CHAIN: Introducing Minimum Controlled Coordination into Random Access MAC

Zheng Zeng†, Yan Gao†, Kun Tan§ and P. R. Kumar†
†Department of Computer Science, University of Illinois at Urbana-Champaign, USA
§Microsoft Research Asia, Beijing, China
Email: {zzeng2,yangao, prkumar}@illinois.edu, kuntan@microsoft.com

***Abstract*—IEEE 802.11 DCF is the dominant protocol used in existing WLANs. However, the efficiency of DCF progressively degrades with the increase of contending clients in the network as well as the wireless link rate. To address this issue, in this paper, we present a distributed random media access protocol, named CHAIN, which significantly improves uplink performance of WLANs. CHAIN mainly uses overhearing to coordinate clients in a network, and thus introduces little control overhead. The key in CHAIN is a novel *piggyback transmission opportunity*. In CHAIN, clients maintain a precedence relation among one another, and a client can immediately transmit a new packet after it overhears a successful transmission of its predecessor, without going through the regular contending process. When the network load is low, CHAIN behaves similar to DCF; But when the network becomes congested, clients automatically start chains of transmissions to improve efficiency. CHAIN is derived from DCF and co-exists friendly with it. Moreover, it possesses all the advantages of the 802.11 DCF standard - simplicity, robustness, and scalability. We analytically prove the correctness and fairness of CHAIN. Our extensive simulations on J-SIM verify our analytical results, and demonstrate significant performance gain of CHAIN over DCF.**

I. Introduction

IEEE 802.11 Distributed Coordination Function (DCF) is a simple and robust MAC protocol that is dominantly used in WLANs nowadays. However, due to its uncoordinated and random access nature, DCF has considerable overhead coming from the the wasted airtime of random backoff as well as collisions. We define the *packet transmission airtime*, $T_s := T_{DIFS} + T_{packet} + T_{SIFS} + T_{ACK}$, where T_{DIFS} denotes the distributed inter-frame space, T_{packet} denotes the transmission time of a data packet, and T_{SIFS} is the short inter-frame space. The *overhead per successful medium access*, $T_{oh} = T_{idle} + T_{coll}$, where T_{idle} (T_{coll}) is the average backoff time (the average time cost for collisions) for each successful medium access. Then, the efficiency of DCF can be computed as follows,

$$\rho_{DCF} = \frac{T_s}{T_s + T_{oh}}. \tag{1}$$

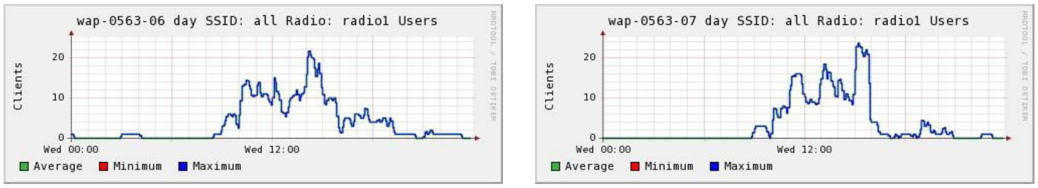
It is well known that DCF has low efficiency in two scenarios [1][2]. The first such scenario is a network with high contention due to a large number of active hosts, where T_{oh} becomes very high. The other scenario is when most of packets have a tiny transmission time (i.e., tiny T_s), thus resulting in low ρ_{DCF}. Unfortunately, both scenarios are quite common in existing WLANs. Firstly, with the popularity of 802.11 networks, wireless channels have become increasingly crowded. As an example, Figure 1 shows the number of clients associated with two APs over 24 hours in the campus WLAN of the University of Illinois, Urbana-Champaign (UIUC). These two APs (AP1 and AP2) are close to each other, and most of their clients are within the transmission range of one another. During the peak hours, each AP may serve over 20 clients, and there are more than 40 clients contending within the transmission range. Secondly, some widely used applications inherently involve small packets that use only small transmission time, *e.g.*, Voice Over IP. Further, with the deployment of new high-speed wireless technologies, like 802.11n, the transmission time of a packet is reduced proportionally. As a result, the MAC efficiency is further reduced [3].

For the downlink traffic, one possible solution is to perform centralized DCF-compatible scheduling at the AP. Several centralized scheduling schemes have been proposed in academia [4] [5] as well as in industry [6]. However, although scheduling schemes outperform DCF in many circumstance, they are difficult to apply to explicitly schedule *uplink* traffic (from client hosts to APs) due to the likely huge communication overhead and vulnerability to missing control messages.

In the literature, many distributed MAC protocols have also been proposed to improve the efficiency of DCF. They can be largely classified into three categories. The first category tries to find the optimal backoff schemes that minimize the overhead caused by backoff waiting and collisions. This category includes a large body of work, such as *Enhanced DCF* [7], *Idle Sense* [8], *MFS* [9]), *Implicit Pipelining* [10] and *CM-CSMA* [11]. However, the efficiency of these protocols still drops significantly when the packet airtime decreases. In the second category, a batch of packets, instead of one as in DCF, are transmitted after each successful medium access. Proposals in this category include *802.11e EDCF* [12], *FCR* [13], *OAR* [14] and *CM* [15]. Suppose that on average k packets get transmitted for each successful media access, then the MAC efficiency is

$$\rho = \frac{k \times T_s}{k \times T_s + T_{oh}} = \frac{T_s}{T_s + T_{oh}/k}. \tag{2}$$

This material is based upon work partially supported by NSF under Contracts CNS-0905397, CNS-1035378, CCF-0939370, CNS-1035340, USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, and AFOSR under Contract FA9550-09-0121.



AP1 in CS Dept. Building of UIUC   AP2 in CS Dept. Building of UIUC

Fig. 1. The trace of the number wireless clients associated to one access point over one 24 hour period.

Comparing (1) and (2), it is clear that higher efficiency is achieved by amortizing the contention overhead over k packets. However, with large batch transmission, the average channel access delay, d_c, increases as well [16] [17]. d_c is defined as the time interval between the two instants when a client host, or say, client, starts backoff and when it sucessfully delivers the packet. When the WLAN has m active clients and each client occupies the channel for a duration of t_o, we have $E[d_c] \propto mt_o$. A large access delay will adversary impact on many applications that have some QoS requirements. Thus, the usage of these approaches is limited. Besides, it does not resolve the contention. The third category is similar to the second one, but without increasing the average channel access delay. *GAMA-PS* [18] is a typical approach in this category. It maintains a "transmission group" consisting of all the hosts that have data to transmit. It divides the transmission channel into a sequence of cycles. For each cycle, the hosts in the "transmission group" transmit packets sequentially. Although *GAMA-PS* is efficient, it suffers from several drawbacks. First, it relies on explicit control packets to maintain the "transmission group" in every cycle. Thus, it is very sensitive to errors or losses on these control packets. Second, *GAMA-PS* requires tight control on all nodes in the network. So it cannot coexist with DCF or other similar protocols, posing a significant problem vis-a-vis deployment.

In this paper, we argue that a new MAC design should meet all of the following requirements. First, it should significantly improve the uplink channel access efficiency without increasing the channel access delay under any traffic pattern. Second, it should introduce negligible control and coordination overhead. Third, the overall scheme should be robust, yet simple and practical. Finally, it must be able to co-exist with DCF and other random access protocols. To the best of our knowledge, we are not aware of any existing design that addresses all of these issues.

This paper presents CHAIN, a Coordinated Heavy-traffic efficient Access scheme, which meets all of the aforementioned requirements. CHAIN adapts its behavior automatically to the dynamic traffic load. When the network is idle, it operates in a way similar to DCF. But when the network becomes congested, the clients automatically adapt their behavior to reduce the contention, without explicit control messages from AP or a scheduler. As a consequence, the network throughput is significantly improved. CHAIN is designed mainly to improve the efficiency for uplink traffic. By improving the uplink efficiency, it also indirectly improves the down-link throughput. We show that CHAIN co-exists fairly with DCF. It is also complementary to other existing designs to improve the down-link performance in WLAN.

Hereafter in the paper, we consider only uplink traffic in WLANs. The rest of the paper is organized as follows. Section II introduces the basic ideas and design details. We present an analytical model for CHAIN in Section III. A thorough simulation study is presented in Section IV. Finally, Section VI concludes the paper.

II. Protocol Design of CHAIN

This section first introduces the basic idea of CHAIN, and then describes CHAIN's design in details.

A. CHAIN basic design

In DCF, every host contends for its own medium access opportunity independently. We therefore address the following question: Can we build a loose bond among a group of hosts, so that when one host in the group gains a medium access opportunity, it can pass the opportunity to other hosts in the group without them having to go through contention? Thus, we arrive at the basic idea of CHAIN: *Each host has two types of transmission opportunities, namely spontaneous transmission and piggyback transmission. Spontaneous transmission* follows a modified DCF contention window-based media access scheme. Before sending a packet, the sender chooses a random number (in a different way from DCF) and starts transmission after waiting for that many idle slots. With *piggyback transmission*, each link l is given one (or multiple) *predecessor* links. After a successful transmission of its predecessor link (i.e., after overhearing an ACK of a packet sent on the predecessor link), instead of counting down its backoff timer, the sender of link l may immediately transmit after waiting an idle duration of SIFS. Since all other hosts should wait for at least a longer DIFS, the transmission on link l will not collide with other transmissions. Under heavy traffic, each host may always have a pending packet. Thus, one successful medium access will trigger a *chain* of collision-free piggyback transmissions from multiple hosts.

CHAIN has three major components: (1) A protocol to form and maintain the piggyback precedence relation among links; (2)The medium access protocol with piggyback transmissions; and (3) A modified exponential backoff scheme.

B. Piggyback precedence relation maintenance

CHAIN forms and maintains a *legal* Piggyback Precedence Relation Set (denoted as R) on link set L in a network. Let $l_{ij} \in L$ denote the directional link from host i to host j. Link l_{nk} can piggyback on to link l_{ij}'s transmission if and only if $(l_{nk}, l_{ij}) \in R$. If $(l_{nk}, l_{ij}) \in R$, we call l_{nk} as l_{ij}'s *follower*, and call l_{ij} as l_{nk}'s *predecessor*. In a general wireless network,

Algorithm 1 PIGGYBACK PRECEDENCE RELATION MAINTENANCE (PM)

$i \leftarrow 1$
while nonstop **do**
    $R_i \leftarrow \emptyset$
    Sort clients $X[1,\cdots,m]$ such that $Th_{i-1}(X[k]) > Th_{i-1}(X[k+1])$, $k = 1,\cdots,m-1$
    Choose j such that $Th_{i-1}(X[j]) > \delta_i > Th_{i-1}(X[j+1])$
    for $k = 1,\cdots,j$ **do**
        Add $(X[k\%j+1], X[k])$ to R_i
        Rate($X[k]$'s ACK) $\leftarrow$ min{Rate($X[k\%j+1]$),Rate($X[k]$)}
    end for
    for $k = j+1,\cdots,m$ **do**
        Add $(X[(k-j)\%(m-j)+j+1], X[k])$ to R_i
        Rate($X[k]$'s ACK) $\leftarrow$ min{Rate($X[(k-j)\%(m-j)+j+1]$),Rate($X[k]$)}
    end for
    Broadcast R_i
    $i \leftarrow i+1$
    Sleep for P
end while

a piggyback precedence relation R is *legal* when it satisfies the following two conditions:

- Condition (1): any two links $(l_{nk}, l_{ij}) \in R$ only if host n either can overhear and decode host i's data transmission while knowing the corresponding ACK time, or can overhear and decode the ACK sent from host j to i.
- Condition (2): any link can have at most one follower, or all its follower links have the same sender. (Otherwise, it is possible that two or more hosts piggyback on to the same transmission which results in an unavoidable collision).

Since only uplink traffic is of interest in this paper, an uplink can be simply referred to by its sender (i.e., the host). For brevity, in the rest of the paper, we say "client i R client j" instead of the longer version $l_{i\to AP}$ R $l_{j\to AP}$.

There may be many legal choices for the precedence relations on the set of uplinks, and we propose a simple protocol to generate such a relation R in a wireless LAN that may contains multiple APs. Our protocol lets each AP generate R locally without requiring a global conflict graph. This *Piggyback Precedence Relation Maintenance Algorithm (PM)* is shown in Algorithm 1. PM runs on each AP with m clients. R is updated periodically in every P seconds (we choose $P = 1$). R_i is the effective relation in the ith period, $Th_i(X[j])$ is the throughput (in unit of packets) of client $X[j]$ in the ith period, δ_i is a throughput threshold. At the beginning of each period, the AP classifies all associated clients as "active" node or "inactive" node by comparing the traffic generated by the client in last period, $Th_{i-1}(X[j])$, to the threshold δ_i. Then, all active clients and all inactive clients form two separated rings. The final relation R_i is the union of these two rings. The AP adjusts the modulation rate of ACK to ensure R_i satisfies condition (1). Note that this is necessary because there is no guarantee that any two clients satisfy condition (1) due to the employment of multi-rate transmissions. Clearly R_i satisfies condition (2). Therefore, we conclude that *PM* can always generate a legal R_i. Note that one subtle overhead here is that the AP may send some ACKs at a lower rate. But we believe this will not be a big issue. The relation assignment is broadcast to all clients every P seconds. If a client does not receive a update for P seconds, the old assignment is expired and the client will disable piggyback transmission until it receives a new assignment.

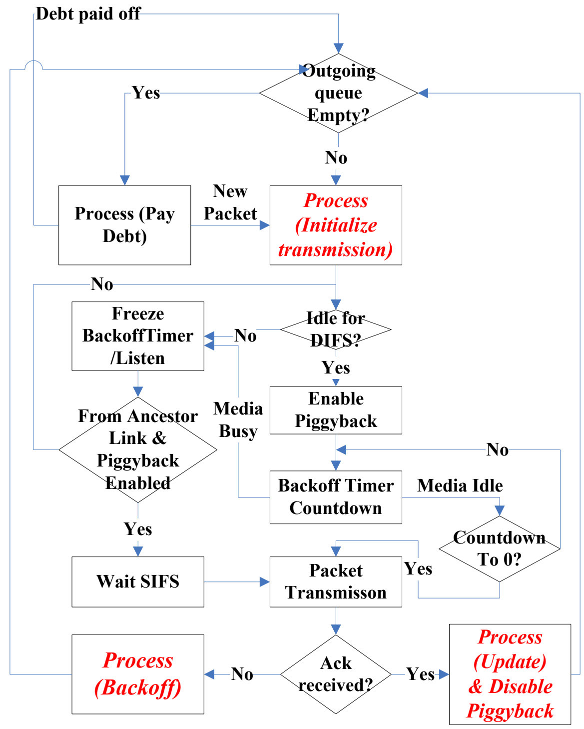


Fig. 2. The flow chart of the MAC protocol of CHAIN

C. MAC protocol

The detailed MAC protocol of CHAIN is shown in Figure 2. The key processes are highlighted in italics and their pseudo code is given in Algorithms 2-4, where BT denotes the backoff timer, IC (*idle slot count*) denotes the number of idle slots counted till a packet transmission, CW denotes the contention window size, β is defined as the ratio of piggyback transmissions over spontaneous transmissions, and λ is a constant chosen less than and close to 1. The CHAIN MAC protocol is derived from DCF, and has the following three major modifications.

Firstly, the new media piggyback access opportunity is added. As explained earlier, piggyback allows a host to transmit immediately after overhearing the ACK of its predecessor's packet, without going through the contention phase. The follower client starts its piggyback transmission after a SIFS and therefore it can grab the medium before other clients, who wait for a longer DIFS.

Secondly, an innovative *debt system* is added. This is done to restrict piggyback clients from taking too much airtime, and is the key feature that allows co-existence of CHAIN with DCF. A *debt* (denoted as D) is the media access opportunity a host owes to the system as a whole when piggyback happens, and it needs to paid back later just like a real debt. *Debt* plays an important role in CHAIN. We explain it using a simple example illustrated in Figure 3. Consider a WLAN with one AP (node C) and two clients (nodes A, B), where link $B \to C$ is a follower

Algorithm 2 Initialize Transmission

1: $r \leftarrow random(0,1]$
2: $BT_1 \leftarrow r \times CW$, $BT_2 \leftarrow \lambda \times r \times D$
3: $BT \leftarrow BT_1 + BT_2$

Algorithm 3 Update

1: $D \leftarrow \lambda \times D + BT_1 - IC$
2: update piggyback ratio β
3: update average contention window size $E[CW]$

of link $A \rightarrow C$. At time t_0, B has two packets in queue and A has one. Both of their debts are initialized to zero. Each node sets its BT to a random number in the range $[0, CW)$, and starts its packet transmission after observing BT idle slots. Obviously $IC \equiv BT$ in DCF. However, this is not true in CHAIN. Looking at Figure 3, at t_0, A chooses $BT_A = 3$ and B chooses $BT_B = 7$. After $DIFS+3$ idle slots, node A's backoff timer fires and it starts transmission. Node B overhears the ACK sent from C to A and starts transmission later after only waiting a SIFS duration; thus $IC_B = 3 < BT_B$. Compared with its behavior in DCF, node B has gotten to transmit its packet $BT_B - IC_B = 4$ slots earlier. We define these 4 slots to be the *debt* of node B (i.e., D_B). At t_1 B receives its ACK from C and initializes the transmission of its second packet with D_B=4. Now instead of choosing BT_B uniformly within range $[0, CW)$, BT_B is chosen uniformly within $[0, CW + D_B)$, as described in Algorithm 2. $BT_{B2} = 2$ can be viewed as the balance B plans to pay. At time t_3, D_B decreases from 4 to 2, as described in Algorithm 3. Then BT_B is set to 5; however, this time B piggybacks on to A's transmission again and owes the system $BT_{B1} - IC_B = 1$ more idle slot. Hence D_B increases to 3 at time t_4. Now B has no packet to transmit, and as the flow chart in Figure 2 suggests, it starts the process "pay debt": After observing the channel being idle for a period of DIFS, D_B is decremented as long as the channel is sensed idle. Finally at time t_5, D_B decreases to zero.

The $debt$ system serves two purposes. First, it prevents a piggyback client from taking too much medium time. This is the key reason why a WLAN running CHAIN can co-exist with a WLAN running DCF. Second, the "amortization" scheme helps to reduce the collision ratio under heavy traffic. We will elaborate more about this in Section III.

Finally, we add an additional rule to prevent a transmission chain from lasting forever. A chain of transmission automatically stops when the current sender has no follower, or the follower has no packet to transmit. Because clients of the same AP belong to a ring under piggyback precedence relation, once the chain of transmissions starts it may grab the medium forever under saturated traffic. We therefore restrict each client to only transmitting once in any chain of transmissions. We have two ways to enforce this rule. The first one is implicit. As shown in Figure 2, a client's piggyback right is disabled after a successful transmission and re-enabled only after it observes the channel being idle for over an interval of length DIFS. Recall that within a chain of transmissions the media cannot be idle for a DIFS duration. If this client can carrier-sense all other clients within the ring, then this scheme guarantees that it cannot transmit more than once in a single chain of transmissions. However, although very rare, clients associated with the same AP can be hidden terminals. In this case, we can use an explicit solution: Each AP keeps track of the chain of transmissions. When it observes that the chain is too long, the AP can always use one reserved bit in the MAC header of the ACK it sends out to forbid the follower to piggyback. Usually, however, the implicit solution is enough, while the explicit solution is always there for safety.

Algorithm 4 Exponential Backoff

1: $D \leftarrow \lambda \times D + \beta \times E[CW]$
2: $CW \leftarrow CW \times 2$
3: $E[CW] \leftarrow$ average contention window size

D. Modified exponential backoff

Algorithm 4 describes the backoff scheme in CHAIN. We can see that with an increasing β, a client increases its debt more aggressively. This is to ensure the fairness among multiple piggyback rings. A client with more piggyback transmissions will see less collisions. This is easy to understand because only the spontaneous transmissions are going through the regular contention process and may result in collisions, and thus exponentially enlarge the contention window. Therefore, clients with a larger piggyback ratio, β, will have an advantage in contention over clients with a smaller β. Algorithm 4 compensates this by making clients enlarge their debts proportionally to β when they detect a collision.

III. Analytical Evaluation

In this section we first provide an analytical model to evaluate CHAIN's performance under an ideal scenario, from which we prove rigorously that CHAIN has several nice properties. Then we relax the constraints of the ideal case and discuss CHAIN's behavior under realistic scenarios with co-existence of other WLANs, mobility, channel errors, and idle clients.

A. Analytical model

We study a single WLAN consisting of one AP and m clients. Each client has saturated uplink traffic. There are no errors on the transmission channel, and any packet losses are only caused by collisions among the m clients.

The CHAIN protocol coordinates the clients to form a piggyback ring of size m. The wireless media is divided into a sequence of transmission cycles by CHAIN. Each cycle begins with a contention/backoff period followed by a transmission period. The transmission period is either a successful delivery of m packets (one packet from each client), or a single packet collision. We use $T(n)$ to denote the nth transmission cycle, saying that $T(n)$ succeeds (fails) if its transmission period succeeds (fails). $I(n)$ is the number of idle slots of $T(n)$'s backoff period. $D_i(n)$, $B_i(n)$, $W_i(n)$ and $r_i(n) \in (0,1)$ are the debt, the backoff interval, the contention window size and the random variable to set the backoff interval of client i at the beginning of $T(n)$. Client i must wait $B_i(n)$ idle slots to start a spontaneous transmission during $T(n)$. According to the

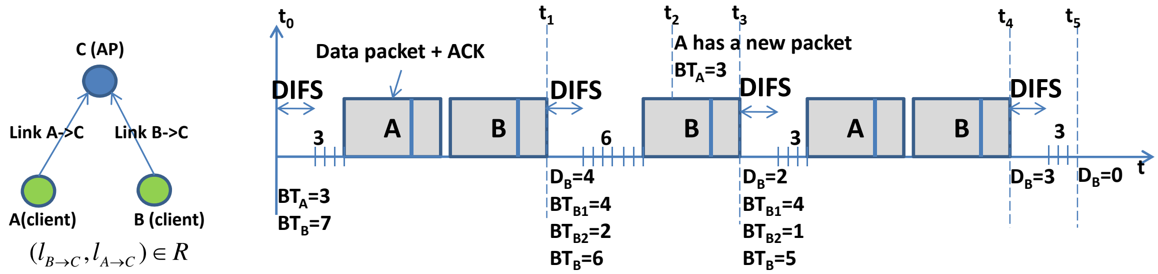


Fig. 3. A simple example illustrating how the debt system works in the CHAIN protocol

CHAIN MAC protocol, $D_i(n+1)$ and $B_i(n+1)$ are set at the end of each $T(n)$ for all clients. If $T(n)$ succeeds, then every client i sets

$$D_i(n+1) = B_i(n) + \lambda[1 - r_i(n)]D_i(n) - I(n), \quad (3)$$

$$W_i(n+1) = CW_{min}, \quad (4)$$

$$B_i(n+1) = \lambda r_i(n+1)D_i(n) + r_i(n+1)W_i(n), \quad (5)$$

where $I(n) = \min_{j=1}^{m} B_j(n)$. If $T(n)$ fails, then for all clients i that encounter this collision, set

$$D_i(n+1) = \lambda D_i(n) + (m-1)E[W_i], \quad (6)$$

$$W_i(n+1) = \max\{2W_i(n), CW_{max}\}, \quad (7)$$

$$B_i(n+1) = \lambda r_i(n+1)D_i(n) + r_i(n+1)W_i(n+1). \quad (8)$$

For other clients i which do not encounter this collision, their debts remain unchanged and their backoff interval decreases by $I(n)$, i.e.,

$$D_i(n+1) = D_i(n), \quad (9)$$

$$B_i(n+1) = B_i(n) - I(n). \quad (10)$$

Lemma 1: If $0 < \lambda < 1$, the debt of any client i is bounded, i.e., there exists a constant C such that

$$\limsup_{n\to\infty} D_i(n) < C. \quad (11)$$

Proof: At the beginning of each $T(n+1)$, $D_i(n+1)$ can be updated in three ways as defined in (3), (6) and (9). If using (3), by substituting $B_i(n)$ in (3) with the expression given in (5), it can be rewritten as

$$\begin{aligned} D_i(n+1) &= r_i(n)W_i(n) + \lambda D_i(n) - I(n) \\ &\leq \lambda D_i(n) + CW_{max}. \end{aligned} \quad (12)$$

If $D_i(n+1)$ is updated using (6) or (9), we have

$$D_i(n+1) \leq \lambda D_i(n) + (m-1)CW_{max}, \quad (13)$$

or

$$D_i(n+1) = D_i(n). \quad (14)$$

Combining (12), (13) and (14) yields

$$D_i(n+1) \leq \max\{\lambda D_i(n) + (m-1)CW_{max}, D_i(n)\}.$$

Because $\lambda < 1$ and $D_i(0) = 0$, it is obvious that for any (i, n), $D_i(n) \leq \frac{(m-1)CW_{max}}{1-\lambda}$. In any case,

$$\limsup_{n\to\infty} D_i(n) \leq \frac{(m-1)CW_{max}}{1-\lambda} = C. \quad (15)$$

■

Since $D_i(n)$ is bounded, $B_i(n)$ and $I(n)$ are bounded too. Therefore the expected values of $I(n)$, $D_i(n)$ and $B_i(n)$ exist, which we denote by $E[I]$, $E[D_i]$ and $E[B_i]$ respectively. Define $E[D] = \sum_{i=1}^{m} E[D_i]/m$, and $E[B] = \sum_{i=1}^{m} E[B_i]/m$.

Proposition 1: The average backoff interval $E[I]$ of every T, when $\lambda > 1 - 2/(m(E[W]+1)$, is bounded by

$$\frac{E[W]-2}{2} \leq E[I] \leq \frac{E[W]+3}{2}. \quad (16)$$

In order to model the system, we assume that for each new transmission attempt of client i, it chooses a backoff interval sampled from a geometric distribution with parameter p_i, where $p_i = 1/(E[B_i]+1)$, and

$$E[B_i] = \frac{E[W_i] + E[D_i] - 1}{2}. \quad (17)$$

This assumption was proposed and justified in [1], and widely adopted by other researchers ever since. Under this assumption, we replace (5) and (8) with

$$B_i(n) \sim Geom(p_i). \quad (18)$$

Note that the geometric distribution is memoryless; hence (10) can be replaced by (18) too. The reason is as follows. If $T(n)$ fails and client i does not encounter this collision, then we must have $B_i(n) \geq I(n)$, and

$$\begin{aligned} P[B_i(n+1) \geq k] &= P[B_i(n) \geq I(n) + k | B_i(n) \geq I(n)] \\ &= P[B_i(n) \geq k], \end{aligned} \quad (19)$$

i.e., $B_i(n+1)$ is also a geometric random variable.

Because $I(n) = \min_{j=1}^{m} B_j(n)$, it is straightforward that $I(n) \sim Geom(p')$, where

$$p' = 1 - \prod_{j=1}^{m}(1 - p_i). \quad (20)$$

The exact value of $E[I]$ depends on every $E[B_i]$ and is difficult to derive since it is possible that $E[B_i] \neq E[B_j]$ when $i \neq j$. However, according to [19], given a set of geometric random variables $\{X_i : i = 1, ..., m\}$ and a set of exponential random variables $\{Y_i : i = 1, ..., m\}$, $E[\min(X_i)]$ can be bounded by $E[\min(Y_i)] = Avg(E[Y_i])/m$ as long as $E[X_i] = E[Y_i]$ for $i = 1, ..., n$. Therefore, we have

$$\frac{E[B]}{m} \leq E[I] \leq \frac{E[B]}{m} + 1. \quad (21)$$

Now we only need $E[B]$ to obtain $E[I]$.

In order to derive $E[B]$, let us consider two successful transmission cycles $T(n-1)$ and $T(n+k)$, with $T(n+k)$ being the first successful transmission cycle after $T(n-1)$. Letting $N_c = k$ denote the number of failure cycles between $T(n-1)$ and $T(n+k)$, and N'_c denote the number of clients that encounter collisions between $T(n-1)$ and $T(n+k)$, statistically

$$E[N_c] = \frac{1 - \prod_{i=1}^{m}(1-p_i)}{\sum_{i=1}^{m} p_i \prod_{j \neq i}(1-p_j)} - 1, \quad (22)$$

$$E[N'_c] = \sum_{i=1}^{m} p_i \prod_{j \neq i}(1-p_j). \quad (23)$$

Note that if one client encounters two collisions, it counts as twice when calculating N'_c. Suppose the expected number of collisions that client i encounters between $T(n-1)$ and $T(n+k)$ is Nc'_i. Note that $\sum_{i=1}^{m} Nc'_i = E[N'_c]$. If collisions do not occur to client i, $D_i(n+k) = D_i(n)$; otherwise $D_i(n+k)$ is updated according to (6). Therefore at the beginning of $T(n+k)$ we have

$$\begin{aligned} E[D_i(n+k)] &= E[D_i(n)] - (1-\lambda)E[Nc'_i]E[D_i(n)] \\ &\quad +(m-1)E[Nc'_i]E[W_i]. \quad (24) \end{aligned}$$

This implies that

$$\begin{aligned} E[D(n+k)] &= E[D(n)] - (1-\lambda)\frac{E[N'_c]}{m}E[D(n)] \\ &\quad +(m-1)\frac{E[N'_c]}{m}E[W]. \quad (25) \end{aligned}$$

Then at the beginning of $T(n+k+1)$ we have

$$\begin{aligned} E[D_i(n+k+1)] &= E[B_i(n+k)] + \frac{\lambda}{2}E[D_i(n+k)] \\ &\quad -(1+E[N_c])E[I], \quad (26) \end{aligned}$$

which implies that

$$\begin{aligned} E[D(n+k+1)] &= E[B(n+k)] + \frac{\lambda}{2}E[D(n+k)] \\ &\quad -(1+E[N_c])E[I]. \quad (27) \end{aligned}$$

Because both $T(n)$ and $T(n+k+1)$ are cycles following successful cycles, when $n \to \infty$ and the system is stable, we should have $E[D(n+k+1)] = E[D(n)]$. Using this equality and substituting (16), (17) and (25) into (27) yields the two inequalities: (28) and (29). If we choose $\lambda = 1 - \frac{1}{m\theta}$, then we derive the lower bound of $E[D]$ from (28),

$$E[D] \geq \frac{(1+E[N'_c])}{(1+E[N_c]+\theta)}(m-1)(E[W]-2) + o(1), \quad (30)$$

and the upper bound of $E[D]$ from (29),

$$E[D] \leq \frac{(1+E[N'_c])}{(1+E[N_c]+\theta)}(m-1)E[W] + o(1). \quad (31)$$

From (22) and (23) we obtain approximations $E[N_c] \approx \frac{m}{2(E[B]+1)} + o(\frac{m}{(E[B]+1)})$ and $E[N'_c] \approx \frac{m}{(E[B]+1)} + o(\frac{m}{(E[B]+1)})$. If we choose small $\theta < \frac{m}{(E[B]+1)}$, then (30) yields

$$E[D] \geq (m-1)(E[W]-2) \quad (32)$$

which implies

$$E[B] \geq m(E[W]-2)/2. \quad (33)$$

(33) yields $E[N_c] \leq \frac{1}{(E[W]-2)} + o(\frac{1}{(E[W]-2)})$, and furthermore $\frac{(1+E[N'_c])}{(1+E[N_c]+\theta)} \leq \frac{E[W]-1}{E[W]-2}$. Then (31) becomes

$$E[D] \leq (m-1)(E[W]+1), \quad (34)$$

which implies

$$E[B] \leq m(E[W]+1)/2. \quad (35)$$

Finally we obtain (16) by substituting (33) and (35) into (21). Proposition 1 is thus proved.

Proposition 2: The probability that one transmission cycle fails remains bounded regardless of m. The packet loss ratio decreases as m increases.

Proof: The probability that one transmission cycle fails is $E[N_c]/(1+E[N_c])$. We have shown that $E[N_c]$ does not increase with m in the proof of Proposition 1. The packet loss ratio is $E[N'_c]/[m(1+E[N'_c])]$, which is proportional to $1/m$. ■

We have thus provided an analysis of CHAIN under the basic scenario. Next, in the remainder of this section, we study CHAIN's behavior under more realistic scenarios by relaxing assumptions made for the ideal case one by one.

B. More causes for packet losses

Now we relax the assumption that the intra-WLAN collision is the only cause of packet losses. A packet loss may be due to channel error or collisions with clients of other WLANs. Suppose that a $T(n)$ that does not encounter intra-WLAN collision has probability q to fail due to channel error or by colliding with a transmission from outside its own WLAN. Hence the values of N_c and N'_c that we derive in Section III-A no longer apply.

Proposition 3: Proposition 1 holds regardless of q.

Proof: We prove this by showing that $\frac{1+N'_c}{1+N_c}$ is independent of q. Denote by p_1 the proportion of T's that do not encounter intra-WLAN collisions, and by p_2 the proportion of packets that do not collide with packets from another client inside the WLAN. Then we have

$$N_c = \frac{1}{(1-p_1)(1-q)}, \quad N'_c = \frac{1}{(1-p_2)(1-q)}, \quad (36)$$

which yields

$$\frac{1+N'_c}{1+N_c} = \frac{1-p_2}{1-p_1}. \quad (37)$$

■

C. Existence of clients with unsaturated traffic

Now we relax the assumption that every client has saturated traffic. Recall that according to the CHAIN protocol design, after the transmission of client i, if its follower has no data packet to transmit, the chain of transmissions in this cycle stops. Therefore a client i with small traffic load staying in the piggyback relation ring where other members have saturated traffic, may reduce the number of packets delivered per cycle, which decreases the efficiency of CHAIN. Moreover the follower of this client may benefit less from CHAIN. The smaller the traffic load of client i, the worse the situation is. However, our piggyback precedence relation maintenance (PM) scheme

$$E[D] \geq \frac{E[W]+E[D]-1}{2} + \frac{\lambda}{2}(E[D] + \frac{(m-1)}{m}E[N_c]E[W]) - (1+E[N_c'])(\frac{E[W]+E[D]-1}{2m}+1) \tag{28}$$

$$E[D] \leq \frac{E[W]+E[D]-1}{2} + \frac{\lambda}{2}(E[D] + \frac{(m-1)}{m}E[N_c]E[W]) - (1+E[N_c'])\frac{E[W]+E[D]-1}{2m} \tag{29}$$

TABLE I
WLAN CONFIGURATION

Parameter	Value
SIFS	10μsec
DIFS	28μsec
Slot time	9μsec
Phy Preamble	16μsec
CW_{min}	16
CW_{max}	1024
Bit rate	54Mbps

as defined in Algorithm 1 only allows clients which are busy enough to join the same ring, it therefore prevents any severe efficiency drop in CHAIN due to relative idle clients. Another design detail targeted at this scenario is that in Algorithm 4 we use the client's piggyback ratio β instead of $m-1$ as given in (8), because when $\beta \neq m-1$, the client benefits as though it were in a piggyback ring of size $\beta+1$ where every member always has data to transmit.

IV. SIMULATION EVALUATION

In this section, we use the J-SIM [20] simulator to evaluate CHAIN. We first compare CHAIN with 802.11 DCF, Enhanced DCF with optimal CW, and 802.11e in a single-AP WLAN. The simulation results also verify the analytical results in Section III. Then, we look into the fairness issue when CHAIN co-exists with DCF. Later in the last two subsections, we study CHAIN's performance with various traffic patterns. Table I summarizes the configuration parameters used in our simulation, which follow the IEEE 802.11g standard [21]. Unless otherwise stated, the default packet size is 1400 Bytes, all clients always have packets to send, and there is no downlink traffic in our simulations. Throughout this section, we call the piggyback precedence relation built by the CHAIN protocol a ring (rings) for short.

A. Basic performance evaluation

We compare the performance of four MAC protocols in a single-AP WLAN with m clients: 802.11 DCF, CHAIN, Enhanced DCF using optimal contention window (CW), and 802.11e EDCF with TxOP=1.5ms. The value of m ranges from 10 to 50. Figures 4 and 5 compare the system throughput when the packet size is 400B and 1400B, respectively. We see that when packet size is 400 Bytes, CHAIN has $70\% \sim 112\%$ throughput gain over DCF, and $65\% \sim 76\%$ throughput gain over enhanced DCF, when n ranges from 10 to 50. When packet size is 1400 Bytes, the throughput gain is lower since the efficiency of DCF increases with packet size, but we still achieve at least 40% throughput gain over enhanced DCF. The throughput gain of CHAIN over EDCF increases with m, because EDCF does not reduce contention as CHAIN does. Probability p_c is defined as the probability that one medium access fails due to collision. Figure 6 indicates that CHAIN has much lower p_c. Figure 7 verifies Proposition 1 that we proved in Section III: $E[I]$ is bounded regardless of m. This property promises that CHAIN can co-exist with DCF fairly.

TABLE II
COMPARISON OF MULTIPLE-AP SCENARIOS

	DCF	1 Ring	2 Rings	4 Rings
Packet loss ratio	38.9%	1.1%	4.1%	10.3%
System throughput (Mbps)	21.65	32.13	36.39	38.08

B. Multiple-AP scenario

In this subsection, we consider a network with multiple APs but all within a single contention domain. There are twelve clients, and we test four cases: (1) clients use DCF and associate with one AP; (2) clients use CHAIN and associate with one AP, hence we have a ring of size twelve; (3) clients use CHAIN and associated with two APs, with each AP serving six clients; (4) clients use CHAIN and associate with four APs, with each AP serving 3 clients. Table II lists the system throughput and the packet loss ratio in the four cases. The system throughput decreases when we have more rings of smaller size. This is because of two reasons. First, the length of a transmission chain is limited by the size of the ring; hence the efficiency of CHAIN increases with the size of the ring. Second, CHAIN can keep intra-ring collision ratio low but cannot reduce inter-ring collision ratio. Nonetheless, the collision ratio when there are four rings is still much lower than DCF. This is easy to understand. In DCF the contention happens among all clients. Thus, there are 12 contending clients. In CHAIN, the contention happens at ring level and there are only 4 contending rings.

C. Fairness

We evaluate fairness of CHAIN in a network with 24 clients. Clients 1-4 are associated with AP1, clients 5-12 are associated with AP2, and clients 13-24 are associated with AP3. All the three APs runs CHAIN, and we have three rings of sizes 4, 8, and 12 respectively. The per-client throughput is : 1.35Mbps for clients 1-4, 1.51Mbps for clients 5-12, and 1.55Mbps for clients 13-24, respectively. We can see that CHAIN achieves fairness very well among rings with different sizes. The throughput of a client of AP1 is only 13% less than that of a client of AP3, even though their ring sizes differ by a factor of three. It is the binary backoff scheme that causes this slight unfairness. Recall that the medium access probability of one ring is inversely proportional to $E[CW]$. Clients from small rings are more likely to encounter collisions and backoff, than clients from large rings; hence have larger $E[CW]$ and a smaller share of the medium.

D. Co-existence with 802.11

Here we study CHAIN's behavior when it co-exists with DCF. We run the following simulation in a two AP network. AP1 has 10 clients (clients 11-20) running DCF, and AP2 has

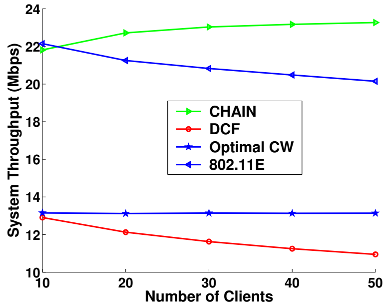


Fig. 4. System throughput (packet size=400B)

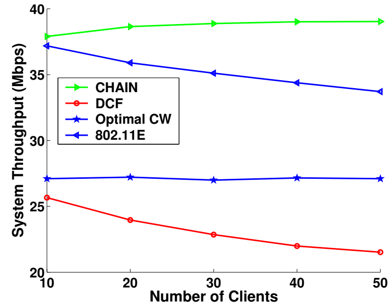


Fig. 5. System throughput (packet size=1400B)

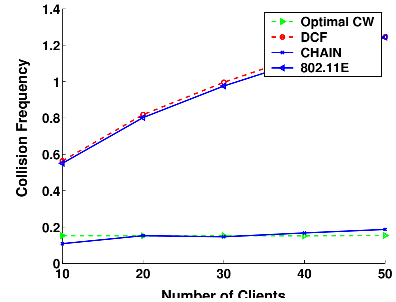


Fig. 6. Media access collision frequency

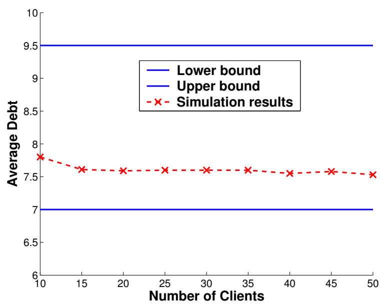


Fig. 7. The average number of idle slots before each media access

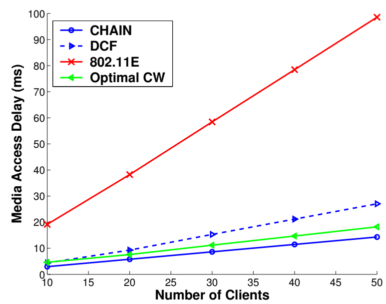


Fig. 8. Media access delay

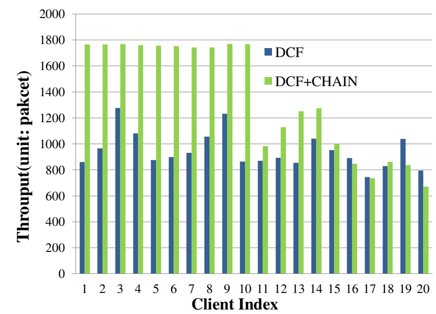


Fig. 9. The comparison of per-client throughput in pure DCF WLANs v.s. DCF+CHAIN WLANs

10 clients which run either DCF or CHAIN with a ring of size ten. Figure 9 compares the throughput of each client when AP2 chooses DCF and when it chooses CHAIN. It is obvious that when CHAIN competes with DCF, clients of AP2 achieve much higher throughput. However, when AP2 switches to CHAIN from DCF, the average throughput of clients in AP1 actually slightly increases by 7%. This is because when AP2 chooses CHAIN, its ten clients act as one contention entity instead of 10 entities; hence the collision probability is reduced and clients of AP1 benefit from it despite the fact that clients of AP2 grab more medium access opportunity. Therefore we conclude that a CHAIN system can co-exist with a DCF system without hurting them badly even under reasonably high contention level.

E. Clients with unsaturated traffic

In this section we study CHAIN's performance under a more realistic traffic pattern. The network in the simulation has one AP with twelve clients. Clients 1-6 have bulk traffic. Clients 7-12 have smaller packet arrival rate, and each has 120 packets to transmit per second. Their packet arrival is a Poisson process. The six of them together only require one fifth of the WLAN capacity. We compare the performance of three MAC protocols: DCF, *static* CHAIN (CHAIN using fixed piggyback precedence relation assignment instead of PM) and the *standard* CHAIN MAC. We generate two legal piggyback precedence relation: R_1 = (1,2,3,4,5,6,7,8,9,10,11,12), R_2 = (1,7,2,8,3,9,4,10,5,11,6,12). Then we run simulations using five different MAC settings: DCF, *static* CHAIN using R_1, *static* CHAIN using R_2, CHAIN with threshold δ_1 <120 packets/sec, CHAIN with threshold δ_2 > 120 packets/sec.

TABLE III

COMPARISON OF THE PERFORMANCE OF FIVE DIFFERENT MAC SETTINGS

MAC	Relative throughput	Delay (ms)	1-$\frac{\min(Th_1,...,Th_6)}{\max(Th_1,...,Th_6)}$
DCF	1	247.5	31.7%
CHAIN,R_1	1.23	45.9	23.2%
CHAIN,R_2	1.16	32.1	1.6%
CHAIN, δ_1	1.22	48.7	2.6%
CHAIN, δ_2	1.22	52.3	0.3%

Table III compares the average throughput of clients 1-6 and the queueing delays of packets of clients 7-12. We see that CHAIN always outperforms DCF. R_2 has less throughput gain than the other three since it is most likely for an idle client to break a chain of transmissions and make CHAIN less efficient. However, R_2 yields the least queueing delay since the chance for clients 7-12 to piggyback is higher in R_2. This property can be very helpful under certain circumstances, for example to support VoIP. If a VoIP user always piggybacks from a client who transmits often, its access latency can be significantly reduced. Another observation is that when R_1 is used, client 1 has larger throughput than client 6 since client 6 piggybacks to a less active predecessor. However, PM has handled such unfairness well by updating the piggyback precedence relation periodically.

F. The choice of throughput threshold δ

We evaluate the impact of setting threshold δ. We use the same network topology as in Section IV-E. Define Λ as the maximum system throughput when there is only one client with saturated traffic in the WLAN. The uplink traffic density of

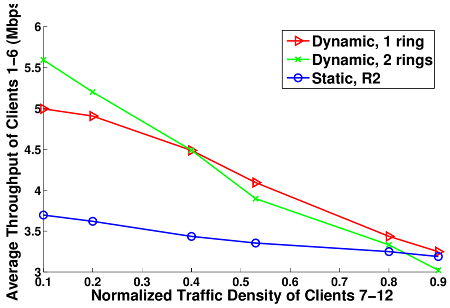


Fig. 10. The comparison of three different CHAIN settings

each client is measured quantitatively in unit of $\Lambda/12$. Through the simulation, the normalized traffic density of clients 1-6 is always 2, and that of clients 7-12, which are of the same value, varies between 0.1 and 0.9. We compare the performance of three MAC settings. The first one is static CHAIN using $R2$. The second one is CHAIN running PM with proper δ settings such that clients 7-12 always join the busy ring, hence there is only one ring. The third one is CHAIN running PM with clients 7-12 always forming a separate ring. Clients 7-12 are able to deliver all their packets under all of the three MAC settings. Figure 10 compares the average throughput of clients 1-6. Using $R2$ provides least throughput because $R2$ makes idle clients break the chain of transmission most often, and therefore is the worst ring formation, and hence one that we should avoid given that our goal is to maximize system throughput. It is clear that when clients 7-12 are less than 40% busy, clients 1-6 have larger throughput if clients 7-12 form a separate piggyback ring. However, when clients 7-12 are more than 40% busy, it is better for all clients to form a single ring. Therefore we suggest that δ be set around 40%. Note that whatever δ is used, CHAIN running PM outperforms static CHAIN using $R2$ significantly, when clients 7-12 are less than 80% busy. The reason is that even when there is only one ring, PM keeps busy clients together in the ring and avoids the formation of the "bad" ring(s) such as $R2$.

V. Discussion

We believe the key ideas of CHAIN, the overhearing and piggyback precedence relation, may be applied to other areas. For example, they can help solve exposed terminal problems. Suppose client A and B are exposed terminals, and A and B are associated with AP1 and AP2, respectively. With DCF, they cannot transmit at the same time due to mutual carrier sensing. With CHAIN, however, this issue can be resolved with the help of a third client C, whose transmissions can be overheard by both A and B. Now, AP1 and AP2 can assign both A and B to be C's follower. This way, A and B can transmit at the same time with piggyback transmissions.

Another direction that has not yet been covered in this paper is how different traffic patterns affect CHAIN's performance, as noted in Section IV-E. How to choose an optimal piggyback precedence with clients' traffic demands under consideration is an interesting next step.

VI. Conclusion

This paper presents CHAIN, a distributed random media access protocol, that significantly improves uplink performance of WLANs. CHAIN introduces a novel *piggyback transmission opportunity* by defining a precedence relation among clients. A client can immediately transmit a new packet after it overhears a successful transmission of its predecessor, without going through regular contending process. When the network load is low, CHAIN behaves similar to DCF; But when the network becomes congested, clients automatically start chains of transmissions to improve efficiency. Therefore, the overall contention overhead is significantly reduced. Based on overhearing, CHAIN is a light-weight protocol that adds little coordination overhead. It also co-exists friendly with DCF, making it possible to be incrementally deployed in existing WLANs. We analytically prove the correctness and fairness of CHAIN. Our extensive simulations on J-SIM verify our analytical results, and demonstrate significant performance gain of CHAIN over DCF.

References

[1] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11: design and performance evaluation of an adaptive backoff mechanism," *IEEE J. on Select. Areas in Commun.*, vol. 18, no. 9, September 2000.
[2] Y. Xiao and J. Rosdahl, "Throughput and delay limits of ieee 802.11," *IEEE Commun. Lett*, vol. 6, p. 355C357, 2002.
[3] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless lan," *SIGCOMM Comput. Commun. Rev.*, 2010.
[4] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "Centaur: realizing the full potential of centralized wlans through a hybrid data path," in *MobiCom*, 2009.
[5] P. Verkaik, Y. Agarwal, R. Gupta, and A. C. Snoeren, "Softspeak: making voip play well in existing 802.11 deployments," in *NSDI'09*.
[6] "http://www.merunetworks.com/."
[7] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. on Networking*, vol. 8, no. 6, December 2000.
[8] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, 2005.
[9] H. Kim and J. Hou, "Improving protocol capacity with model-based frame scheduling in ieee 802.11-operated wlans," in *Mobicom*, 2003.
[10] X. Yang and N. H. Vaidya, "A wireless mac protocol using implicit pipelining," *IEEE Trans. on Mobile Computing*, vol. 5, no. 3, 2006.
[11] Y. C. Tay, K. Jamieson, and H. Balakrishnan, "Collision-minimizing csma and its applications to wireless sensor networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 22, 2004.
[12] L. Romdhani and C. Bonnet, "Performance analysis and optimization of the 802.11e edca transmission opportunity (txop) mechanism," *WiMOB*, vol. 0, p. 68, 2007.
[13] Y. Kwon, Y. Fang, and H. Latchman, "Design of mac protocols with fast collision resolution for wireless local area networks," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 793–807, 2004.
[14] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. W. Knightly, "Oar: An opportunistic auto-rate media access protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 39–53, 2005.
[15] Y. Xiao, "Ieee 802.11 performance enhancement via concatenation and piggyback mechanisms," *IEEE Trans. on Wireless Comm.*, vol. 4, 2005.
[16] G. Sharma, A. Ganesh, and P. Key, "Performance analysis of contention based medium access control protocols," *IEEE Trans. Inf. Theor.*, vol. 55, no. 4, pp. 1665–1682, 2009.
[17] A. Grilo and M. Nunes, "Performance evaluation of IEEE 802.11 e," in *Proc. IEEE PIMRC*, 2002.
[18] E.-S. Jung and N. H. Vaidya, "An energy efficient mac protocol for wireless lans," in *INFOCOM*, 2002.
[19] G. C. Lawrence, L. M. Leemis, and D. Nicol, "On the minimum of independent geometrically distributed random variables," 1994.
[20] Jsim, "http://j-sim.cs.uiuc.edu/."
[21] "Ieee 802.11: Wireless lan medium access control (mac) and physical layer (phy) specifications," IEEE-SA, June 2007.