Wichita State University Libraries
SOAR: Shocker Open Access Repository

Mehmet Bayram Yildirim Industrial Engineering

A short-term fair MAC protocol for WLANs

Coskun Cetinkaya
coskun.cetinkaya@wichita.edu

Mehmet Bayram Yildirim
Wichita State University, bayram.yildirim@wichita.edu

Recommended citation
Cetinkaya, Coskun. and Mehmet Bayram Yildirim. 2007. A short-term fair MAC protocol for WLANs. *Ad Hoc Networks* 6(6), pp. 860–877

This paper is posted in Shocker Open Access Repository
http://soar.wichita.edu/dspace/handle/10057/3434

A short-term fair MAC protocol for WLANs ☆

Coskun Cetinkaya *, Mehmet B. Yildirim

Department of Electrical and Computer Engineering, Wichita State University, 1845 Fairmount, Wichita, KS 67260 0044, United States

Received 27 July 2006; received in revised form 14 May 2007; accepted 22 July 2007

Available online 1 August 2007

Abstract

Designing a medium access control (MAC) protocol that simultaneously provides high throughput and allows individual users to share limited spectrum resources fairly, especially in the short-term time horizon, is a challenging problem for wireless LANs. In this paper, we propose an efficient cooperative MAC protocol with very simple state information that considers only collisions, like the standard IEEE 802.11 MAC protocol. However, contrary to the IEEE 802.11 MAC, the cooperative MAC gives collided users priority to access the channel by assigning them shorter backoff counters and inter-frame-spaces than users who did not participate in the collision event. In other words, collided users are the only ones allowed to transmit in the following contention period. For the cooperative MAC protocol, we utilize an analytical throughput model to obtain the optimal parameter settings. Simulation results show that the cooperative MAC provides significant improvement in short-term fairness and access delay, while still providing high network throughput.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Wireless LANs; MAC protocol; Short term fairness; Protocol design and analysis

1. Introduction

In recent years, wireless local area networks (LANs) have received significant attention due to their attractive properties such as high throughput, easy deployability with or without any infrastructure, self-configurability, etc. Wireless LANs have limited spectrum, shared by many users. Hence, a MAC protocol should simultaneously provide high throughput and allow individual users to share the limited wireless spectrum fairly, especially in the short-term time horizon, while still requiring very simple state information. A short-term fair MAC protocol has the potential to improve the performance of real-time traffic by reducing delay jitter, and to improve the performance of TCP applications by smoothing the inter-arrival of ACKs and packets as well as fair access in any time horizon.

Time-division multiple access (TDMA) achieves short-term fairness and high throughput under heavy network loads [3,14], since any successful user must wait for other users to transmit before the next packet is sent. However, TDMA requires a central authority, synchronization among users, and

---

☆ Earlier version of this paper appeared in Proceedings of Med HOC 2004.

* Corresponding author. Tel.: +1 316 978 6314; fax: +1 316 978 5408.

E mail addresses: coskun.cetinkaya@wichita.edu (C. Cetinkaya), bayram.yildirim@wichita.edu (M.B. Yildirim).

knowledge of the number of users in the system. Further, the TDMA requires knowledge of the number of users which might be dynamically changing in wireless LANs. The IEEE 802.11 standard [1] defines a random access mechanism under Distributed Coordination Function (DCF) which does not require a central authority or synchronization. The IEEE 802.11 contention-based MAC protocol [1] requires very simple state information, since only collisions are considered in the backoff procedure. However, the IEEE 802.11 MAC protocol favors successful users, which results in short-term unfairness among them [2] by assigning larger backoff counters to the collided users.

Fair channel access can be provided by improving the fairness of the IEEE 802.11 MAC protocol (see [4–9,12,16–18,21] and references therein) by designing efficient access protocols or combining access protocols with scheduling algorithms. Nandagopal et al. [4] propose a distributed dynamic p-persistent MAC protocol to achieve proportional fairness by measuring idle time durations, collisions, etc. Vaidya et al. [6] adapt self-clocked fair queueing [19] to achieve fairness by piggybacking local virtual times and adjusting IEEE 802.11 backoff policies. Luo et al. [7] propose three localized fair queueing models, and users emulate start-time fair queueing [20] to achieve global weighted fairness in ad hoc networks. Finally, Kanodia et al. [5] emulate EDF-like reference scheduling to achieve the desired fairness level. All of the above schemes use some form of information sharing that requires additional signaling and packet overhead to allow users to cooperate with each other. Furthermore, they necessitate more computational complexity and memory requirements.

In this paper, first we develop a novel and efficient short-term fair cooperative MAC (C-MAC) protocol, which requires very simple state information since only collisions are considered in the backoff procedures, like the IEEE 802.11 MAC protocol. However, contrary to the IEEE 802.11 MAC protocol, the cooperative MAC assigns collided users shorter backoff counters and interframe-spaces to access the channel, unlike regular users who did not participate in the collision event, i.e., the collided users are the only ones allowed to transmit in the following contention period. To achieve short-term fairness, similar to TDMA, a successful user chooses a larger backoff counter to allow other users to transmit before the next packet transmission. This paper introduces a different access method and mathematical analysis than the earlier version [22].

Second, we analyze the short-term fairness property of the C-MAC by using the sliding- window size method [2] and Jain's fairness index [24]. The C-MAC achieves short-term fairness within sliding window sizes of two to three packets per user, whereas the IEEE 802.11 achieves the same level of fairness within sliding windows of 79–160 packets per user. As a result of the short-term fairness, the C-MAC achieves a significant reduction in access delay. Furthermore, increasing the number of users does not degrade the performance of the C-MAC, unlike the IEEE802.11.

Third, we derive an analytical throughput equation that depends on two system parameters and the average number of users in the system. We use an optimization toolbox to find the optimal setting for the C-MAC protocol, assuming that the average number of users is known. The average number of users can be estimated simply by monitoring the number of successful transmissions because of the short-term fairness property of the C-MAC protocol. When the average number of users dynamically changes and an estimation technique cannot be utilized, we determine the system parameters that will maximize throughput by using the minimum mean square error estimation.

Finally, we perform a large set of simulation experiments by using the ns-2 simulator [25] for Wireless LANs. We compare throughput of the C-MAC with both the IEEE 802.11 MAC and basic FCR MAC [10] protocols. The C-MAC achieves compatible throughput performance with the IEEE 802.11 when the RTS/CTS mechanism is applied, and achieves significantly better performance with the basic access mechanism. The FCR MAC outperforms the C-MAC at the expense of achieving short-term fairness at a sliding window size of 352 packets per user in the best case. We also investigate the performance of TCP applications under both the C-MAC and the IEEE 802.11, when we have different packet error rates. We find that the C-MAC handles packet errors much better than the IEEE 802.11, since the C-MAC transmits corrupted packets immediately, which results in up to 12% throughput gain. When we evaluate the C-MAC under different network load, heterogenous traffic, and variable capacity, we found that the C-MAC performs similar or better than the IEEE 802.11 MAC. We would like to note that this paper focuses on the design of a short-term fair protocol, and providing service differentiation and Quality of Service are out of the scope of this paper. We refer the interested reader to [29] for a service differentiation mechanism which uses the C-MAC protocol as an underlying MAC protocol.

The remainder of this paper is organized as follows. In Section 2, we describe the C-MAC protocol. Next, in Section 3, we describe the sliding window method and Jain's fairness index. We derive a throughput equation for the C-MAC protocol in Section 4. Then, in Section 5, we compare and contrast the IEEE802.11 MAC, FCR MAC, and C-MAC protocols, using the results of a simulation-based performance study. Finally, we provide conclusions in Section 6.

2. Cooperative MAC protocol

In this section, our goal is to design a cooperative MAC (C-MAC) protocol that provides short-term fair access to all users in wireless LANs via a contention-based MAC protocol with minimum computational complexity and memory requirements while providing high network throughput.

The C-MAC, similar to the IEEE 802.11 MAC, only considers collisions in the backoff procedure. However, the C-MAC introduces a two-level strict-priority mechanism by assigning different backoff counters and interframe-spaces (IFS) to each level. Contrary to the IEEE 802.11 MAC, collided users have higher priority to access the channel than regular users who did not participate in the collision event. Furthermore, the C-MAC, similar to the TDMA, assigns a larger backoff counter to a successful user to achieve short-term fairness. This paper introduces an improved access method over our earlier version [22], which requires each user to freeze its backoff counter when it detects a collision until the user detects two consecutive successful transmissions. Furthermore, the frozen user increases its counter by one for each collision it detects during the freezing mode.

Let us formally define the C-MAC protocol. When a user becomes active for the first time, it senses the channel to send a packet. If the user finds the channel to be idle for a duration of Distributed IFS (DIFS), the transmission will proceed. If the channel is busy, the user defers its transmission until the channel is idle for a time period of DIFS. Then, it generates a random backoff counter chosen uniformly from the discrete range $[W_s, 2W_s - 1]$, where W_s is the contention window size for regular users. After that, the user decreases its backoff counter by one for every idle duration of slot time. If the channel becomes busy, the counter is frozen until the channel is idle again for a duration of DIFS. The user will keep decreasing its backoff counter as long as the channel remains idle.

When the backoff counter reaches zero, the user transmits a request-to-send (RTS) message if the RTS/CTS mechanism is used. The receiver responds with a clear-to-send (CTS) message after a time period of Short IFS (SIFS). Any other users, who hear RTS/CTS packets, defer their transmissions and update their network allocation vectors (NAV). The sender responds to the CTS with a data packet and waits for an ACK packet. If the sender does not hear a CTS/ACK, it assumes that a collision has occurred and proceeds according to the procedure described below. If a successful transmission happens, the sender chooses a new backoff counter uniformly from the discrete range $[W_s, 2W_s - 1]$ in order to let the other users capture the channel. This will lead to short-term fairness since the new backoff counter is larger than any regular user's backoff counter regardless of the value of W_s as shown in Fig. 1.

If a user is involved in a collision, it chooses a new backoff counter from the discrete range $[0, W_c - 1]$, where W_c is the contention window size for the collided users. Then, the collided user starts decreasing its counter if the channel is idle for a duration of Priority IFS (PIFS) rather than a duration of DIFS. In our protocol, we set

$$\text{DIFS} = \text{PIFS} + W_c \cdot \text{SlotTime}, \tag{1}$$

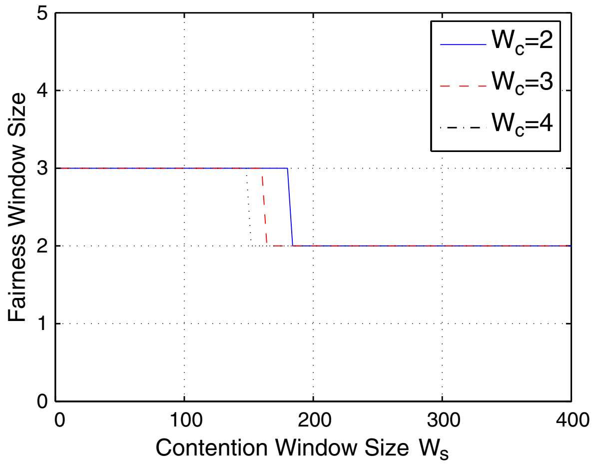


Fig. 1. Effects of contention window size on fairness index for 100 users.

in order to give every collided user a strict priority to access the channel over the regular users.

Collided users, who have higher priority over regular users, can encounter consecutive collisions among themselves. We need an additional mechanism to allow only the users that were involved in the latest collision to contend for the channel. The users that collided before, but were not involved in the latest collision, should defer their transmissions until the latest collision is resolved. However, they still should have a higher priority to transmit than the regular users. To satisfy this purpose, if a collided user identifies another collision before its transmission, it sets its backoff counter to zero and waits for the channel to be idle for a duration of DIFS like any regular user. Therefore, it would not contend for the channel until the latest collision is resolved. But it still has a higher priority over regular users, since the lowest backoff counter any regular user could have is one. Note that any user (except transmitting ones) who does not receive RTS/CTS correctly is able to sense that the medium is busy since the interference power received is sufficiently higher than the noise floor. As a result of this mechanism, when a collision occurs, the user identifies the collision and sets its NAV to the Extended IFS (EIFS) [1,10]. After a successful transmission, the collided user also chooses a new backoff from the discrete range $[W_s, 2W_s - 1]$ and needs to detect a DIFS amount of idle time to reduce its backoff counter.

In our C-MAC protocol, we set PIFS $= 30\,\mu s$ and SlotTime $= 20\,\mu s$. To illustrate the C-MAC protocol, we provide an example in Table 1. In this example, it is assumed that $W_s = 4$, $W_c = 3$, and the number of users is $M = 4$. Using Eq. (1), DIFS is equal to $90\,\mu s$ (in the IEEE 802.11 standard, DIFS $= 50\,\mu s$). Furthermore, the data rate is 1 Mbits/s, and packet size is 1 Kbyte. Hence, a collision takes $696\,\mu s$ of channel time, and a successful transmission takes $8416\,\mu s$ when the RTS/CTS mechanism is used. The users' current backoff counters (BCs) and the amount of required idle duration (RID) to transmit are given in the second row, and initially, no collided users are in the system. First, each user detects a DIFS amount of idle time. Then users 0, 1, and 2 transmit but collide with each other. User 3 does not transmit, since it needs an additional $20\,\mu s$, as shown in the third row. Next, users 0 to 2 choose new backoff counters from the range $[0, W_c - 1]$ (i.e., $[0, 2]$), which are 1, 1, and 2, respectively. After PIFS amount of time, users 0 to 2 reduce their backoff counters by one (1). User 3 cannot reduce its backoff counter, although there is an idle time of $50\,\mu s$, since it first needs to detect a DIFS amount of idle time. At $t = 0.83$, users 0 and 1 transmit but collide with each other. Since user 2 detects a collision before its transmission, it resets its backoff counter to zero, and needs to detect DIFS amount of idle time to transmit. In this way, user 2 defers its transmission to allow only users 0 and 1 to contend for the channel. However, user 2 still has a higher priority to access the channel than user 3, since user 3's backoff counter is one. Meanwhile, users 0 and 1 generate new backoff counters, which are 1 and 2, respectively. Then, user

Table 1
C MAC algorithm example

Time (ms)	User 0		User 1		User 2		User 3		Event
	BC	RID	BC	RID	BC	RID	BC	RID	
0	0	90	0	90	0	90	1	110	In progress
0.09	0	0	0	0	0	0	1	20	Collision of 1, 2, 3
0.78	*1*	*50*	*1*	*50*	*2*	*70*	1	110	In progress
0.83	*0*	*0*	*0*	*0*	*1*	*20*	1	60	Collision of 1, 2
1.53	*1*	*50*	*2*	*70*	0	90	1	110	In progress
1.58	*0*	*0*	*1*	*20*	0	40	1	60	Successful transmission by 0
9.99	4	170	*1*	*50*	0	90	1	110	In progress
10.04	4	120	*0*	*0*	0	40	1	60	Successful transmission by 1
18.46	4	170	6	210	0	90	1	110	In progress
18.55	4	80	6	120	0	0	1	20	Successful transmission by 2
26.97	4	170	6	210	5	190	1	110	In progress
27.08	3	60	5	100	4	80	0	0	Successful transmission by 3
35.49	3	150	5	190	4	170	5	190	In progress
35.64	0	0	2	40	1	20	2	40	Successful transmission by 0
44.06	7	230	2	130	1	110	2	130	In progress

0 transmits successfully, followed by user 1's transmission. Note that users 0 and 1 choose new backoff counters from the range $[W_s, 2W_s - 1]$ (i.e., [4, 7]), which are larger than the backoff counters of users 2 and 3, to achieve short-term fairness. Since there is no collided user left in the system, every user detects a DIFS amount of idle time. As a result, user 2 transmits successfully, and then user 3 transmits successfully.

3. Short-term fairness analysis

In this section, we describe the method used to analyze the short-term fairness of the C-MAC protocol using the sliding window method (SWM) via simulation. Koksal et al. [2] evaluate the short-term fairness property of the IEEE 802.11 MAC protocol by using SWM and the Markov chain with a reward method. They show that the IEEE 802.11 MAC protocol does not have short-term fairness but does possess the long-term fairness property. In addition, the Markov chain with a rewarding method encounters state-space explosion under dense networks. The SWM can be applied analytically by using an underlying stochastic process. However, in our setting, it is highly impractical due to its computational complexity. For the short-term fairness analysis of the C-MAC protocol, we will utilize SWM via simulation.

Short-term fairness depends on the length-of-time window over which the fairness analysis is performed. Let us consider the following sequence:

$\ldots AABBAABBAABB \ldots$

This sequence does not show short-term fairness on a two-packet time horizon, since 50% of the time we have two packets from the same user. However, when a four-packet time horizon is considered, the sequence possesses perfect fairness, since each user transmits an equal number of packets in every four-packet frame size.

In SWM, the channel accessions of users are traced and recorded. For a window of size w, the access time-fraction of each user is computed, and the window is then slid across the entire sequence of transmissions moving one element at a time. Each window is defined as a snapshot. Having T transmissions implies $(T - w + 1)$ snapshots. For the previous example, let w be a two-packet time window. Then, for the first window, we have 1 and 0 for the access-time fractions of users A and B, respectively. The second window has $\{0.5, 0.5\}$, and the third has $\{0, 1\}$. For each of these snapshots, we measure the per-snapshot fairness index. Koksal et al. [2] suggest two methods to calculate short-term fairness: Jain's fairness index [23,24] and the Kullback Leibler distance from information theory [26]. We use Jain's fairness index defined as follows:

$$F_j = \frac{\left(\sum_{i=1}^{M} \gamma_i\right)^2}{M \sum_{i=1}^{M} \gamma_i^2}, \tag{2}$$

where M is the number of users in the network and γ_i is the access-time fractions of node i at snapshot j.

The average of the per-snapshot indexes gives the fairness metric corresponding to a window size of w. Since short-term fairness depends on the length of the short-term window, this process is repeated for different window sizes. The actual short-term fairness characteristics of the protocol are observed by plotting the average fairness values versus the window sizes. Koksal et al. suggest that the protocol becomes fair if the average fairness index becomes more than 95%. We will use this criteria in our results, which are presented in here and Section 5.3.

Fig. 1 shows the fairness window size, where the average fairness index is more than 95%, for any pair of (W_c, W_s) when 100 users are present. As shown in Fig. 1, the C-MAC achieves short-term fairness at the fairness window size of three packets per user in the worst case for any given pairs of (W_c, W_s). This means that when you look at any $3 \cdot M$ consecutive successful transmissions, there are three packets from each user with probability 0.95. On the other hand, as shown in Fig. 3a, the network throughput heavily depends on the selection of (W_c, W_s) pair. In the next section, we will investigate the effect of (W_c, W_s) pair on the network throughput.

4. Throughput analysis and optimization

In this section, we analyze the throughput of the C-MAC protocol. Our goal is to find the best values of W_c and W_s to optimize throughput of the C-MAC protocol. Three components for throughput of contention-based MAC protocols include the following: successful transmissions, transmission failures, and idle slots due to backoff at each contention period (see Fig. 2).

Under heavy network traffic (i.e., each user always has a packet to transmit), we can express

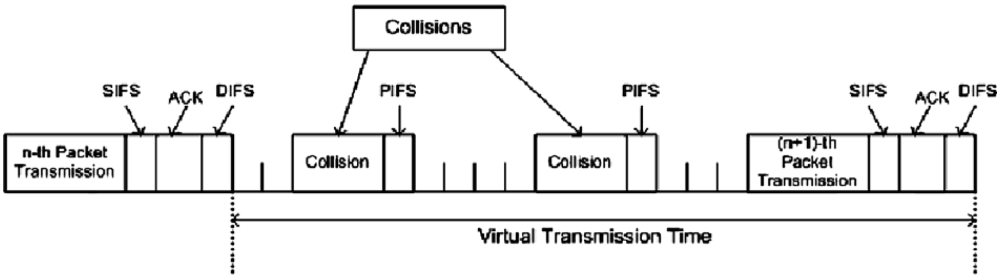


Fig. 2. Basic packet transmission structure of the C MAC.

the throughput of a contention-based MAC protocol for one successful transmission cycle as follows [11,13,15]:

$$\rho = \frac{m}{E[N_c] \cdot T_c + (E[T_i] \cdot \mathrm{st} + T_s + m + T_h)}. \qquad (3)$$

In Eq. (3), $E[N_c]$ is the average number of collisions, T_c is the time lost due to collision, $E[T_i]$ is the average idle time in each contention period, st is the length of a slot, T_s is the time spent in addition to packet transmission time during a successful transmission, and m and T_h are the times it takes to transmit an averaged-sized packet and the MAC header, respectively.

The C-MAC decouples the collision resolution and transmission attempt since it has a two-level strict-priority mechanism: collided users are in priority level 1, and regular users are in priority level 2. In Eq. (3), only $E[T_i]$ and $E[N_c]$ depend on the number of users (M) in the system and the transmission probability (p) of each user in priority level 2. Cali et al. [11] show that, assuming an equal contention window size for each user, a station transmits a packet with probability of $p = 1/(E[B]+1)$, where $E[B]$ is the average value of the backoff counter. Under the C-MAC protocol, each user in priority level 2 always has the same contention window size and chooses its backoff counter uniformly from the discrete range $[W_s, 2W_s - 1]$. Therefore, we can express the average value of the backoff counter as $E[B] = W_s + (W_s - 1)/2$. Hence, the packet transmission probability is equal to $p = 2/(3W_s + 1)$.

For this analysis, we assume that the number of users in the system M is known. In Section 5.1, we present a simple monitoring technique to estimate the number of users. Since each user transmits with probability p, the probability of a slot being idle (i.e., the probability that none of the users transmits in a slot when there is no user in priority level 1) is as follows:

$$P\{\text{idle}\} = P_i = \binom{M}{0}(1-p)^M = (1-p)^M, \qquad (4)$$

and the probability of a slot being busy is

$$P\{\text{busy}\} = P\{\text{transmission}\} = P_t = 1 - (1-p)^M. \qquad (5)$$

Under the C-MAC protocol, none of the users in priority level 2 can transmit as long as there is at least one user in priority level 1. Therefore, the average number of collisions ($E[N_c]$) in virtual transmission time depends on the number of users (M_t) who enter priority level 1, when no users are in priority level 1. We express the probability of k users transmitting ($P\{M_t = k\}$) under the condition that there is a transmission as follows:

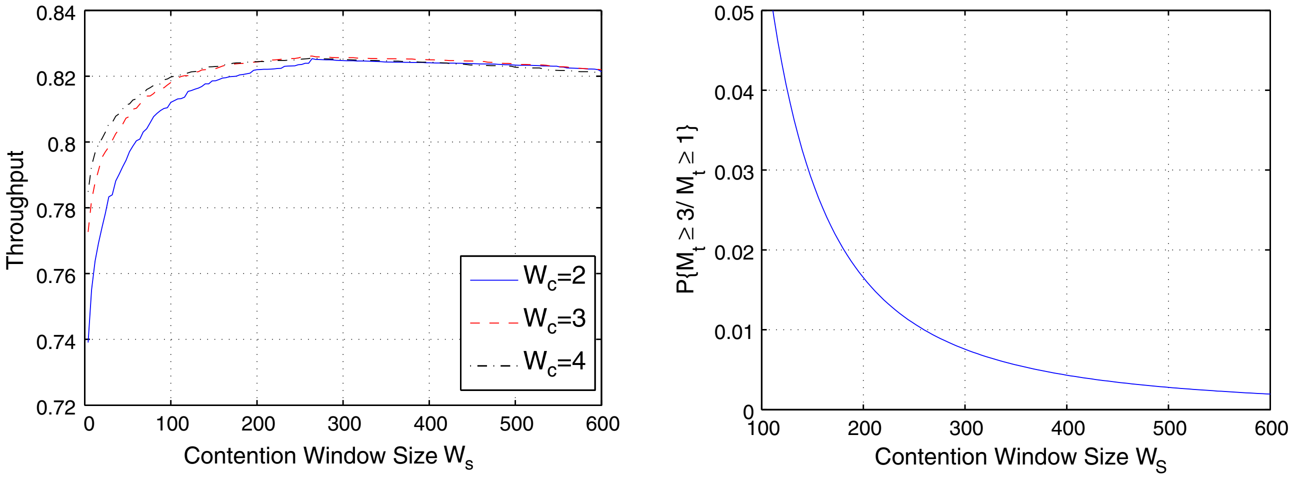
$$P\{M_t = k/M_t \geqslant 1\} = \binom{M}{k}\frac{p^k \cdot (1-p)^{M-k}}{1-(1-p)^M}. \qquad (6)$$

We can calculate $E[N_c]$ by using the total probability as

$$E[N_c] = \sum_{k=2}^{M} E[N_c/k] \cdot P\{M_t = k/M_t \geqslant 1\}, \qquad (7)$$

where $E[N_c/k]$ is the expected number of collisions when k users enter priority level 1. Although $E[N_c/k]$ can be calculated by using basic probability theory, because of the correlation among consecutive collisions, the derivation involves very complex calculations.

Before we calculate $E[N_c/k]$, let us investigate the effects of W_c and W_s on network throughput. In order to maximize network throughput, we need to minimize lost time due to both idle slots and collisions, as shown in Eq. (3). Fig. 3a shows network throughput under different values of W_c and W_s for the number of users $M = 100$ and 1 Kbyte packet size, when the RTS/CTS mechanism with DSSS specifications (shown in Table 2) is used. The throughput function of the C-MAC protocol has a



(a) Throughput

(b) Probability of 3 or More UsersTransmitting

Fig. 3. Effects of contention window size on throughput and the probability of 3 or more users transmitting for 100 users.

Table 2
System parameters for MAC protocols

Parameter	Values
SIFS	10 µs
DIFS	50 µs
SlotTime	20 µs
RTS	160 bits
CTS/ACK	112 bits
MAC header	224 bits
PHY header	192 bits
DataRate	1 Mbits/s

concave shape with increasing W_s values for every fixed W_c value. Cali et al. [11] similarly show that throughput of the IEEE 802.11 MAC protocol has a concave shape with increasing contention window sizes. Increasing the W_s value decreases the transmission probability of the individual user. As a result, the expected number of collisions $E[N_c]$ decreases, which initially improves throughput. On the other hand, increasing the W_s value results in a larger amount of idle time $E[T_i]$, which in turn results in throughput degradation for large values of W_s.

In wireless LANs, the amount of time that collisions consume is relatively higher than the time of an idle slot. For example, a collision consumes the amount of 35-idle-slot times when the data rate is 1 Mbits/s and the RTS/CTS mechanism is used (a collision, which includes RTS, SIFS, and CTS, takes 696 µs of channel time and SlotTime 20 µs). Therefore, we need to choose a W_s so large that the expected number of collisions $E[N_c]$ is minimized. When we increase the W_s, as shown in Fig. 3b, the probability of more than two users entering priority level 1 becomes very small. As a result, we can ignore their contribution to the expected number of collisions. Similarly, Cali et al. [11] show that the transmission probability p should be large enough to ensure that the number of users contending for the channel should be less than two, i.e., cases where more than three users contending for a channel can be ignored. As a result, we will assume that the number of users entering priority level 1 is at most two users. With this assumption, we express the expected number of collisions $E[N_c]$ as

$$E[N_c] \simeq E[N_c/k=2] \cdot Pr(M_t = 2/M_t \geqslant 1). \tag{8}$$

When two users enter priority level 1, they will collide initially. Then they will each choose a new backoff counter uniformly from the discrete range $[0, W_c - 1]$ after every collision. For every contention period, these two users collide with probability p_c $1/W_c$ and succeed with probability p_s $(W_c - 1)/W_c$. Therefore, the expected number of collisions including the initial one is

$$E[N_c/k=2] = 1 + \sum_k k \cdot P_c^k \cdot P_s = 1 + \frac{P_c}{P_s} = \frac{W_c}{(W_c-1)}. \tag{9}$$

Hence,

$$E[N_c] \simeq \frac{W_c}{(W_c-1)} \frac{M(M-1)}{2} \frac{p^2(1-p)^{M\ 2}}{1-(1-p)^M}. \tag{10}$$

We can divide the average idle time $E[T_i]$ into two components: idle time in priority level 1 ($E[T_{i1}]$) and in priority level 2 ($E[T_{i2}]$). Let us first calculate $E[T_{i2}]$. Any given slot k, there is no transmission with probability P_i [Eq. (4)] or a transmission with probability P_t [Eq. (5)]. Therefore, we can calculate $E[T_{i2}]$ by using geometric distribution as follows:

$$E[T_{i2}] = \sum_k k \cdot P_i^k \cdot P_t = \frac{P_i}{P_t} = \frac{(1-p)^M}{1-(1-p)^M}. \tag{11}$$

When using larger W_s, we assume that there are at most two users entering priority level 1. They choose a backoff counters from a discrete range $[0, W_c - 1]$. Therefore, we can express the average value of the backoff counter in priority level 1 as $E[B] \quad (W_c - 1)/2$. Hence, the packet transmission probability is equal to $p1 \quad 2/(W_c + 1)$. Any given slot k in priority level 1, there is no transmission with probability $P_{i1} \quad (1-p1)^2$ or a transmission with probability $P_{t1} \quad 1-(1-p1)^2$. Therefore, we can calculate $E[T_{i1}]$ per collision by using geometric distribution as follows:

$$E[T_{i1}] = \frac{P_{i1}}{P_{t1}} = \frac{(1-p1)^2}{1-(1-p1)^2}. \tag{12}$$

Under the C-MAC protocol, within two consecutive successful transmissions, users need to sense the channel to be idle for DIFS amount of time only one time when using large W_s. For the remaining transmission attempts, users sense only PIFS amount of idle time, as shown in Fig. 2. Therefore, we can express T_s and T_c under the RTS/CTS mechanism as follows:

$$T_s = \mathrm{PIFS} + W_c \cdot \mathrm{st} + \mathrm{rts} + \mathrm{cts} + \mathrm{ack} + 3 \cdot \mathrm{SIFS},$$
$$T_c = \mathrm{PIFS} + \mathrm{rts} + \mathrm{cts} + \mathrm{SIFS},$$

and under the basic access mechanism as follows:

$$T_s = \mathrm{PIFS} + W_c \cdot \mathrm{st} + \mathrm{rts} + \mathrm{cts} + \mathrm{ack} + 3 \cdot \mathrm{SIFS},$$
$$T_c = \mathrm{PIFS} + m + T_h + \mathrm{ack} + \mathrm{SIFS}.$$

Let us re-write Eq. (3) by using Eqs. (10)–(12), as follows:

$$\rho = \frac{m}{\frac{W_c}{(W_c \quad 1)} \frac{M(M \quad 1)}{2} \frac{p^2(1 \quad p)^{M \quad 2}}{1 \quad (1 \quad p)^M} \left[T_c + \frac{(1 \quad p1)^2}{1 \quad (1 \quad p1)^2} \mathrm{st}\right] + \frac{(1 \quad p)^M}{1 \quad (1 \quad p)^M} \cdot \mathrm{st} + W_c \cdot \mathrm{st} + T_r + m + T_h}, \tag{13}$$

where $T_r \quad \mathrm{PIFS} + \mathrm{rts} + \mathrm{cts} + \mathrm{ack} + 3 \cdot \mathrm{SIFS}$, i.e., $T_r \quad T_s - W_c \cdot \mathrm{st}$.

In Eq. (13), m is assumed to be a constant. As a result, maximizing throughput ρ is equivalent to minimizing the terms in the denominator of Eq. (13), which are a function of W_s and W_c. The following mathematical model is proposed to determine the optimal values of W_s and W_c:

$$\begin{aligned}
\min \quad & \frac{W_c}{W_c - 1} \frac{M(M-1)}{2} \frac{p^2(1-p)^{M \quad 2}}{1-(1-p)^M} \\
& \cdot \left[T_c + \frac{(1-p1)^2}{1-(1-p1)^2} \mathrm{st}\right] + \left[W_c + \frac{(1-p)^M}{1-(1-p)^M}\right] \mathrm{st}, \\
\text{s.t.} \quad & p = \frac{2}{3W_s + 1}, \\
& p1 = \frac{2}{W_c + 1}, \\
& W_s \geqslant \frac{2M+1}{3}, \\
& W_c \geqslant 2, \\
& W_c, W_s \text{ integer}.
\end{aligned} \tag{14}$$

In this formulation, the first constraint comes from the definition of transmission probability in priority level 2, and second constraint comes from the definition of transmission probability in priority level 1. To minimize collisions, the average number of users entering priority level 1 should always be less than or equal to 1, i.e., $M \cdot p \leqslant 1$. This is ensured by constraint $W_s \geqslant \frac{2M \quad 1}{3}$. The fourth constraint is required to eliminate collisions. Note that this mathematical model is a non-linear mixed-integer programming model. In the continuous version of this problem where W_s and W_c are nonnegative variables, the objective function has a global minima, since the second partial derivatives with respect to W_s and W_c are both positive.

The above mixed-integer non-linear model is solved using the GAMS/BARON solver [27]. BARON is a general purpose solver for non-linear optimization problems with integer variables. The detailed experimental results will be presented in Section 5.2.

5. Simulation results

In this section, we present simulation studies for the C-MAC protocol and compare it to the IEEE

802.11 and FCR MAC [10] protocols in the ns-2 simulator [25] for a Wireless LAN. We use DSSS specifications in our simulation experimentation (see Table 2). In the following subsections, we first present throughput results, then short-term fairness results. Finally, we modify the C-MAC protocol to coexist with the IEEE 802.11 MAC in a heterogenous environment. In the simulation experimentation, we vary the number of nodes M from 10 to 200, and the average packet sizes $m = \{0.25, 0.5, 1, 2\}$ Kbyte. We run ten simulations for each setting and report the average values of these simulations. Data rate is 1 Mbits/s for every simulation unless it is specified otherwise.

5.1. Estimation techniques

In this section, we present a simple technique to estimate the number of users and the average packet size. In the optimization model [in Eq. (14)], it is assumed that the number of users and the average packet size are known. Note that we need to know the average packet size in the case where we use basic access rather than the RTS/CTS mechanism.

Relying on the short-term fairness property of the C-MAC protocol, each user counts the number of successful transmissions (not the user IDs) between its two consecutive transmissions, and calculates the average of the past ten values in order to estimate the average number of users in the system. When each user always has a packet to transmit and there is no mobility, the estimation gives an exact result.

In the case of mobility and variable rates, in order to have uniform estimation in the wireless LAN, each user puts its estimated value into the packet header, which requires 1 byte packet overhead for the number of users up to 255 users or 2 byte for more users, and calculates the average number of users with this information. If computation and power consumption are not a big factor, we can use a more complex algorithm for the user estimation such as Kalman Filters in [28].

In order to estimate the average packet size, each user calculates its own average packet size and the wireless LAN-wide average packet size with the above technique. Then, the user puts these values into its own packet header, which requires 2 byte overhead for each calculated value. Note that the proposed estimation algorithm requires only 2 byte packet overhead due to estimation of the number of user for the RTS/CTS mechanism. In case of the basic access mechanism, it requires 6 byte packet overhead due to estimations of the number of users and packet size. We will refer to the described estimation algorithm as the user estimation in the remainder of this text. We would like to note that in case where Access Point (AP) is available, only AP estimates both the number of users and the average packet size, and then notifies all users.

We also propose a method if the estimation technique cannot be used due to power constraints or hidden terminals under ad hoc wireless LAN settings. In this case, by using the RTS/CTS mechanism, one can eliminate the effect of variable packet size since the optimal parameter values do not depend on average packet size. For this scenario, we obtain the constant values for W_c^* and W_s^* by using the minimum mean square error estimation (called mms estimation) for a variable number of users with the following equation:

$$[W_c^*, W_s^*] = \arg \min_{(W_c, W_s)} \sum_M (\rho_{\text{opt}}(M) - \rho(M, W_c, W_s))^2,$$

where $\rho_{\text{opt}}(M)$ is the optimal throughput value for given M and $\rho(M, W_c, W_s)$ is the throughput value for W_c and W_s in Eq. (13), where $p \quad 2/(3W_s + 1)$.

5.2. Throughput analysis

In this section, we first compare analytical throughput values with simulation results and investigate the effect of W_c on network throughput when the number of users is constant during simulations. Then, for the case where the number of users is dynamically changing, we compare analytical throughput with throughput values for the parameters obtained by user estimation and by minimum mean square error estimation, as well as throughput of the IEEE 802.11 MAC and throughput of the basic FCR MAC protocol [10], which has the same level of complexity as the C-MAC protocol under homogenous data traffic with heavy load. After that, we investigate the effects of heterogenous load and data traffic. Finally, we compare the performance of TCP under the C-MAC and IEEE 802.11 MAC protocols.

Table 3 shows analytical throughput (ρ_{opt}), simulation throughput (ρ_{sim}) values, and the optimal (W_c, W_s) for a different number of users, packet sizes, and access mechanisms when the number of users is constant. Fig. 4 shows network throughput under different values of W_c and W_s for $M \quad 100$

Table 3
The throughput of the C MAC protocol

m	$M = 10$			$M = 100$			$M = 200$		
	W_c, W_s	ρ_{opt} (%)	ρ_{sim} (%)	W_c, W_s	ρ_{opt} (%)	ρ_{sim} (%)	W_c, W_s	ρ_{opt} (%)	ρ_{sim} (%)
Basic									
0.25	4, 58	51.74	51.36	4, 603	51.52	51.19	4, 1209	51.51	51.12
0.5	4, 77	66.90	66.51	4, 803	66.66	66.30	4, 1609	66.65	66.26
1	5, 102	78.83	78.32	5, 1065	78.63	78.24	5, 2135	78.62	78.21
2	5, 142	87.01	86.67	5, 1486	86.82	86.37	5, 2978	86.81	86.21
rts/cts									
0.25	3, 30	54.39	53.96	3, 305	54.28	53.83	3, 610	54.27	53.81
0.5		70.46	70.06		70.36	69.99		70.36	69.95
1		82.72	82.46		82.65	82.33		82.65	82.31
2		90.51	90.01		90.47	89.88		90.47	89.82

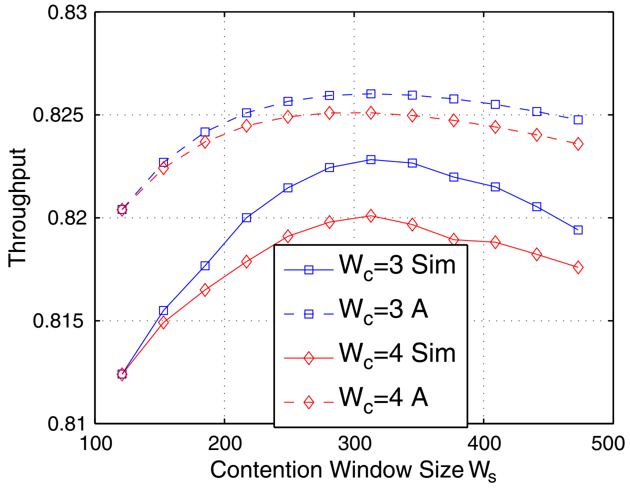
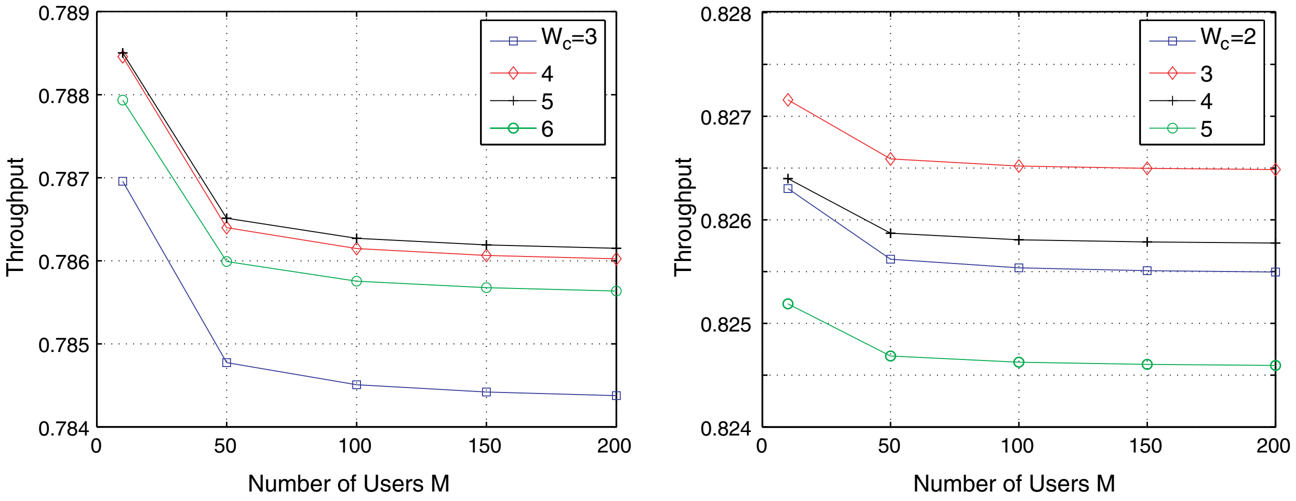


Fig. 4. Throughput of the C MAC under different contention window size W_s and W_c for $M = 100$.

and average packet size ($m = 1$ Kbyte) when the RTS/CTS mechanism is used. As we stated in Section 4, throughput of the C-MAC protocol depends on the (W_c, W_s) pair and has a concave shape with increasing W_s values for every fixed W_c value. When the W_s value is small, the average number of collided users is high. For example, let us assume W_c 4 and W_s is small such that there are 64 users collided. Since each chooses a backoff counter uniformly in the discrete range of [0,3], we assume that there will be a 16-node collision next. Then, these 16 nodes choose their backoff counters and the remaining 48 nodes defer their transmissions. We will see another collision of four nodes. Next, these four nodes will only content for the channel, and we can expect a successful transmission. As shown in the example, it will take several collisions before a successful transmission for the small W_s value. As a result, the throughput of the C-MAC protocol is low compare to optimal value. We would like to note that, the C-MAC still provides short-term fairness for small W_s values. Increasing the W_s value decreases the transmission probability of an individual user. As a result, the expected number of collisions decreases, and this improves throughput initially. On the other hand, increasing the W_s value further results in a larger amount of idle time since the transmission probabilities would be low, and users would not transmit even the channel is idle. Therefore, the C-MAC protocol encounters degradation for large values of W_s.

Fig. 4 shows that throughput values obtained by using simulation (marked as *Sim*) and optimization (marked as *A*) are very similar, i.e., we reach optimal throughput at the same pair of (W_c, W_s). Furthermore, the throughput difference between ρ_{opt} and ρ_{sim} is less than 1% in the worst case and gets smaller around optimal pair of (W_c, W_s). Thus, one can conclude that the proposed mathematical model captures the behavior of the C-MAC protocol.

Second, we investigate the effect of W_c on network throughput. Fig. 5a and b shows network throughput under different W_c values for the basic access and RTS/CTS mechanisms, respectively, given that the average packet size is 1 Kbyte and the number of user is constant. For any given W_c value, a W_s value maximizes throughput in Eq. (13). However, only one pair of (W_c, W_s) maximizes network throughput, as shown in Fig. 5. Simulation results show that the optimal W_c value depends only on the collision time (T_c), and this is independent of the number of users in the system. As shown in Table 3, for the RTS/CTS mechanism, W_c 3 maximizes throughput, regardless of packet size and number of users. For the basic access mechanism,



(a) Basic Access Mechanism

(b) RTS/CTS Mechanism

Fig. 5. Throughput for different W_c values.

when the average packet size increases (i.e., T_c increases), the optimal W_c value increases. For example, while W_c 4 optimizes throughput when the average packet size is less then or equal to 500 byte, the larger packet sizes, W_c 5 is optimal.

Next, we compare analytical throughput with throughput values for the parameter obtained by user estimation and minimum mean square error estimation, as well as throughput of the IEEE 802.11 MAC and throughput of the basic FCR MAC protocol [10], when the number of users is dynamically changing during simulations. For the simulations, it is assumed that the lifetime of each user is exponentially distributed, with mean time varying from 0.5 to 30 min. We also vary the inter-arrival time of individual users so that a different average number of users are in the system for different simulations. In Fig. 7a and b, ρ_{opt} represents the simulation results for optimal values of W_c and W_s, which are changed depending on the number of users at any given instant, i.e., representing the best result under the C-MAC protocol. ρ_{est} represents simulation results when we estimate the number of users and the average packet size (the user estimation), and ρ_{mms} represents results for the minimum mean square error estimation (i.e., W_s 440 and W_c 7 for the basic access with 1 Kbyte packet size, and W_s 97 and W_c 4 for the RTS/CTS). Fig. 7a and b shows the throughput comparison for the basic access and RTS/CTS mechanisms, respectively, when the average packet size is 1 Kbyte, and the mean life time is 5 min.

Let us first compare ρ_{est} with ρ_{opt}. The estimation algorithm encounters up to 15% throughput degradation when the lifetime of users is less than 3 min, since users are active for a short period of time resulting in errors on the estimation of both the average packet size and average number of users. However, for higher lifetime durations, the estimation algorithm provides compatible throughput with optimal values, i.e., it provides accurate estimated values. For example, the worst case throughput difference is 2% and 0.7% for the basic access and RTS/CTS mechanisms, respectively. In Fig. 6, we depicted the throughput of the C-MAC protocol for M 100 users in the system, while we vary the (W_c, W_s) pair assuming the estimated number of

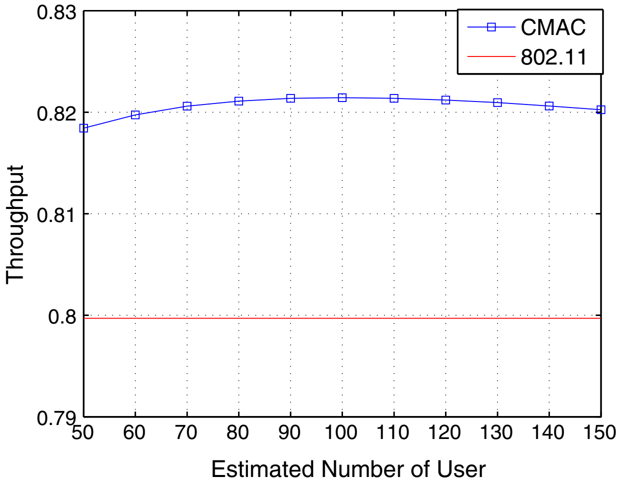
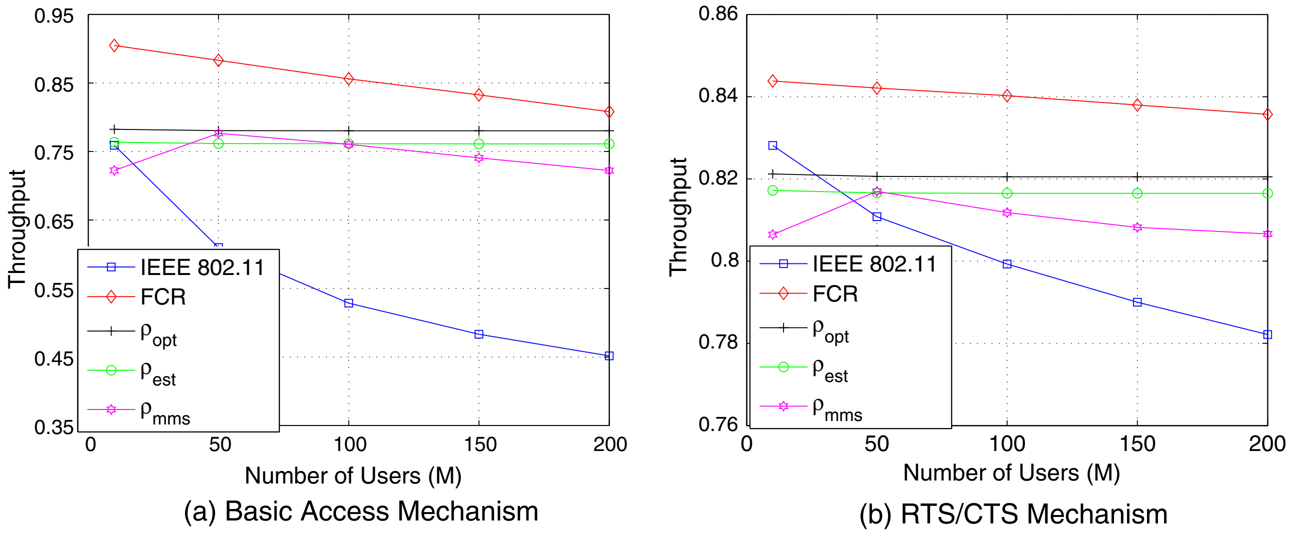


Fig. 6. The effect of estimation error.



(a) Basic Access Mechanism

(b) RTS/CTS Mechanism

Fig. 7. Throughput comparison for the mean life time 5 min.

users is 50–150 users. As shown in Fig. 6, the estimation error does not cause a significant performance degradation, and the C-MAC can provide high throughput even if the estimation error is 50%.

When we cannot utilize estimation technique, we can use mms estimation for parameter settings. Fig. 7a and b show that ρ_{mms} has a concave shape since W_c and W_s values are always constant for any given M. As a result, constant values produce lower throughput than ρ_{opt} for a small number of users due to a high number of idle slots (W_s for mms is bigger than for the optimal value), and for a high number of users due to high collision rates (W_s for mms is smaller than for the optimal value). However, the worst case difference is 5% and 2% for the basic access and RTS/CTS mechanisms, respectively.

The FCR MAC protocol [10] has very simple state information that considers only collisions, like the standard IEEE 802.11 MAC protocol and the C-MAC protocol. However, contrary to the C-MAC protocol, the FCR MAC favors the successful user by minimizing backoff counters. A successful user chooses a backoff counter from a discrete range $[0, 3]$ while other users choose their backoff counters from a discrete range $[0, 2047]$ in the worst case. As a result, as shown in Fig. 7, the FCR MAC protocol always provides the highest throughput. However, it encounters throughput degradation when the number of users increases, up to 10% under the basic access mechanism and up to 0.8% under the RTS/CTS mechanism. On the other hand, the C-MAC protocol provides steady throughput for both the optimal parameters and the user estimation parameters. When we compare throughput of the C-MAC with throughput of the IEEE 802.11 MAC, the IEEE 802.11 MAC protocol outperforms the C-MAC for 10 users under the RTS/CTS mechanism, 0.7% better than the optimal result and 2% better than ρ_{mms}. When the basic access mechanism is considered, the C-MAC has 3% more throughput for the optimal case, 0.5% more for ρ_{est}, but 2.6% less for ρ_{mms}. When the number of users is higher, the C-MAC always outperforms the IEEE 802.11 MAC. For 200 users, the C-MAC has 4% and 2.5% more throughput for the optimal and ρ_{mms} under the RTS/CTS mechanism, and 33% and 27% more throughput under the basic access.

Until now, we consider homogenous traffic type with heavy load, i.e. each user always has a packet to transmit. Let us first investigate the effect of variable load size in the network on throughput, mean and maximum packet delay. For this simulation, we assume that each node generates packets according to the poisson process. We set the number of users to 10 and 100 for two different settings, and we vary the total normalized traffic load from 0.1 to 1.5. Individual nodes generate traffic with a rate such that total normalized load is fixed. In Fig. 8, we depicted the throughput for the IEEE 802.11 MAC and the C-MAC with optimal and mms settings. As shown in Fig. 8, the throughput is linearly increasing with normalized load until load is 80% of the link capacity, i.e. we reach heavy load region. Then, it remain constant for both IEEE 802.11 MAC and the C-MAC. The performance relation of both protocols is the same as shown in Fig. 7.

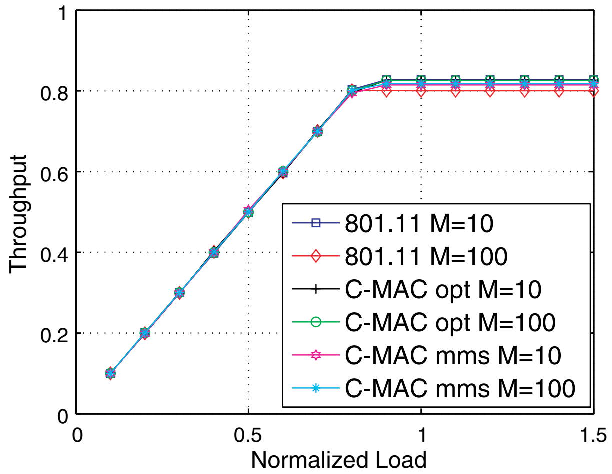
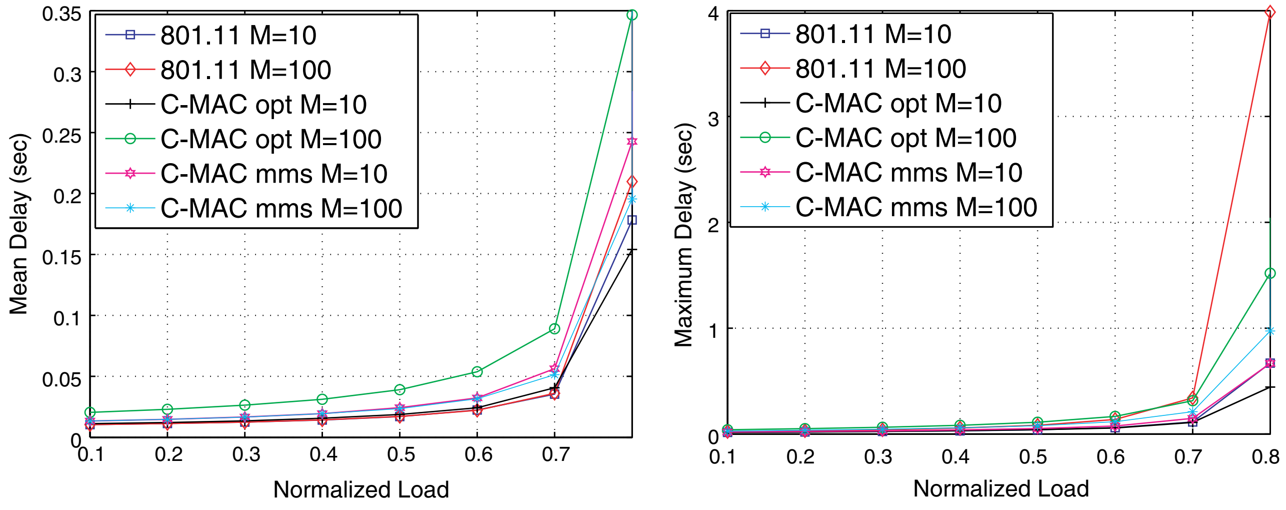


Fig. 8. The effect of variable traffic load.

When we compare the mean packet delay in Fig. 9a, the IEEE 802.11 MAC always provides a lower mean packet delay. For $M = 10$, the C-MAC with optimal setting performs similar to the IEEE 802.11 MAC, with a 15% larger delay in the worst case. When the network load increases the difference in delay get smaller, as low as 5%. However, the C-MAC with mms settings provides a higher mean packet delay than the IEEE 802.11 MAC due to a higher amount of idle slots. For example, for 10% data load, the mean packet delays are 10.5, 11.2, and 13.4 ms, while, for 80% data load, they are 178.4, 154.2, and 212.7 ms for the IEEE 802.11 MAC, the C-MAC with optimal setting, and the C-MAC with mms setting, respectively. For $M = 100$, for a lower data load, the mms setting gives better performance than the optimal setting while the optimal setting is better for high data load. For example, for 10% data load, the mean packet delays are 11.3, 18.5, and 13.6 ms, while, for 80% data load, they are 209.8, 195.5, and 346.5 ms for the IEEE 802.11 MAC, the C-MAC with optimal setting, and the C-MAC with mms setting, respectively.

When we compare the maximum packet delay in Fig. 9b, the IEEE 802.11 MAC provides a low value of the maximum packet delay for lower data load, while the C-MAC outperforms significantly better than the IEEE 802.11 MAC due to the short-term fairness property of the C-MAC protocol. For example, for 10% data load and $M = 10$, the maximum packet delays are 14.9, 16.0, and 19.3 ms, while, for 80% data load, they are 669.8, 444.1, and 667.3 ms for the IEEE 802.11 MAC, the C-MAC with optimal setting, and the C-MAC with mms setting, respectively. On the other hand, for 10% data load and $M = 100$, the maximum packet delays are 22.8, 39.5, and 27.6 ms, while, for 80% data load, they are 3.98, 0.97, and 1.52 s for the IEEE 802.11 MAC, the C-MAC with optimal setting, and the C-MAC with mms setting, respectively.

Next, we investigate the effect of heterogenous traffic types on the throughput when there are two traffic types with the packet size 160 byte and 1 Kbyte. We also assume that both traffic types always have a packet to transmit, and traffic is generated by the end users. We set the number of users in the system to a constant ($M = 50$) but vary the



(a) Mean Packet Delay

(b) Maximum Packet Delay

Fig. 9. The effect of variable traffic load on packet delay.

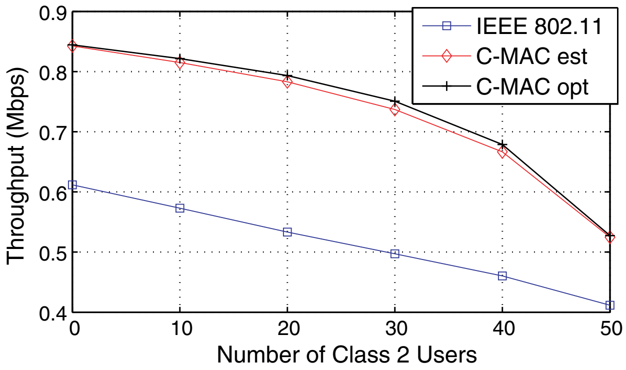


Fig. 10. The effect of heterogenous traffic type.

ratio between traffic types. In Fig. 10, we depicted the throughput of both the IEEE 802.11 MAC and the C-MAC with optimal and estimation setting when basic access mechanism is used. The *x*-axis corresponds the number of users with a 160 byte packet size. As shown in Fig. 10, the C-MAC outperforms the IEEE 802.11 MAC when heterogenous traffic types are present. We would like to note that we can further improve the performance of the C-MAC by including different traffic types and variable data load into mathematical model presented in Section 4.

In this paper, we focus on the design of a short-term fair protocol. However, when heterogenous traffic types with different data loads and performance objectives are present, the C-MAC itself cannot accommodate the needs of these traffic types. Providing service differentiation and Quality of Service is out of the scope of this paper. However, we present a solution for the service differentiation by using the C-MAC protocol as an underlying MAC protocol in [29] to which we refer the interested reader.

The IEEE 802.11a, 802.11b, and 802.11g provide multi-rate capability. We investigate the performance of the C-MAC protocol when different users can use different data rates. For this setting, we use the IEEE 802.11b standard setting with multiple rate capability (2, 5.5, and 11 Mbits/s rates). Holland et al. [31] presents a rate-adaptive MAC protocol (RBAR) to achieve throughput fairness while Sadeghi et al. [30] presents an opportunistic MAC protocol (OAR) to achieve temporal fairness under a multi-rate environment. Fig. 11 shows throughput of the IEEE 802.11 and the C-MAC when both protocols apply either the RBAR or the OAR. Under multi-rate environment, applying the RBAR or OAR as a multi-rate scheduler over the C-MAC protocol provides further throughput gain over the IEEE 802.11 MAC. For example, when the OAR is a scheduler, the C-MAC can provide up to 310 Kbit more throughput, while it is up to 250 Kbit more throughput for the RBAR scheduler.

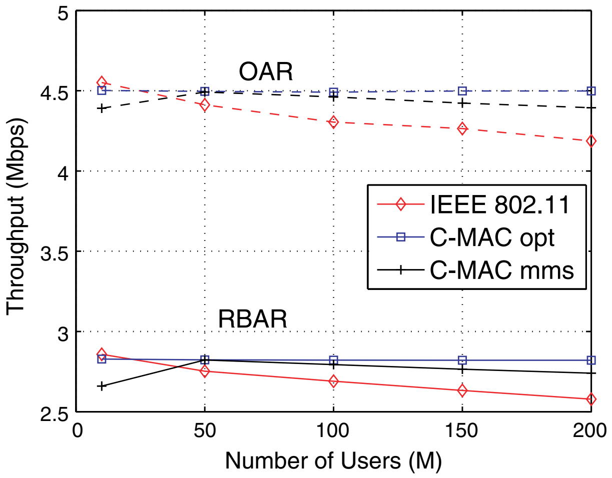


Fig. 11. The throughput under multi rate environment.

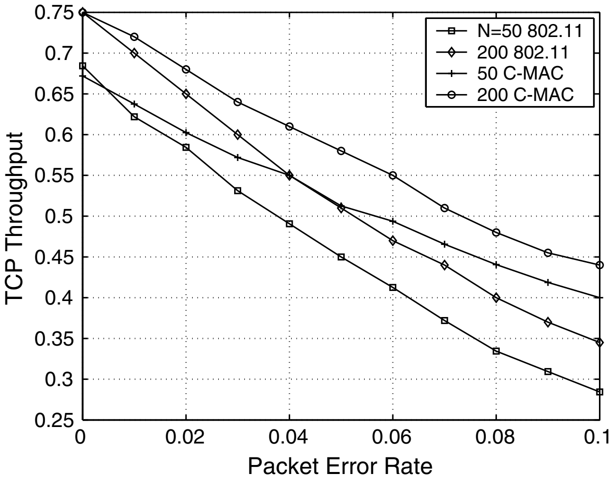


Fig. 12. TCP performance.

Finally, when the packet error rate of the transmitted packets is varied from 0.0% to 0.1%, the performance of TCP under both the C-MAC protocol with mms parameters and the IEEE 802.11 MAC protocol is compared in Fig. 12. When there is no packet error, TCP performs very similarly under both MAC protocols. However, when the packet error rate increases, the C-MAC protocol provides up to a 12% higher throughput than the IEEE 802.11 protocol, since the C-MAC transmits the corrupted packets immediately as a result of not getting an ACK back.

5.3. Short-term fairness analysis

In this section, we compare the short-term fairness of the C-MAC, FCR MAC, and IEEE 802.11 MAC protocols using the sliding window method presented in Section 3, assuming that the number of users is constant during simulations. We choose a sliding window size proportional to the number of users and vary the window size from 1 to 200. Note that Koksal et al. [2] assume that a MAC protocol is fair when the average fairness index passes 95%.

Fig. 13 shows the fairness of the IEEE 802.11, FCR MAC, and C-MAC protocols with optimal parameters and mms estimation parameters for different sliding window sizes of w/N; the x-axis shows the packets-per-user sliding window size. The IEEE 802.11 MAC becomes fair after a sliding window size of 79–160 packets per user, depending on the number of users; but increasing the number of users does not always cause fairness degradation, as shown in [2]. On the other hand, the C-MAC protocol reaches 95% fairness within sliding window sizes of two to three packets per user, depending on the number of users. This means that for any transmission sequence of $3 \cdot M$ consecutive successful transmissions, each user transmits an equal number of packets with probability 0.95.

The FCR MAC reaches 95% fairness at a sliding window size of 352 packets per user in the best case scenario, since it significantly favors the successful user. We would like to note that the C-MAC protocol achieves short-term fairness when the sliding window size is W 3 packets per user in the worst case. However, for the same window size of 3, both the IEEE 802.11 MAC and FCR MAC protocols can only reach 65% and 14% fairness, respectively, in the very best case. Furthermore, our protocol's fairness increases very sharply when the sliding window size is increased. For example, the C-MAC reaches 99% fairness within seven packets per user, while the IEEE 802.11 MAC reaches 97.8% fairness at 200 packets per user in the very best case.

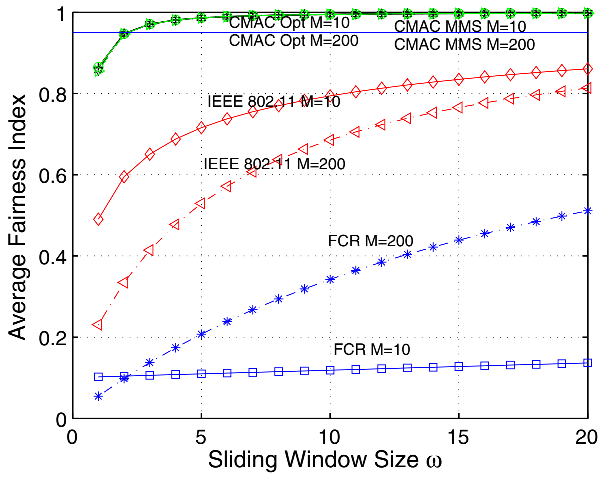


Fig. 13. Average fairness index.

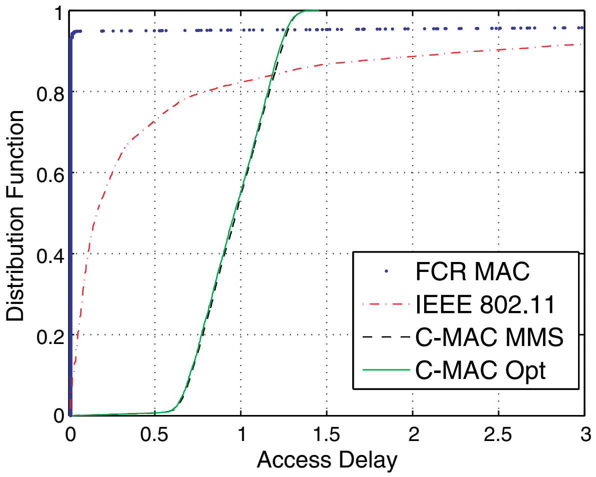
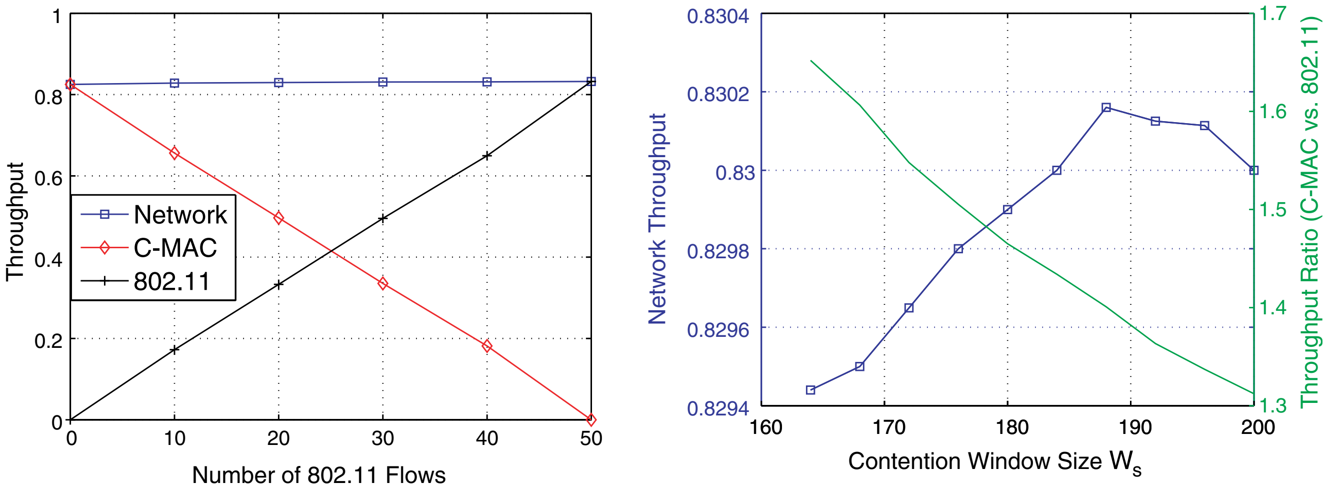


Fig. 14. Access delay distribution for $M = 100$.

In Fig. 14, we depict the access delay distribution of the IEEE 802.11 MAC, FCR MAC, and C-MAC protocols for 100 users under the RTS/CTS mechanism. Real-time and TCP applications prefer a small access delay between consecutive packet transmission, since it improves the performance of real-time traffic and TCP by smoothing the ACKs and the packets [2]. Fig. 14 shows that the C-MAC protocol has a very small variation on access delay: the maximum values are 1.32 and 1.37 s for optimal and mms parameters, respectively. On the other hand, the IEEE 802.11 and FCR MAC protocols have heavy-tailed distributions, and the maximum access delay is 11.78 and 24.15 s, respectively.

5.4. Heterogenous device

We believe that MAC protocols will be implemented as a softMAC rather than as a firmware in near future. However, there would be some cases where the C-MAC must coexist with the IEEE 802.11 MAC protocol. The C-MAC requires $W_c \geqslant 2$ in order to provide two-level strict priority mechanism. Even if we set W_c 2, a regular node with the C-MAC has to sense the channel to be idle at least DIFS $\text{Sifs} + 3 \cdot \text{SlotTime}$, while a node with the IEEE 802.11 MAC needs only DIFS



(a) Throughput (b) Throughput Ratio under Different W_s

Fig. 15. The effect of heterogenous device.

Sifs + 2 · SlotTime. With this setting, and under a heavy load, nodes with the C-MAC encounter throughput starvation. In order to eliminate this problem, we make a simple modification to the C-MAC protocol. We set DIFS Sifs + 2 · SlotTime as in the IEEE 802.11 MAC. If there is a collision, collided nodes need to sense the channel to be idle a duration of either PIFS Sifs + SlotTime or DIFS Sifs + 2 · SlotTime. This setting still gives collided nodes a high priority to transmit since a regular node will have a backoff counter bigger than or equal to 1. Additionally, we require a collided node to use DIFS when it sense a new collision before it transmits.

For simulation, we set the number of users to 50 and vary the ratio between nodes with the C-MAC and nodes with the IEEE 802.11 MAC. We first calculate the optimal value of W_s 74 when there are 50 nodes with the C-MAC. For other settings, we set W_s 74. In Fig. 15a, we depict the network throughput, the throughput of nodes with the C-MAC and with the IEEE 802.11 MAC. First, the modified C-MAC encounters 2% throughput degradation when compared with the original C-MAC with optimal setting. Second, the network throughput slightly increases when the number of nodes with the IEEE 802.11 MAC increases, up to 0.6%. Third, when the number of nodes with the IEEE 802.11 MAC is small, they acquire more than their fair share of throughput. For example, when there are 10 nodes with the IEEE 802.11 MAC, they receive 18.22% of the network throughput rather than their fair share (16.57%). However, when the number of nodes with the IEEE 802.11 MAC increases, this setting favors nodes with the C-MAC. For example, when there are 10 nodes with the C-MAC, they receive 19.17% of the network throughput rather than their fair share (16.67%). We repeat the simulation for other number of users, and we achieve similar results. The simulation results indicate that the modified C-MAC and the IEEE 802.11 MAC can coexist, and they will receive throughput within 8% of their fair share, in the worst case. We also investigate the effect of W_s on network throughput and throughput ratio (the C-MAC vs. the IEEE 802.11 MAC) when there are 30 nodes with the C-MAC and 20 nodes with the IEEE 802.11. Fig. 15b shows that network throughput (solid line with diamonds) has a concave shape as explained before. When we consider the throughput ratio (solid line), a smaller W_s favors nodes with the C-MAC while a larger value favors nodes with the IEEE 802.11 MAC.

6. Conclusions

The goal of this work is to design a distributed medium access protocol to achieve short-term fairness in order to improve the performance of real-time and TCP applications. To achieve this, we propose a novel and efficient cooperative MAC protocol that requires a simple backoff algorithm where users who collide use shorter backoff counters and interframe-spaces to access the channel than users who did not participate in the collision event. The proposed C-MAC protocol has the same level of complexity as the standard IEEE 802.11 MAC protocol, since both consider only collisions in the backoff procedures. The proposed C-MAC protocol achieves short-term fairness within a sliding window size of two to three packets per user, whereas the IEEE 802.11 MAC protocol requires sliding windows of 79–160 packets per user. As a result of the short-term fairness property, the C-MAC protocol yields significant reduction in access delay.

For the proposed cooperative MAC protocol, we utilize an analytical throughput model to obtain the optimal parameter settings for a known number of users. We introduce a simple technique that relies on the short-term fairness property of the C-MAC protocol to estimate the average number of users in the system. Simulation results show that both optimal and estimated parameters provide steady throughput for any given number of users. We also use the minimum mean square error estimation to obtain constant system parameters if the user-estimation technique is not used. Simulation results show that the C-MAC protocol, when used with the optimal, user estimation, or minimum mean square error estimation parameters, achieves throughput performance compatible with the IEEE 802.11 MAC protocol when the RTS/CTS mechanism is applied and achieves significantly better performance with the basic access mechanism.

References

[1] IEEE, IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[2] C.E. Koksal, H. Kassab, H. Balakrishnan, An Analysis of short term fairness in wireless media access protocols (poster), in: Proceedings of ACM SIGMETRICS'00, Santa Clara, CA, June 2000.

[3] A. Chandra, V. Gummalla, J.O. Limb, Wireless medium access control protocols, IEEE Communications Surveys (2000) (Second Quarter).

[4] T. Nandagopal, T. Kim, X. Gao, V. Bhargavan, Achieving MAC layer fairness in wireless packet networks, in: Proceedings of ACM MOBICOM'00, Boston, MA, August 2000.

[5] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E. Knightly, Ordered packet scheduling in wireless ad hoc networks: mechanisms and performance analysis, in: Proceedings of ACM MobiHoc'02, July 2002.

[6] N.H. Vaidya, P. Bahl, S. Gupta, Distributed fair scheduling in a wireless LAN, in: Proceedings of ACM MOBICOM'00, Boston, MA, August 2000.

[7] H. Luo, P. Medvedev, J. Cheng, S. Lu, A self coordinating approach to distributed fair queueing in ad hoc wireless networks, in: Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, April 2001.

[8] T. Ozugur, M. Naghshineh, P. Kermani, J. Copeland, Fair media access for wireless LANs, in: Proceedings of IEEE GLOBECOM'99.

[9] V. Bharghavan, A.J. Demers, S. Shenker, L. Zhang, MACAW: a media access protocol for wireless LANs, in: Proceedings of ACM SIGCOMM'94, London, England, August 1994.

[10] Y. Kwon, Y. Fang, H. Latchman, A novel MAC protocol with fast collision resolution for wireless LANs, in: Proceedings of IEEE INFOCOM 2003, San Francisco, CA, Aprril 2003.

[11] F. Cali, M. Conti, E. Gregori, Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit, IEEE/ACM Transactions on Networking 8 (6) (2000) 785 799.

[12] C.L. Fullmer, J.J. Garcia Luna Aceves, Floor acquisition multiple access (FAMA) for packet radio networks, in: Proceedings of ACM SIGCOMM'95, Cambridge, MA, September 1995.

[13] V. Bharghavan, Performance evaluation of algorithms for wireless medium access, in: IEEE International Computer Performance and Dependability Symposium IPDS'98, 1998.

[14] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard, B. Ramamurthi, Packet reservation multiple access for local wireless communications, IEEE Transactions on Communications 37 (8) (1989) 885 890.

[15] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications 18 (2000) 535 547.

[16] H. Luo, S. Lu, A topology independent fair queueing model in ad hoc wireless networks, in: Proceedings of IEEE ICNP'00, Osaka, Japan, August 2000.

[17] H. Luo, S. Lu, V. Bharghavan, A new model for packet scheduling in multihop wireless networks, in: Proceedings of ACM MOBICOM'00, Boston, MA, August 2000.

[18] N. Vaidya, P. Bahl, Fair scheduling in broadcast enviroments, Microsoft Research Tech. Rep. MSR TR 99 61, 1999.

[19] S.J. Golestani, A self clocked fair queueing for broadband applications, in: Proceedings of IEEE INFOCOM 1994, Toronto, Ontario, April 1994.

[20] P. Goyal, H.M. Vin, H. Chen, Start time fair queueing: a scheduling algorithm for integrated services packet switching network, IEEE/ACM Transactions on Networking 5 (5) (1997) 690 704, October.

[21] C. Barrack, K.Y. Siu, A distributed scheduling algorithm for quality of service support in multiaccess networks, in: Proceedings of IEEE ICNP'99, October 1999.

[22] C. Cetinkaya, F. Orsun, Cooperative medium access control for dense wireless networks, in: Proceedings of Med HOC 2004, Bodrum, Turkey, June 2004.

[23] D. Chiu, R. Jain, Analysis of the increase and decrease algorithms for congestion avoidance in computer networks, in: Computer Networks and ISDN Systems, vol. 17, 1989.

[24] R. Jain, The Art of Computer Systems Performance Analysis, John Wiley & Sons, 1991.

[25] VINT Group, UCB/LBNL/VINT network simulator ns (version 2). http://mash.cs.berkeley.edu/ns.

[26] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley & Sons, 1991.

[27] N. Sahinidis, M. Tawarmalani, BARON Global Optimiza tion Software, http://archimedes.scs.uiuc.edu/baron/baron.html.

[28] G. Bianchi, I. Tinnirello, Kalman Filter estimation of the number of competing terminals in an IEEE 802.11 network, in: Proceedings of IEEE INFOCOM 2003, San Francisco, CA, April 2003.

[29] F. Orsun, H. Topakkaya, M.A. Tunc, C. Cetinkaya, Service differentiation mechanism via cooperative medium access control protocol, in: Proceedings of Med HOC 2005, Porquerolles Island, France, June 2005.

[30] B. Sadeghi, V. Kanodia, A. Sabharwal, E. Knightly, Opportunistic media access for multirate ad hoc networks, in: Proceedings of ACM MOBICOM'02, Atlanta, GA, September 2002.

[31] G. Holland, N. Vaidya, P. Bahl, A rate adaptive MAC protocol for multi hop wireless networks, in: Proceedings of ACM MOBICOM'01, Rome, Italy, 2001.