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

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

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

I. INTRODUCTION

Wireless LANs based on 802.11 technology are becoming increasingly ubiquitous. With the aim of supporting rich multimedia applications such as high-definition television (HDTV, 20Mbps) and DVD (9.8Mbps), the technology trend is towards increasingly higher bandwidths. Some recent 802.11n proposals seek to support PHY rates of up to 600 Mbps ([4], [6], [7], [40]). However, higher PHY rates do not necessarily translate into corresponding increases in MAC layer throughput. Indeed, it is well known that the MAC efficiency of 802.11 typically decreases with increasing PHY rate [9], [41]. The reason is that while increasing PHY rates lead to faster transmission of the MAC frame payload, overhead such as PHY headers and contention time typically do not decrease at the same rate and thus begin to dominate frame transmission times. This behaviour is illustrated in Figure 1, where it can be seen that even under best case conditions the MAC efficiency falls from 42% at a PHY rate of 54Mbps to only 10% at 432Mbps.

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

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

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

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

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

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

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

[1]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 part(s) of a frame.

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

TABLE I

NOTATION USED IN THIS PAPER.

[a] $T_{ack} = T_{SIFS} + T_{hdr}^{phy} + T_{ack}^{pld} + T_{DIFS}$, where T_{ack}^{pld} denotes the time duration to transmit an ACK frame. Note that we define T_{ack} in this way for notation brevity.

[b] $T_{EIFS} = T_{ack}$

waiting aggregation scheme, the frame sizes adapt automatically to the PHY rate and channel state, thereby maximising the throughput efficiency while minimising the holding delay.

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

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

II. MOTIVATION

A. DCF and Its Inefficiency

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

[2] In the DCF scheme, there is only one packet in each frame, so the packet size and the payload size of one frame are the same.

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

$$\eta = \frac{T_p}{T_p + T_{oh}^p} \tag{1}$$

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

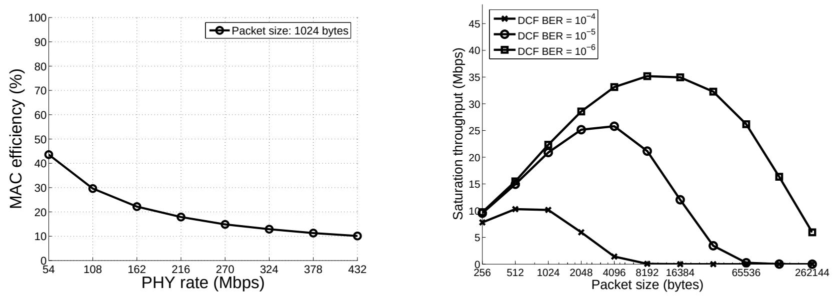
As the PHY rate increases, the contention time T_{CW} does not decrease towards zero due to the constraints placed on the minimum slot size by clock synchronisation requirements and on DIFS by the need for backward compatibility. Similarly, the duration of the PHY header is not expected to decrease with increasing PHY rate owing to backward compatibility and PHY-layer channel equalisation requirements [4]. Thus as the PHY rate is increased, the time to transmit a frame quickly becomes dominated by the fixed overheads associated with the PHY header, contention time etc. Much work has been done to minimise the contention time component of the overhead by regulating the randomized backoff process (e.g., [13] [42] [31]) to reduce the number of collisions and idle slots. However, in very high-speed networks, the MAC throughput efficiency is still intolerable even without these problems. For example, we illustrate in Fig. 1(a) the efficiency in the ideal case where the channel is perfect with neither collisions nor errors [41], hence the overhead of the backoff process is minimised. It can be seen that the efficiency decreases dramatically as the PHY rate increases. In a 216Mbps WLAN, the efficiency is only about 20%. When PHY data-rate increases to 432Mbps, the efficiency decreases to around 10%.

B. Burst ACK and Block ACK

The Burst ACK (e.g., [36] [33] and [32]) and Block ACK (e.g., [3], [41]) schemes have been proposed in the literature for improving efficiency. Burst ACK performs the backoff process once for a series of data and ACK frames (See Fig. 6 for details), while Block ACK goes one step further by using a single ACK frame for multiple data frames (Fig. 6), thus reducing the number of ACKs and SIFS.

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

However, the per packet PHY header overhead T_{hdr}^{phy} and the MAC header overhead T_{hdr}^{mac} are left untouched. According to the proposal 802.11n [4] for the future WLANs, it is likely to take at least $44\mu s$ to transmit a PHY header (and $48\mu s$



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

(a) MAC inefficiency in ideal case (b) DCF with very large packets (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 rate. (b) Large frames transmission in DCF where PHY rate is 54 Mbps. (c) MAC and PHY parameters used.

when two antenna radios are used [4]). For comparison, the transmission duration of a 1024-byte frame at a PHY rate of 216Mbs is $40\mu s$, and at 432Mbs is $20\mu s$. As the PHY rate is increased, the time to transmit a frame quickly becomes dominated by PHY headers, the throughput efficiency rapidly decreases and efforts to increase the system capacity purely by increasing the data rate are thus of limited effectiveness even when Burst ACK or Block ACK are employed.

C. *Aggregation Schemes*

Aggregation schemes seek to amortize the PHY header overhead across multiple packets. This is achieved by transmitting multiple packets in a single large frame. However, there is a traditional dislike for transmitting large frames in wireless networks since in a noisy channel (e.g., $BER \geq 10^{-5}$), the throughput can fall as larger frames are used [23]. We illustrate this in Fig. 1(b). However, we note that in traditional retransmission schemes a whole frame is retransmitted even if only one bit is lost. This raises the question of whether it is possible to retransmit only the erroneous part(s) of a frame – if properly designed, such partial retransmission could be expected to improve performance. This is a key motivation of the work presented here.

Although this idea seems simple at first glance, it is actually a radical challenge for PHY and MAC technology. From the PHY viewpoint, the traditional small-packet rule does not hold any more. The PHY layer 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]. From the MAC viewpoint, any retransmission scheme carries an associated signalling overhead and hence a trade-off exists between system efficiency and the granularity of retransmission. Moreover, since real traffic is typically bursty/on-off in nature, this raises questions as to the optimal policy for aggregating packets into frames, for example how long should the MAC wait for sufficient packets to arrive to form a large frame.

Our previous work on aggregation schemes resulted in a proposal for the forthcoming IEEE 802.11n standard. In [5], [25] we proposed to aggregate multiple packets into a single large frame and, should an error occur, the damaged packets are retransmitted. The present paper substantially extends this previous work, see Section II-D. In parallel with our work, there are other activities in the 802.11n standard working group on this topic (e.g., [4], [6], [7]). These support similar functionalities to our scheme, with a special *delimiter* for locating each fragment in a frame. Other related work includes that of Ji et al. [21] where an aggregation technique is used to solve an unfairness problem in WLANs. Ji et al. suggest removing the DIFS, SIFS and backoffs before a series of packets, transmitting the packets together in a large PHY layer frame. However, a small PHY header ($12\mu s$) is used to identify each packet within a frame. In [17], a two-level (one at MAC, another at PHY) aggregation scheme is proposed that uses a similar *delimiter* to that in the TGn Sync proposal [4].

D. *Open Questions*

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

- How do we aggregate packets? The frames we want are larger than a typical packet. If the packets from the upper layer are large and arrive rapidly, then aggregation is simple. If not, should a timing mechanism be used to wait for sufficient packets to arrive to form a large frame? If so, how long do we wait to maximise throughput while minimising delay?
- What is an appropriate transmission and retransmission unit? Should very large packets be divided for retransmission?
- A suitable analysis of aggregation throughput and delay performance is missing.
- How does packet aggregation impact application traffic, e.g. voice, video and data traffic.

We address these open questions in this paper.

III. Fundamental Considerations

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

A. *Throughput Efficiency*

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

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

$$\eta_p = \frac{T_p}{T_p + T_{oh}^p} = \frac{L_p/R}{L_p/R + T_{oh}^p} \tag{2}$$

We can see that $T_p = L_p/R$ scales with $1/R$. In order to maintain throughput efficiency η_p, we require that the per packet overhead T_{oh}^p also scales with $1/R$. Considering T_{oh}^p in more detail, we can typically decompose it into the following elements (where r denotes the number of transmissions before all fragments from this packet are transmitted successfully, and other notation is listed in Table I):

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

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

- The number of packets M in a frame should be proportional to R, that is $M = bR$ for some constant b. This ensures that the overhead T_{hdr}^{phy}, T_{hdr}^{mac}, T_{ack} and T_{CW} translate into a per packet overhead that scales with R.
- Since there is only one MAC header and one ACK per frame, when M is proportional to R there is no fundamental constraint on the rate at which MAC headers and ACK frames are transmitted. The same is not true for fragment headers.
- For a given fragment size L_{frag}, the number of fragments in a frame m increases with the number of packets M in a frame, i.e., $m = m'M$ where m' is the number of fragments per packet, we thus have $m = m'bR$ when $M = bR$. Hence, for T_{hdr}^{frag}/M to scale with $1/R$ the rate at which fragment headers are transmitted must be chosen proportional to R, in which case $T_{hdr}^{frag}/M = mL_1/R = m'L_1/R$.

When the per packet overhead satisfy these conditions, the per packet throughput efficiency is

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

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

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

$$\tilde{\eta_p} \quad = \quad \frac{L_p}{L_p + r \cdot m' \cdot L_1} = \frac{1}{1+d} \tag{5}$$

where $d = (rm'L_1)/L_p$.

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

B. Zero-waiting

When packets are large and arrive rapidly from the upper layer, it is straightforward for the MAC layer to assemble these into large frames. However, it is also common for packets to arrive more slowly or in bursts (e.g., packets from a VoIP stream, video traffic, web traffic etc). One approach is then to consider waiting at the MAC until sufficient packets arrive to form the desired size of frame. However, it turns out that when the scaling conditions in Section III are satisfied, fundamentally there is no need to wait for packets to accumulate and it is sufficient instead to simply start a transmission whenever the MAC wins a transmission opportunity. In this case a frame is formed by aggregating the currently queued packets. Evidently, such a policy minimises holding delay at the MAC layer. We show that this opportunistic aggregation policy also maximises network throughput where feasible.

We first characterise the maximum achievable efficiency and the maximum throughput that any aggregation MAC scheme can support. Assuming that there are no collisions and errors in the network[3], corresponding to $r = 1$, we can write the **per frame** throughput efficiency as

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

and it is straightforward to show that the maximum achievable efficiency $\eta_{max} = \tilde{\eta_p}$ and the maximum throughput $S_{max} = R/(1+d)$.

Let the mean arrival rate of the offered load be $\nu = \alpha S_{max} = \alpha R/(1+d)$ bits per second. During the time $T_f + a + d \cdot T_f$ to transmit a frame, on average we expect $\nu \cdot (T_f + a + d \cdot T_f)$ arrivals at the queue. Selecting the frame size to be the same as queue size $q(k)$, we have that,

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

These queue dynamics can be written as

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

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

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

[3]The proof for more complicated cases is left as further work.

Combining Equation (6) and (8), we derive that

$$\eta_f = \frac{\alpha}{1+d} = \alpha \cdot \eta_{max}$$

As $\alpha \to 1$, we can see that the zero waiting policy achieves the maximum efficiency.

From equation (8), we can see two important features of zero-waiting:

First, when the offered load is light (i.e., α is small) small frames will be used. As the load increases, larger frame sizes will be automatically used. Thus, zero-waiting elegantly creates a feedback loop whereby throughput efficiency is regulated based on queue backlog. When the current level of efficiency is too low for the offered load, a queue backlog will develop which in turn induces larger frames and increased efficiency. The frame size used thus adapts to the minimum required to service the offered load.

Second, for a given level of load α, the frame size L_f scales with R. Therefore, with a multi-rate enabled wireless card, the frame size also adapts automatically to changing PHY rate R.

IV. The AFR Scheme

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

A. AFR Implementation

Clearly, new data and ACK frame formats are a primary concern in developing a practical AFR scheme. Difficulties for new formats include (i) respecting the constraints on overhead noted previously and (ii) ensuring that in an erroneous transmission the receiver is able to retrieve the correctly transmitted fragments – this is not straightforward because the sizes of the corrupted fragments may be unknown to the receiver.

In our scheme, a MAC frame consists of a frame header and a frame body (Fig. 2(a)). In the new MAC header, all the fields of the DCF MAC header remain unchanged, and we add three fields — *fragment size*, *fragment number* and a *spare* field. The *fragment size* represents the size of fragment used in the MAC frames. The *fragment number* represents the number of fragments in the current MAC frame. The *spare* field is left for future extension and maintaining alignment. The frame body consists of fragment-headers, fragment bodies and the corresponding Frame Check Sequences (FCS) (See Fig. 2(b) and (c)).

The fragment-header section of the frame body has a variable size. It includes from 1 to 256 fragment headers, each of which is protected by a FCS. The length of each fragment header is constant (8 bytes) and known to both the sender and the receiver. For the receiver, it knows where the first 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*.

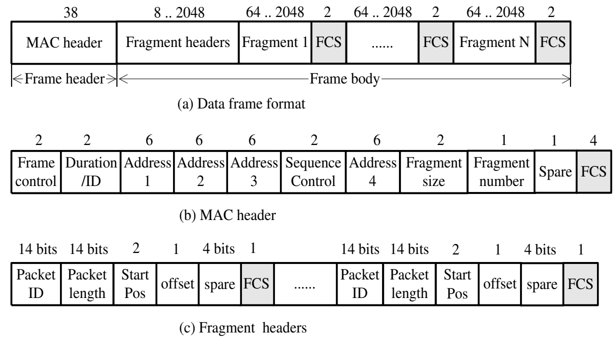


Fig. 2. Data format in the AFR scheme.

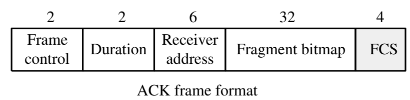


Fig. 3. ACK format in the AFR scheme.

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 (See Fig. 3).

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 $L_{p1} = 1025$ bytes and $L_{p2} = 40$ bytes. The frame length is $L_f = 2048$ bytes and the fragment length is $L_{frag} = 512$ bytes[4]. Then AFR divides pkt_1 and pkt_2 into 3 and 1 fragments respectively and put them into the sending queue. A frame with *fragment size* of 512 bytes and *fragment number* of 4 is constructed. The corresponding fragment headers are shown in Table. II. After receiving the frame, the receiver operates as shown in *Algorithm* 1 to recover the fragments.

B. Comments

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

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 can refer to [16], [34] and [19].

[4]To show that AFR can support arbitrary sizes of fragmentation, we do not restrict ourselves to the fragmentation algorithm introduced in Section V-D.

	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.

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 receiving queue according to the ACK bitmap.
  check the receiving queue and transfer all correctly received packets
  upwards, and remove them from the receiving queue.
else
  discard this frame and defer an EIFS before next transmission.
end if
```

3) Multiple destinations: Thus far, we have focussed only on aggregation between a single source-destination pair. This facilitates a clear understanding of the pros and cons of the aggregation itself. However, our frame format can be easily extended to support multiple destinations by inserting a destination address field in each fragment header. Adding another field to the fragment header will of course increase the transmission overhead, but this seems unavoidable and the approach proposed here carries only a small overhead compared to other solutions in the literature [17]. Specifically, [17] proposes the use of a *physical delimiter*, which is transmitted at 6Mbps. Transmitting the *delimiter* at 6Mbps leads to a constant $8\mu s$ duration. In addition, this *delimiter* technique requires extra zeros to be added at the PHY layer [28], see Section V-E for an example of this overhead. In AFR, both MAC and fragment headers are transmitted at the current data rate and so their duration decreases with increasing data-rate.

4) Multi-rate: In the current WLANs, a commonly used technique to resist channel noise is to lower the PHY rate after measuring a high packet (or bit) error rate, and when the channel state improves, the PHY layer increases its rate accordingly. There are two issues to be addressed if multi-rate is to be supported in AFR: (i) Should we change the frame size with the PHY rate? (ii) Should we support one-to-many aggregation where receivers have different channel states?

The first issue has been discussed in Section III-B.

For the second question, a simple extension of AFR is required. To do this, we combine packets for the same destination into a group, before which a sub-physical header is added for negotiating a suitable rate. As a consequence less efficiency is expected compared to one-to-one aggregation. Details analysis of one-to-many aggregation is however out of the scope of the present paper.

V. Theoretical Analysis

Building on previous modeling work [12], [38], [30], [27] and [14], in this section we develop a model and use it to analyse the saturation throughput and delay of the AFR scheme over a noisy channel.

A. Model

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

- *Idle duration* T_I: When all STAs are counting down, no station transmits a frame and we have
$$T_I = \sigma. \tag{9}$$
- *Success/Error duration* T_3: When a frame is successfully transmitted or it is corrupted due to channel noise[5], the slot duration is the sum of a frame, a SIFS and an ACK duration,
$$T_3 = T_{hdr}^{phy} + T_f + T_{ack}. \tag{10}$$
- *Collision duration* T_C: When two or more stations transmit at the same time a collision occurs. In this case the sender waits for an EIFS before the next transmission and so
$$T_C = T_{hdr}^{phy} + T_f + T_{EIFS}. \tag{11}$$

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

$$P_I = (1-\tau)^n, \tag{12}$$

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

and

$$P_C = 1 - P_I - P_3. \tag{14}$$

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

[5]Recall that in the AFR scheme we consider frames that are partially corrupted by channel noise as successful transmissions

assumes that a frame can be retransmitted infinite times, which is inconsistent with the 802.11 specification [1]. Wu et al. relax this assumption [38] and thus we use Equations (8) and (9) from [38] for greater accuracy.

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

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

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

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

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

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

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

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

$$\begin{aligned} E[L] &= \sum_{i=0}^{m} [\Delta \cdot (L_f - i \cdot L_{frag})] \\ &= L_f \cdot (1-p_e^{frag}). \end{aligned} \tag{18}$$

We thus have that

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

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

B. Improvements over DCF

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

Fig. 4(c) plots the throughput efficiency ($\frac{Throughput}{PHY\,Rate} \cdot 100\%$) of the DCF and AFR schemes as PHY rate is increased. It can be seen that whereas the DCF efficiency rapidly decreases with increasing PHY rate (falling from 42% at 54Mbs to less than 10% at 432Mbs) the AFR efficiency is approximately constant with increasing PHY rate as discussed above. Observe that the efficiency falls with increasing BER as expected, but that the efficiency remains relatively high even under noisy conditions, e.g., achieving approximately 70% throughput efficiency for a BER of 10^{-5} and 60% efficiency for a BER of 10^{-4} .

C. Maximum frame size

It can be seen in Fig. 4(b) that the AFR throughput asymptotically approaches a maximum value as frame size is increased. We can determine this asymptotic value analytically as follows. As the frame size $L_f \to \infty$, we have that (since T_3=T_C)

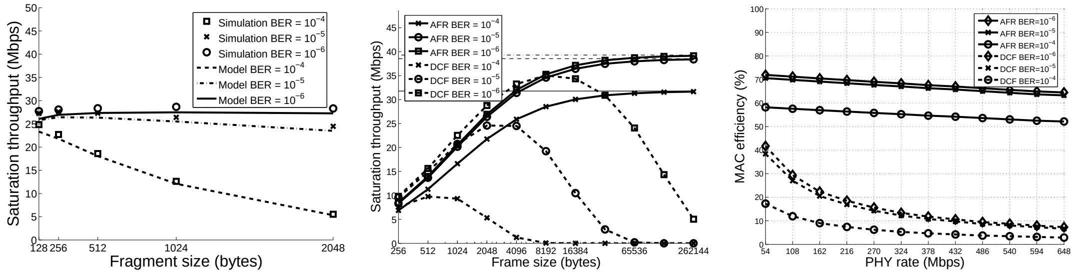
$$\begin{aligned} S_{AFR} &= \frac{P_3 \cdot (1-p_e^{frag})}{(P_3+P_C) \cdot T_3/L_f} \approx \frac{P_3 \cdot (1-p_e^{frag})}{(1-P_I) \cdot T_f/L_f} \\ &= \frac{P_3 \cdot (1-p_e^{frag})}{(1-P_I) \cdot \frac{((L_{frag}+L_{FCS}+L_{frag}^{hdr})/L_{frag})*8*Symbol}{Ndbps}}. \end{aligned} \tag{20}$$

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

In practice, of course, arbitrarily large frame sizes are often not feasible. The upper limit on frame size depends on the PHY's abilities and is also constrained by interface memory and the size of the STA's sending buffer. Fortunately, it can be seen in Fig. 4(b) that the gap between the maximum and actual throughput narrows rapidly with increasing frame sizes. Table 4(d) gives the loss in throughput (compared to the maximum achievable throughput) versus the frame size for a range of data-rates. If we consider operation at 90% or higher of the maximum achievable throughput to be our target, it can be seen that a maximum frame size of 32768 bytes is acceptable for data-rates up to 216 Mbps over a wide range of channel conditions while a maximum frame size of 65536 bytes is acceptable for data-rates up to 648 Mbps. We note that 65536 bytes is also the maximum size proposed in TGn's 802.11n proposal [4].

D. Optimal fragment size

Fragmentation plays a central role in aggregation schemes such as AFR, with fragments being the unit used for retransmission. When a very small fragment size is used, only corrupted bits are retransmitted but since each fragment has a fixed size header the overhead is relatively large. When a large fragment size is used, the overhead created by the fragment header is small but many bits will be unnecessarily retransmitted since a single damaged bit in a fragment will lead



(a) Model Validation (b) AFR vs DCF with increasing frame sizes (c) AFR vs DCF with increasing PHY rates

Rates	frame=32768	frame=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%

(d) Loss in throughput

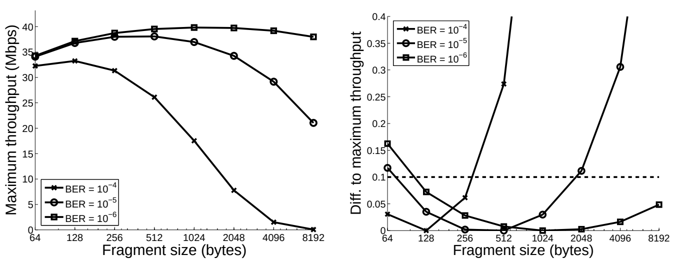
Fig. 4. (a) AFR: model vs. simulations. (b) The influence of frame size. (c) AFR vs. DCF with increasing PHY rate. (d) In the first column, the PHY rates are on the left of the slash, the basic rates are on the right. The unit of the rates is Mbps. The values in the second and the third columns are differences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

	Fig. 4(a)	Fig. 4(b)	Fig. 4(c)	Fig. 5	Fig. 7	Fig. 8
Number of STAs (n)	10	10	10	10	10	10
Application rate (Mbps)	54	54	=R	54	432	=R
Data rate (Mbps) (R)	54	54	varied	54	432	varied
Basic rate (Mbps)	6	6	=R	6	54	=R
AFR sending queue (packets)[a]	200	N/A	N/A	N/A	N/A	N/A
AFR IFQ (packets)[b]	200	N/A	N/A	N/A	N/A	N/A
Packet (bytes)	2048	$= L_f$	1024	$= L_{frag}$	2048	1024
Frame (bytes) (L_f)	2048	256, $\cdots$, 65536*4=262144	65536	8192	8192	varied
AFR fragment (bytes)(L_{frag})	128, $\cdots$, 2048	256	256	32, $\cdots$, 8192	256	256

TABLE III

THE PARAMETERS USED IN THE THEORETICAL MODEL AND ITS VALIDATION.

[a]AFR sending queue is the queue at MAC layer to 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.



(a) Optimal fragment size (b) Diff. to maximum throughput

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

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 MAXIMUM 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.

to the entire fragment being retransmitted. For a given BER there therefore exists an optimal fragment size that balances the tradeoff between fragment header overhead and excessive retransmission. Fig. 5(a) plots throughput versus fragment size from which the existence of an optimal fragment size that maximises throughput is evident. Observe that the optimal fragment size depends on the BER, as is to be expected (128, 512 and 1024 bytes for BER=$10^{-4}, 10^{-5}, 10^{-6}$ respectively).

In practice, we are interested in determining a simple scheme that approximates the optimal fragment sizes performance. It can be seen from Fig. 5(a) that the throughput peak is relatively flat and broad and thus we expect that the throughput reduction resulting from an approximate scheme can be kept relatively small. Fig. 5(b) plots the reduction in throughput, compared to that achieved with the optimal fragment sizes, of using a sub-optimal fragment size. From this plot we can see that if we can tolerate a throughput loss of up to 10%, then fragment sizes of 128 bytes and 256 bytes are near-optimal across a wide range of BERs. Corresponding data for a range of PHY rates are summarised in Table IV. It can be seen that fragment sizes of 128 and 256 bytes are always able to achieve within 10% of the maximum possible throughput. We have obtained similar results under a wide range of conditions including different numbers of stations, but these are not included here due to their similarity to the results in Table IV.

Based on these results, we propose a simple fragmentation algorithm: namely, for a packet P with a size of L_p, find the

m' which satisfies

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

where $m' = 1, 2, ..., 256$. 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 fall between 128 and 256 bytes. More importantly, the resulting sizes are almost the same. For example, a 257 byte packet is divided into one 128 byte and one 129 byte fragment, rather than one 256 byte and one 1 byte fragment.

E. Comparison with Similar Schemes

In this section, we compare the throughput performance of AFR with four other schemes proposed in the literature: Burst ACK ([32] [33] [36]), Block ACK ([3] [41]), Packet Concatenation (PAC) [21] and *Aggregation* [17].

These schemes can be classified into two categories: 1) Burst ACK and Block ACK; 2) PAC, *Aggregation* and AFR. The schemes in the first category transmit multiple frames at each transmission opportunity. The schemes in the second category transmit only one frame and use packet aggregation. AFR is the only scheme to use both fragmentation and aggregation. In the Burst ACK and Block ACK schemes, collisions lead to the whole Burst/Block being lost while errors lead to retransmission only of the corrupted packet. The PAC scheme is similar to our AFR scheme, except that before each packet in a frame there is a sub-physical-header, which is of a $12\mu s$ duration with an IEEE 802.11a PHY. The *Aggregation* scheme in [17] uses a special *delimiter* before each packet in a frame. As shown in [28], delimitation techniques need support from the PHY layer. In particular, zeros should be inserted to ensure the particularity of the *delimiter*. The number of zeros inserted depends on the sizes of the *delimiter* and the packet. For an 8-bit *delimiter* as in [17], $L_p/(2^{\varsigma+1}-2)$ zeros are required, where L_p is the packet size, and $\varsigma = 5$ [28].

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

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

Results are shown in Fig. 7. It can be seen that the schemes employing aggregation (the second category) consistently outperform the Burst and Block ACK schemes. It can also be seen that the PAC scheme has the lowest throughput amongst schemes in the second category. This is due to the long duration of the sub-physical-header. AFR achieves the highest throughput regardless of the number of stations.

F. Delay Analysis

Our model can be extended to estimate the MAC layer delay, i.e., the mean time between a packet reaching the

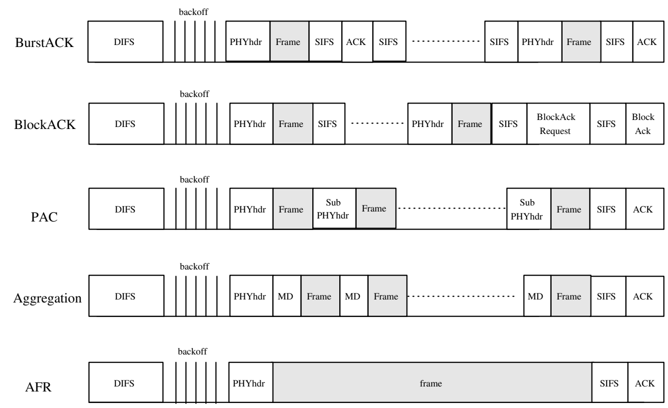


Fig. 6. Five schemes compared in this paper. 1) Burst ACK. 2) Block ACK. 3) Packet Concatenation from [21]. 4) Aggregation from [17]. 5) AFR.

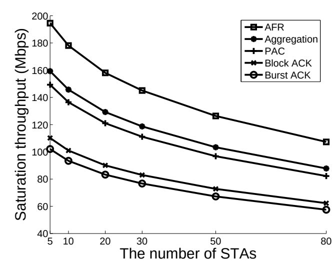


Fig. 7. Saturation throughput of the five schemes compared in this paper. The PHY data rate is 432 Mbps, basic rate is 54 Mbps. $BER = 10^{-5}$. The other parameters are listed in Fig. 1(c) and Table III.

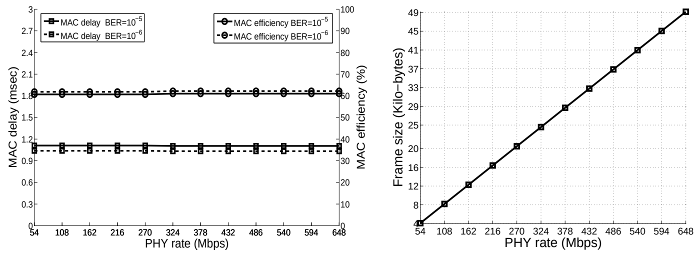
head of the MAC interface queue and being successfully transmitted. Let S^{frame} be the system throughput in frames-per-second rather than bits-per-second. That is, the MAC layer can transport S^{frame} frames in one second, thus the delay to successfully transmit one frame is $1/S^{frame}$, where

$$S^{frame} = \frac{E[number\ of\ frames]}{E[T]}. \tag{21}$$

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

$$\begin{aligned}&(1-p_e^{frag})+(p_e^{frag})(1-p_e^{frag})+\ldots+(p_e^{frag})^{r'-1}(1-p_e^{frag})\\&=1-(p_e^{frag})^{r'}.\end{aligned} \tag{22}$$

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



(a) Efficiency/delay vs PHY rate (b) Frame sizes

Fig. 8. Delay performance: In Fig. 8(a) we vary the frame sizes while increasing the PHY rates so that the throughput efficiency and MAC layer delay maintain roughly constant, and the corresponding frame sizes are shown in Fig. 8(b). The other parameters are listed in Fig. 1(c) and Table III.

attempts can be written as

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

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

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

For a fixed PHY rate, we expect the MAC delay to increase with frame size owing to the larger transmission time T_f for a frame. However, this is not the case when we choose the frame size to be a function of the PHY rate. In particular, by scaling the frame size in proportion to the PHY rate not only do we maintain MAC efficiency but we also maintain an approximately constant frame transmission time in which case the MAC delay is invariant with PHY rate. This is illustrated in Fig. 8(a), which plots the MAC delay with increasing PHY rate. The corresponding frame size as a function of PHY rate is shown in Fig. 8(b). Note that while the MAC efficiency and MAC delay are constant, the actual throughput increases from $54 * 60\% = 32$ Mbps to $648 * 60\% = 388.8$ Mbps.

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

VI. Simulations

As a complement to the theoretical analysis in Section V, we have implemented the AFR scheme in the network simulator *NS-2* [10], [11]. The network topology that we used is a peer-to-peer one where STA i sends packets to STA $i+1$. We report here the simulation results for three types of traffic (TCP, HDTV and VoIP), all of which follow the requirements of the 802.11n usage model [8]. See our technical report for other details about the simulation [26].

A. Metrics

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

- Throughput (= $c * L_p * 8/t$ Mbps): Throughput represents the maximum rate at which the MAC layer can forward packets from senders to receivers. Since in a WLAN, all the STAs share a common medium, this throughput is that achieved by the whole system rather than by a single STA.
- Peak delay (= $max\{d_1^{max}, d_2^{max}, \cdots, d_n^{max}\}$, where d_i^{max} denotes the maximum delay among all the packets successfully received by STA i): Peak delay is the maximum delay experienced by a successfully transmitted packet. This metric is used for HDTV.
- Percentage delay: The metric we use for VoIP is the percentage delay at the application level. It is defined as the percentage of packets whose delay is greater than a delay upper limit (e.g, at the application layer, the system should have less than 1% of packets whose delays are greater than 30 ms. This is the criterion proposed in IEEE 802.11n's requirement [8]). At the MAC layer, we use a similar threshold, i.e., less than 1% of packets may have delay greater than 15 ms.

B. TCP traffic

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

First, we evaluate AFR performance in a heavily-loaded WLAN with 50 STAs. Each STA performs a large FTP download, the data packet length is 984 bytes which yields an IP packet size of 1024 bytes when TCP and IP headers are added, TCP SACK functionality is used as this is prevalent in real networks. From Fig. 9(a) we can see that AFR achieves considerable throughput gains (by a factor of between 2 and 3 depending on channel conditions) over DCF. As discussed previously, AFR performance is relatively insensitive to the choice of fragment size in the range 128-256 bytes, although as might be expected the choice of fragment size becomes more important at higher BERs.

Second, we evaluate AFR performance as the number of STAs is varied from 10 to 80. Fig. 9(b) shows both the AFR and DCF throughput. AFR achieves between 2.5 and 3 times the throughput of DCF over this range of network conditions.

	Fig. 9	Fig. 10	Table VI
Number of STAs (n)	(a)50 (b)varied	varied	varied
Application rate (Mbps)	N/A	20	0.096
Data rate (Mbps) (R)	432	432	54
Basic rate (Mbps)	54	54	6
AFR sending queue (packets)[a]	10	10	10
AFR IFQ (packets)[b]	10	10	10
DCF IFQ (packets)[c]	20	20	20
Packet (bytes)	1024	1500	120
DCF frame (bytes)	1024	1500	120
AFR frame (bytes)	8192	9000	1200
AFR fragment (bytes)	(a)varied (b)512	750	120

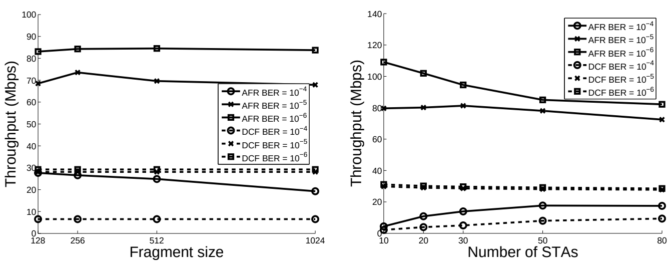
TABLE V

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

[a]AFR sending queue 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.



(a) Throughput vs fragment size (b) Throughput vs STAs

Fig. 9. Simulation results for TCP traffic. The parameters are listed in Fig. 1(c) and Table V.

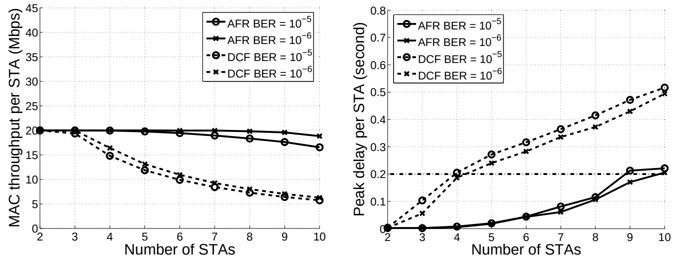
C. HDTV

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

We investigate AFR HDTV performance with a 432Mbps PHY data rate. Fig. 10 shows the throughput and delay performance of the AFR and DCF schemes as the number of STAs (and so HDTV flows) is varied. The peak delay constraint of 200ms is marked on Fig. 10(b). It can be seen that DCF can support only 2 simultaneous HDTV streams before the delay requirement is violated and the per flow throughput rapidly falls below the offered load. In contrast, AFR can support up to 9 and 10 streams for $BER = 10^{-5}$ and $BER = 10^{-6}$ respectively. That is, the HDTV capacity is increased by a factor of 5 over the legacy DCF. The overall network throughput achieved with 9 flows and $BER = 10^{-5}$ is 162Mbs.

D. VoIP

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



(a) Throughput (b) Peak delay

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

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

TABLE VI

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

We consider a WLAN with pure VoIP traffic. We use Brady's model [37] of VoIP traffic in which the mean ON and OFF periods are 1500 ms. Our performance requirement is to have less than 1% of packets with delays larger than 15 ms. Table VI shows the percentage of packets with delay exceeding 15 ms for a range of network conditions and numbers of voice calls. It can be seen that AFR's delay percentages are substantially less than the DCF's under all conditions, demonstrating the effectiveness of the AFR scheme even for traffic with very small packet sizes.

VII. SCOPE OF THE PAPER

In this paper, we restrict consideration to independent and identically distributed (i.i.d.) channel noise. Although we recognise that such a memory-less model is unable to capture fading characteristics in wireless channels, we comment that the PHY characteristics of IEEE 802.11n are still unknown at this time making the selection of a more accurate channel model problematic. We note that provided the channel coherence time is long enough to support large frame transmissions, it is relatively straightforward to modify our analysis to encompass more complex channels. Moreover, it can be argued that i.i.d. noise is in fact a worst case for aggregation schemes since in fading environments the bit errors tend to cluster together into bursts [15] (see also the measurement of the bit error distribution from an IEEE 802.11a test-bed [29]). An uneven error distribution typically benefits aggregation schemes since fewer retransmission are required compared to i.i.d. noise with the same mean BER [9]. For instance, if there are ten corrupted bits in one frame which contains ten fragments, and each fragment has exact one corrupted bit, then all the fragments have to be retransmitted. If all the ten corrupted bits occur in burst and gather into say five fragments, it is obvious that less retransmission is needed.

In this paper we focus on the fundamental issues affecting the performance of aggregation schemes in 802.11 WLANs. Thus several other techniques for further optimising CSMA/CA performance are not addressed here. These include optimization of the CSMA/CA contention window, which has been the subject of much attention in the literature, see [13], [42], [19], [31] and references therein for further details. Two-way aggregation is also possible, in which large frames piggyback in the ACK frames ([4], [24], and [39]).

VIII. Conclusion

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

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

IX. Acknowledgements

The first four authors would like to acknowledge the support of Science Foundation Ireland grant IN3/03/I346.

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] AFR Implementation, http://www.hamilton.ie/tianji_li/afr.html

[12] 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.

[13] 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.

[14] D. Malone, K. Duffy and D.J. Leith, "Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions", To appear in IEEE/ACM Transactions on Networking.

[15] R. Gallager, "Information Theory and Reliable Communication", John Wiley & Sons, 1968.

[16] G. R. Cantieni, Q. Ni, C. Barakat, and T. Turletti, "Performance Analysis under Finite Load and Improvements for Multirate 802.11", Elsevier Computer Communications Journal, Vol. 28, Issue 10, June 2005, pp. 1095-1109.

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

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

[19] 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.

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

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

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

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

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

[25] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "A New MAC Scheme for Very High-Speed WLANs". IEEE WOWMOM 2006.

[26] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "Aggregation with Fragment Retransmission for Very High-Speed WLANs". Technical Report, Hamilton Institute, NUIM, Ireland. 2006.

[27] 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.

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

[29] A. Miu, H. Balakrishnan, and C. E. Koksal, "Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks", Mobicom 2005.

[30] 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.

[31] Q. Ni, I. Aad, C. Barakat, and T. Turletti. "Modelling and Analysis of Slow CW Decrease for IEEE 802.11 WLAN", PIMRC 2003.

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

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

[34] G. Tan and J. Guttag, "Time-based Fairness Improves Performance in Multi-rate Wireless LANs", USENIX 2004.

[35] G. Tan and J. Guttag, “The 802.11 MAC Protocol Leads to Inefficient Equilibris”, INFOCOM 2005.
[36] J. Tourrilhes, “Packet Frame Grouping: Improving IP multimedia performance over CSMA/CA”, ICUPC 1998.
[37] 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.
[38] 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.
[39] Y. Xiao, “IEEE 802.11 Performance Enhancement via Concatenation and Piggyback Mechanisms”, IEEE Transactions on Wireless Communications, Vol. 4, No. 5, Sep. 2005, pp. 2182- 2192.
[40] Y. Xiao, “IEEE 802.11n: Enhancements for Higher Throughput in Wireless LANs”, IEEE Wireless Communications, Dec. 2005, pp. 82-91.
[41] 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.
[42] X. Yang and N. Vaidya, “A Wireless MAC Protocol Using Implicit Pipelining”, to appear in the IEEE Transactions on Mobile Computing.

PLACE PHOTO HERE

Tianji Li received the B.S. and M.S. degrees in computer science from JiLin and ZhongShan Universities, China, in 1998 and 2001, respectively, and the M.S. degree in networking and distributed computation from the University of Nice Sophia Antipolis, France, in 2004. Currently, he is working towards the Ph.D. degree at the Hamilton Institute, National University of Ireland at Maynooth, Ireland. From 2001 to 2003, he was a software engineer at the Beijing Research Institute of Huawei Technologies, China. His research interests are performance evaluation and optimization in wireless networks.

PLACE PHOTO HERE

Qiang Ni is currently a lecturer in the Electronic & Computer Engineering, School of Engineering and Design, Brunel University, UK. Prior to that, he was a Senior Research Scientist at the Hamilton Institute, National University of Ireland Maynooth, involved in a 5.6M Euros Science Foundation Ireland Wireless Project with Intel (SFI/03/IN3/I346). His research interests are wireless networking and mobile communications including wireless LANs, WiMaX, wireless PAN and 3G/4G mobile cellular networks. He has published over 40 refereed papers in international leading journals, book chapters, and conferences in the above fields. He worked with INRIA France as a Researcher for three years (2001-2004). He received his Ph.D. degree in Engineering in 1999 from HuaZhong University of Science and Technology (HUST), China and subsequently spent two years as a postdoctoral research fellow at Multimedia and Wireless Communication Research Laboratory, HUST, China. During the period of 2000-2001, he was a visiting researcher at the Wireless and Networking Group of Microsoft Research Asia Lab. He has conducted extensive research work on the IEEE 802.11 wireless MAC and PHY layer protocols design, analysis and standardization. Since 2002 he has been active as an IEEE 802.11 wireless standard working group Voting Member, and a contributor for the IEEE wireless standards. He is a member of IEEE and was a recipient of the Best Student Award from Cisco Networking Academy.

PLACE PHOTO HERE

David Malone received B.A.(mod), M.Sc. and Ph.D. degrees in mathematics from Trinity College Dublin. During his time as a postgraduate, he became a member of the FreeBSD development team. He is a research fellow at Hamilton Institute, NUI Maynooth, working on wireless networking. His interests include wavelets, mathematics of networks, IPv6 and systems administration. He is a co-author of O'Reilly's ”IPv6 Network Administration”.

PLACE PHOTO HERE

Douglas Leith graduated from the University of Glasgow in 1986 and was awarded his PhD, also from the University of Glasgow, in 1989. In 2001, Prof. Leith moved to the National University of Ireland, Maynooth to assume the position of SFI Principal Investigator and to establish the Hamilton Institute (www.hamilton.ie) of which he is Director. His current research interests include the analysis and design of network congestion control and distributed resource allocation in wireless networks.

PLACE PHOTO HERE

Yang Xiao worked at Micro Linear as an MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement work before he joined Department of Computer Science at The University of Memphis in 2002. He is the director of W4-Net Lab, and was with CEIA (Center for Information Assurance) at The University of Memphis. He is currently with Department of Computer Science at The University of Alabama. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He is an IEEE Senior Member. He currently serves as Editor-in-Chief for International Journal of Security and Networks (IJSN) and for International Journal of Sensor Networks (IJSNet). He serves as a referee/reviewer for many funding agencies, as well as a panelist for NSF and a member of Canada Foundation for Innovation (CFI)'s Telecommunications expert committee. He serves as TPC for more than 70 conferences such as INFOCOM, ICDCS, ICC, GLOBECOM, WCNC, etc. His research areas are wireless networks, mobile computing, and network security. He has published more than 140 papers in major journals (more than 40 in various IEEE Journals/magazines), refereed conference proceedings, book chapters related to these research areas.

PLACE PHOTO HERE

Thierry Turletti received the M.S. (1990) and the Ph.D. (1995) degrees in computer science both from the University of Nice - Sophia Antipolis, France. He has done his PhD studies in the RODEO group at INRIA Sophia Antipolis. During the year 1995-96, he was a postdoctoral fellow in the Telemedia, Networks and Systems group at LCS, MIT. He is currently a research scientist at the Planète group at INRIA Sophia Antipolis. His research interests include multimedia applications, congestion control and wireless networking. Dr. Turletti currently serves on the Editorial Board of the Wireless Communications and Mobile Computing (WCMC), Wireless Networks (WINET) and Advance on Multimedia (AM) journals.