. Extensive simulations show that AFR fundamentally outperforms the legacy MAC protocol. It is particularly effective for applications with high data rates and large packet sizes such as HDTV and high-rate UDP traffic. For applications with very low data rates and small packet sizes such as Voice over IP, AFR performs slightly better.

1. Introduction

Delivering rich multimedia applications such as high-definition television (HDTV, 20Mbps), DVD (9.8Mbps), etc. is the main goal for the upcoming Wireless LANs (WLANs) [8]. To support these applications, the physical layer (PHY) rate in such networks is expected to exceed 216Mbps, some 802.11n proposals 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 solution for the MAC inefficiency.

The legacy MAC is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. It has been demonstrated that the overhead of MAC is the basic reason for the inefficiency of WLANs (e.g., [9, 15, 34]). Here, the overhead refers to backoffs, distributed interframe space (DIFS), acknowledgment (ACK), short interframe space (SIFS) and PHY layer header. To combat the overhead drawback, Burst ACK (e.g., [28] [29] and [31]) and Block ACK (e.g., [3], [34]) have been proposed. In these two schemes, backoff are performed for a burst/block of data packets and ACKs. In the future WLANs, however, a PHY header will be transmitted in at least $44\mu s$, which approximates the duration of a 1024-byte packet ($40\mu s$) transmission at 216 Mbps. For small packets such as MAC layer ACKs, or for faster PHY layers, PHY headers will waste more channel resources.

In this paper, we design an Aggregation with Fragment Retransmission (AFR) scheme to further mitigate the overhead. The idea is to send, in only one frame[1], a whole burst/block of packets which are transmitted in multiple frames in Burst ACK or in Block ACK. If errors occur during the transmission, we retransmit only parts of the corrupted frame.

The main novelties lie in the following. A frame format is designed for supporting all the functionalities of AFR. This format allows for higher throughput with less overhead compared to previous proposals. We

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

[†] Qiang Ni is currently with the Electronic & Computer Engineering, School of Engineering & Design, Brunel University, West London, UK. His email: Qiang.Ni@ieee.org

[1] In this paper, we define a *packet* as what the MAC receives from the upper layer, define a *frame* as what the MAC transfers to the PHY, and a *fragment* as a part of a *frame*.

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 IEEE

propose a zero-waiting mechanism, which enhances the aggregation technique with a self-adaptive ability. In addition, we study a fragmentation technique, in which packets longer than a threshold are divided into fragments before being aggregated. This technique is of vital importance for supporting jumbo frame transmissions.

We implemented the AFR scheme in the *NS-2* simulator. This implementation enable us to simulate applications with diverse requirements according to 802.11n's requirements [8]. Besides traditional CBR and TCP traffic, we simulate multimedia applications with diverse characteristics. Results show that AFR outperforms the legacy DCF fundamentally, hence it is a promising MAC technique for very high-speed WLANs.

The remainder of the paper is organized as follows. Related work is introduced in Section 2. In Section 3, we review the legacy 802.11 DCF. Section 4 presents the AFR scheme in detail. Section 5 describes the implementation details and the corresponding simulation results. Finally Section 6 concludes this paper.

2. Related Work

Some ongoing activities about the aggregation are being carried out in the 802.11n working group (e.g., [4], [5], [6], [7]) and in academia (e.g., [16] and [20]).

In [4] and [16], the authors use a clever idea to distinguish packets in a large frame. They suggest to employ a special *delimiter* at the beginning of each packet in a frame. In [16], by using another *delimiter* which is transmitted at basic-rate at PHY, their scheme can also support multi-destinations. This paper is based on the MAC scheme we have proposed to the upcoming IEEE 802.11n standard working group [5]. Our previous proposal supports a constant packet size and assumes that the queue size at the MAC layer is infinite. In this paper, we relax these two restrictions.

Ji et al. [20] used the aggregation technique to solve the unfairness problem in WLAN. They suggest to remove the DIFS, SIFS and backoffs before a series of packets, and transmit them together in a large PHY layer frame. But there is still a small PHY header ($12\mu s$) for each packet. As explained before, the PHY header will be even longer in very-high speed WLANs and could be a major source of overhead, thus should be removed if possible.

3. The Legacy DCF

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.

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 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].

4. The AFR Scheme

4.1. 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 and retransmits only fragments when errors are detected by their Frame Check Sequence (FCSs). An example of the AFR scheme is shown in Fig. 1. In particular, at the sender, every outgoing packet is segmented according to a fragmentation threshold. Before transmission, all the fragments

n	Number of STAs
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)

Table 1. Notation used in this paper

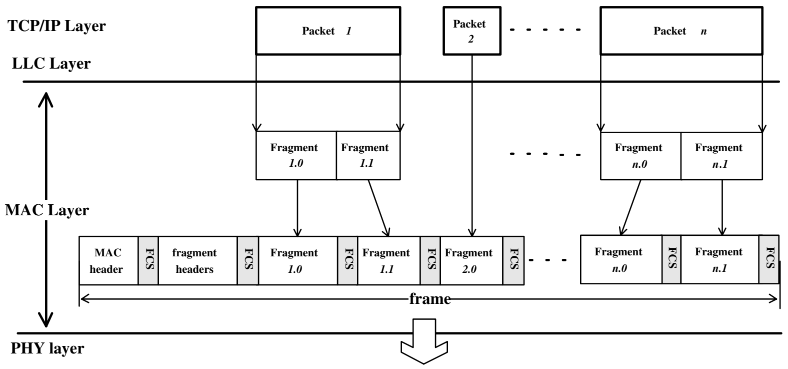


Figure 1. The Aggregation with Fragment Retransmission (AFR) scheme.

are marked as 'undelivered' and kept temporarily in a MAC layer sending-queue (Sq). MAC 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 large enough. Then, MAC transmits this frame (see Fig. 2) according to the DCF procedure.

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. 3) 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 4.2.2 for the reason.

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

4.2. Design Issues

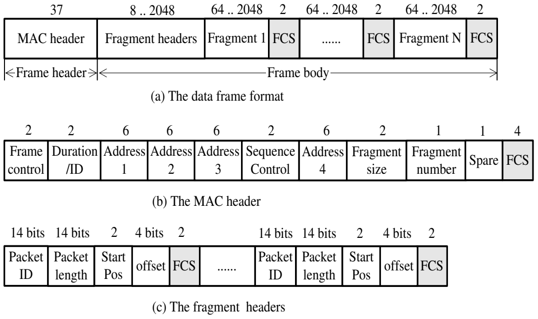
4.2.1. Frame Formats

Clearly, new data and ACK formats are the first concern of the AFR scheme. This is not easy due to: After an erroneous transmission, the receiver should be able to retrieve the correctly transmitted fragments. This is hard because the sizes of the corrupted fragments may be unknown to the receiver. In the meantime, 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. 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 FCSs (Fig. 2(b) and (c)).

The fragment-headers in 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 may be corrupted during the transmission.

The fragment header consists 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

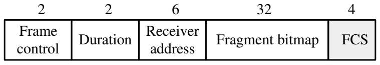


(a) The data frame format

(b) The MAC header

(c) The fragment headers

Figure 2. Data format in the AFR scheme.



The ACK frame format

Figure 3. ACK format in the AFR scheme.

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 $offset = \lfloor pLEN/fragment\ size \rfloor$ 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
```

belongs. *StartPos* is used to indicate the position of the fragment body in this frame and *offset* is for recording the position of this fragment in packet P.

The new ACK format is simple, a 32-byte bitmap field is added. Each bit of the bitmap is used to indicate the correctness of a fragment.

To clarify the usage of the new formats, we give an example below. Suppose there are two packets (pkt_1 and pkt_2) with length $l_1 = 1025$ bytes and $l_2 = 40$ bytes in the Sq, frame length is $l^f = 2048$ bytes and fragment length is $l^{frag} = 512$ bytes. 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. 2.

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

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

	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 2. A usage of the frame formats.

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.

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].

4.2.3. 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 not adequate fragments 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.

4.3. Comments

4.3.1. Optimal frame and fragment

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. In [23], we use an analytical model to show that 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]. Also using the same model, we derive the optimal fragment sizes in [23].

4.3.2. Fairness

AFR strictly follows the basic principle of CSMA/CA, therefore the same fairness characteristics hold as in the legacy DCF. Improvements to the DCF fairness issue are all suitable for AFR. Interested readers should refer to [20], [18], [29] or the TXOP mechanism in IEEE 802.11e [3].

4.3.3. 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 in the MAC header.

Adding more fields in fragment headers will result in more overhead, which is unavoidable. But comparing to the solution in the literature [16], our scheme would still have smaller overhead. To support multi-destinations, the authors of [16] propose to use a *physical delimiter*, which is transmitted at 6Mbps. The *delimiter* technique needs special PHY layer supports [25]. The corresponding analysis is missing in their paper. Moreover, using a 6Mbps rate 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.

5. Simulations

5.1. The simulation setup

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

In the *NS-2* version 2.27 [10], PHY header is 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.

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

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

where n stands for the number of STAs and S_i stands for 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:

$$p_e = 1 - (1 - p_b)^{L_f}. \qquad (2)$$

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[2].

- RTS/CTS: We do not use RTS/CTS in our simulations. Basically the RTS/CTS technique does not change the running logic of both AFR and 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 would unlikely be a good option in single-hop WLANs.

5.2. 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.

- 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.

5.3. 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 retransmissions. Thus we use it as a basic test for evaluating the functionalities of the MAC layer.

For the first example, we compare AFR and DCF while increasing the PHY data rates. As illustrated in Fig. 4(a), the DCF scheme's efficiency is always bounded by that of the ideal case. But the AFR scheme exceeds this limit easily and improves MAC efficiency to around 60% and 35% for 54Mbps to 432Mbps PHY rates. Comparing to DCF, the improvement[3] of AFR ranges from 50% up to 200% (Fig.4(b)).

In this example, we use a constant PHY header duration (20μs) 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 of AFR over DCF will be greater than shown in this example.

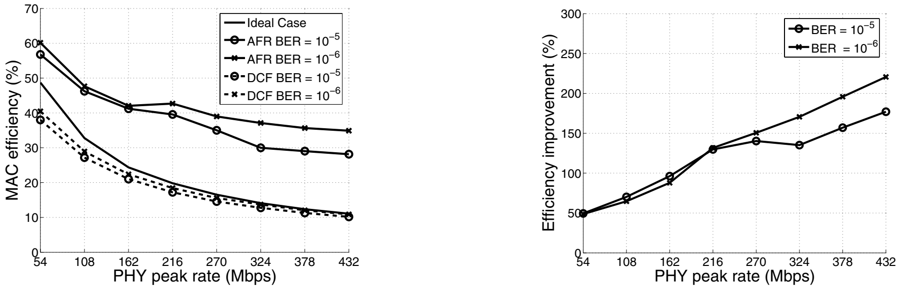
The second example shows how the network's load influences the performance. In this example, the number of STAs is increased and all the other parameters are kept unchanged. As shown in Fig. 5(a), AFR always outperforms DCF. But, the gap between them becomes narrower when the network is heavily contended.

[2] 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 this case is higher than in Gaussian channels, because fewer retransmissions are required [9].

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

	Fig. 4	Fig. 5	Fig. 6(a)	Fig. 6(b)	Fig. 7	Table 4
Number of STAs (n)	10	varied	10	10	varied	varied
App. rate (Mbps)	R / n	54	54	54	20	0.096
Data rate (Mbps) (R)	varied	432	432	432	432	432
Basic rate (Mbps)	R / 9	54	54	54	54	54
AFR Sq (packets)	10	10	10	10	10	10
AFR IFQ (packets)	10	10	10	10	10	10
DCF IFQ (packets)	20	20	20	20	20	20
Packet (bytes)	1024	1024	1024	8192	1500	120
DCF frame (bytes)	1024	1024	1024	8192	1500	120
AFR frame (bytes)	8192	8192	8192	8192	9000	1200
AFR fragment (bytes)	512	256	varied	varied	750	120

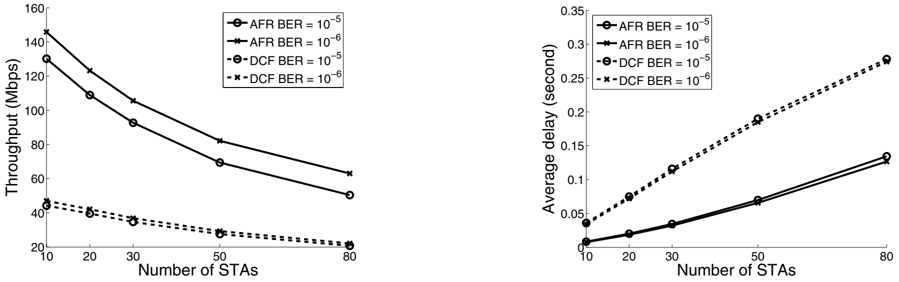
Table 3. Parameters used in *NS-2* simulations.



(a) MAC efficiency

(b) Improvement over DCF

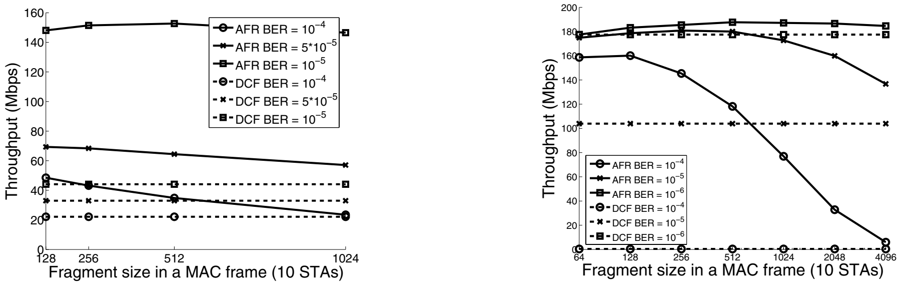
Figure 4. Simulation for CBR traffic with different PHY data rates. Parameters are listed in Table 3.



(a) Throughput

(b) Average delay

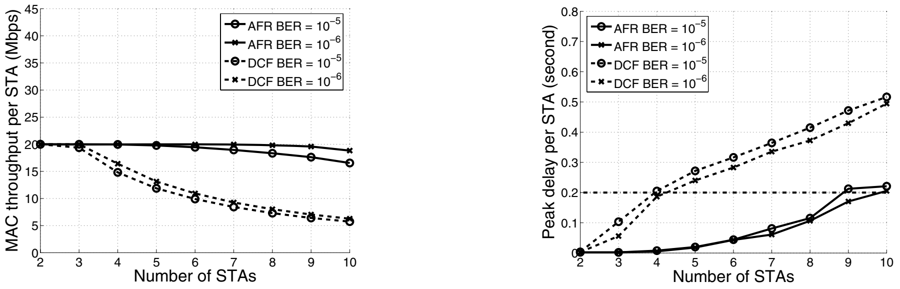
Figure 5. Simulation for CBR traffic with different number of STAs. Parameters are listed in Table 3.



(a) Packet size = 1024 bytes

(b) Packet size = 8192 bytes

Figure 6. Simulation for CBR traffic with different fragment sizes. Parameters are listed in Table 3.



(a) Throughput

(b) Peak delay

Figure 7. Simulation results for HDTV traffic, parameters are listed in Table 3.

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. 5(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 5.4 and 5.5 to explore this characteristic.

In the third example, we investigate the impacts of the fragment sizes. As demonstrated in Fig. 6(a), in a very noisy channel (e.g., $BER = 10^{-4}$), 128-byte fragments lead to 30Mbps higher throughput than 1024-byte ones. In a channel where $BER = 5 * 10^{-5}$, AFR with 128-byte fragments achieves around 10Mbps higher throughput than AFR with 1024-byte fragments. Moreover, the fragment size has negligible impact on throughput in very clear channels such as $BER = 10^{-5}$, because most of the frames are transmitted successfully without errors, and fragmentation only adds some unnecessary overhead. Therefore, fragmentation is useful in a noisy channel.

In Fig. 6(a), the packet size is 1024 bytes. If we need to support *jumbo frame* which is widely used in Gigabit Ethernet, the packet size would be very large. Intel Pro 1000 Ethernet Adapter even supports a huge packet size of 16110 bytes[4]. We simulate a 8192-byte packet size as shown in Fig. 6(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 a less than 512-byte fragment size achieves more than 100Mbps than AFR with 4096-byte does. Moreover, DCF in the $BER = 10^{-4}$ case can barely transmit anything (throughput is almost zero). This example confirms the importance of fragmentation.

5.4. 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. 7. 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 rate applications in very high-speed WLANs.

5.5. 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. For this kind of application there would be not enough packets for AFR to aggregate, thus DCF and AFR are expected to achieve more or less the same performance.

To characterize the variety of the sending rate, we use the Brady's model [32] in which both ON and OFF

[4] http://darkwing.uoregon.edu/˜joe/jumbo-clean-gear.html

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 4, 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 4. 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 Table 3.

6. 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. Thus, the frames will be very large as long as there are enough packets to be aggregated. The large frames in AFR are divided into fragments before transmission. If errors occur, only fragments that are acknowledged with errors will be retransmitted. To support the functionalities provided by AFR, new MAC frame formats and the corresponding dynamic logic including timing, queueing, and retransmission mechanisms are designed and implemented in the *NS-2* simulator.

Extensive simulations have been carried out for different scenarios. From the results we drew the following conclusions: First, the AFR scheme is very effective in WLANs with very high-speed PHY layer. Second, its behavior for applications with high sending rate, or large packet size, 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 are not addressed. Combined with them, an integrated solution may be more effective. These techniques include:

- Backoff optimization for WLANs: to curb the inefficiency caused by exponential backoff, much work has been done before (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 frame.

- Two-way aggregation is another method, in which large frames piggyback in the ACK frames [4] [22].

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 802.11-04/889r0, Aug. 2004.

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

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

[23] T. Li, Q. Ni, D. Malone, D. Leith, T. Turletti, and Y. Xiao, "Aggregation with Fragment Retransmission in Very High-Speed WLANs", in submission.

[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", accepted to appear in Wiley Journal of Wireless Communications and Mobile Computing (JWCMC).

[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.