A Medium Access Control Scheme for Wireless LANs with Constant-Time Contention

Zakhia Abichar, *Student Member*, *IEEE*, and J. Morris Chang, *Senior Member*, *IEEE*

Abstract—In today's wireless networks, stations using the IEEE 802.11 Standard contend for the channel using the Distributed Coordination Function (DCF). Research has shown that DCF's performance degrades especially with the large number of stations. This becomes more concerning due to the increasing proliferation of wireless devices. In this paper, we present a Medium Access Control (MAC) scheme for wireless LANs and compare its performance to DCF and to other efficient schemes. Our scheme, which attempts to resolve the contention in a constant number of slots (or constant time), is called CONTI. The contention resolution happens over a predefined number of slots. In a slot, the stations probabilistically send a jam signal on the channel. The stations listening retire if they hear a jam signal. The others continue to the next slot. Over several slots, we aim to have one station remaining in the contention, which will then transmit its data. We find the optimal parameters of CONTI and present an analysis on its performance. More comprehensive evaluation is presented in the simulation results where we compare CONTI, DCF, and other efficient schemes from the literature. We consider the number of slots used, the collision rate, the throughput, the delay, and the fairness. The highest throughput was achieved by CONTI. Moreover, our results provide measurements from each of the schemes that we consider and provide the insight on each scheme's operation.

Index Terms—Computer networks, wireless LAN, access protocols.

1 Introduction

The use of wireless networks in everyday computing has been a success story and new wireless technologies continue to emerge. Nowadays, wireless networks are a necessary part of the computing world. This was made possible by the IEEE 802.11 Standard [1] which provides technical specifications for the wireless interfaces. In addition, the Wi-Fi Alliance was formed to certify interoperability of wireless products from various vendors. The Medium Access Control (MAC) scheme in the standard that is most widely used is the Distributed Coordination Function (DCF). Its function is to arbitrate the use of the medium to multiple stations that are connected to one Access Point (AP) in the infrastructure mode. In addition, DCF can be used in the infrastructure-less, or ad hoc, mode in which there is no AP. This paper presents a study on the MAC schemes. We propose a new scheme and we make a comparison between our scheme, DCF and several other efficient schemes.

The contention with DCF works as the following. The stations use Contention Windows (CW) to randomize their access and try to avoid collisions. Initially, a station waits for DIFS (DCF Inter Frame Space) and transmits if the channel is idle. However, if the channel is busy, the CW is used. The CW is initially assigned to a preset value, CW_{min}, which depends on the physical layer. Then, a station sets a backoff (BO) counter to a random value chosen from a uniform distribution from $[0, CW]$. The station decreases the BO counter by one for every time slot the channel is idle. If a busy channel is detected, the BO counter is freezed and the countdown resumes from the freeze value after the channel is idle for a duration of DIFS. The station transmits when its BO counter reaches zero. If two or more stations reach zero at the same time, there will be a collision and the transmitted frames won't be received correctly. The colliding stations will not receive an ACK frame and they will double their CW (until it reaches the maximum value equal to CW_{max}). On the other hand, when a station transmits a data frame successfully, its CW is reset to the initial value CW_{min}.

There are numerous studies on DCF in the literature. Some studies presented performance models of DCF to characterize its performance [2], [3], [4]. One observation from the results indicates that DCF's performance degrades significantly with an increase in the number of stations. While this wasn't an issue at the inception of DCF, now more and more people use wireless connections and this becomes a limitation practically. The decrease of performance in this case is attributed to the large number of collisions with the increase in number of stations. Other evaluations [5] of DCF show that its delay might be very large with busy traffic conditions. Finally, the fairness of DCF has been considered [6] and it was shown that DCF doesn't have a high fairness in the short-term, although its fairness increases as the stations contend for longer periods.

In this paper, we present a MAC scheme that provides access by resolving the contention between stations. The main feature of our scheme is that it attempts to resolve the contention in the same number of slots every time. Our scheme, which attempts to resolve the contention in a CONstant TIme, is called CONTI. The contention resolution has several slots. At the first slot, all the stations with frames to transmit contend. The stations, with a probability that we define, choose an event of sending a jam on the channel for the slot duration. The jam is simply a burst of energy (similar to the HIPERLAN scheme [7] and Blackburst

- *The authors are with the Department of Electrical and Computer Engineering, Durham Center, Iowa State University, Ames, IA 50011. E-mail: {abicharz, morris}@iastate.edu.*

Manuscript received 28 July 2009; revised 2 Jan. 2010; accepted 23 Jan. 2010; published online 25 Aug. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-07-0304. Digital Object Identifier no. 10.1109/TMC.2010.157.

1536-1233/11/$26.00 © 2011 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

scheme in [8]) and doesn't need to contain any specific information. With the complementary probability, the stations choose an event of listening to the medium. During a slot, stations retire from the contention if they were listening and hear a jam, which we call preemption. The remaining stations move on to the next slot and repeat the contention. We aim to have one remaining station at the end of contention to provide it with access to the medium.

We present two methods for finding the parameters. The parameters are the number of slots in a contention and the probability corresponding to every slot. The first method aims at bounding the collision rate. The smallest number of slots and the corresponding probability vector are found so that the collision rate is bounded by a given value. It uses an algorithmic approach and finds the parameters with a low computation time. The second method uses an optimization approach to find the parameters that maximize the throughput. While this method requires a larger computation time, we only need to find the parameters offline and make them known to the stations. Thus, during the data transmission, this method doesn't need to operate; instead the parameters that it has produced are readily used.

Additionally, we present an analysis that characterizes the throughputs of CONTI and DCF and compare them to each other. We also show analytically the effect of the slot duration on the performance achieved by CONTI.

In the simulation results, we compare CONTI and DCF, and we program the simulation code of other proposed MAC schemes in the literature. We implement all of the schemes in the same simulation environment to ensure the fairness of comparison. We present results on the number of slots used in the contention resolution, the collision rate, and the throughput. We show that CONTI achieves the highest throughput among the schemes, and we provide the insight on the behavior of the various schemes. We also show the results of delay and the fairness of the schemes and explain the trends that were observed.

The rest of the paper is organized as follows: Section 2 presents the related work that we compare against, in addition to other schemes in the MAC area. Section 3 presents the specifications of our scheme and Section 4 gives an analysis to obtain the optimal parameters of CONTI by bounding the collision rate. Section 5 presents another type of analysis on the parameters in order to maximize the throughput. Section 6 characterizes the performance gain of CONTI and analyzes the effect of the slot duration on the performance. Section 7 presents the simulation results comparison of multiple MAC schemes and, finally, Section 8 presents the conclusion of the paper.

2 Related Work

There have been numerous MAC schemes proposed in the literature. We reference in this section some of the well-known schemes and we compare against them in the simulation results. We highlight here two types of schemes: 1) the contention window (CW)-based schemes and 2) the jamming-based schemes.

2.1 PREMA

The scheme *Prioritized Repeated Eliminations Multiple Access* (PREMA) was proposed in [9]. PREMA is a jamming-based scheme. It works as the following. Contending stations transmit a jam, whose length in slots is drawn from a geometric distribution with parameter q. After the last jam slot, the stations do one slot of carrier sense. If they hear another ongoing slot, they're out of this contention. If not, it means they passed this elimination. The stations with the longest burst will survive the elimination. Following, they do another elimination by choosing another random number from the same distribution and jamming and then one slot of carrier sense. The number of eliminations is a parameter called h. The authors of PREMA use the parameters $h = 4$ and $q = 0.5$. We use these parameters when we compare our work with PREMA.

2.2 k-EC

The scheme *k-Round Elimination Contention* (k-EC) was proposed in [10], which, like PREMA, is a jamming-based scheme. It also has several rounds of eliminations in a contention. There are k rounds of elimination, where k is a parameter. A round of k-EC consists of at most m slots. The contending stations choose a random number uniformly from $[0, m-1]$ and transmit only one jam in the slot number. If a station chooses 0, then it's the first slot, etc. If the station is not jamming, then it should be listening by carrier sense. When a station hears a jam while it's listening, it drops out of the contention and the round is finished for it. The other stations survive and move to the next round. Since the jam can happen in any slot, a round of k-EC is at most m slots long. The authors of k-EC use $k = 7$ and $m = 3$ as the best parameters. As a result, a contention of k-EC with these parameters takes anywhere from 7 slots to 21 slots.

In the proposal of k-EC in [10], the authors compare k-EC to our initial work CONTI that was published in [11]. However, we have the following comments on their comparison:

- In k-EC [10], the authors indicate that they take the results of CONTI from our paper [11], rather than programming CONTI in the simulator. This comparison will reduce the accuracy of the result, since the computation environments of the two simulations are different and the physical layer characteristics would not be the same. However, in this paper, we implement both CONTI and k-EC in the same environment and use the same PHY characteristics.
- Second, both k-EC and CONTI use the jams in the same way. They need the same requirements in transmit length, since they serve a similar function in both schemes. In k-EC, they use the jam slot time of $10\ \mu$s. However, in CONTI, we used the slot time of $20\ \mu$s as in the standard. So, to have a fair comparison of the MAC schemes, we need to have the same parameters on the slot time, which we do in our simulations.

2.3 Idle Sense

The scheme *Idle Sense* was originally proposed in [12] and then it was revised in [13]. In our work, we consider the revised Idle Sense scheme as in [13]. Unlike PREMA and k-EC, Idle Sense is based on the contention window (CW) mechanism, like the standard's DCF scheme. The main idea of Idle Sense is observing that there is an optimal number of slots between two consecutive transmissions. This number is deemed to be $n_i^{target} = 3.91$ for the 802.11b physical layer. It is suitable for a large number of scenarios with varying number of contending stations. Hence, in Idle Sense, all the stations observe the number of slots and adjust the CW up or down to match the number of observed idle slots to the target value. The CW is adjusted up by $CW \leftarrow CW + \epsilon$, and down by $CW \leftarrow \alpha.CW$. The optimal parameters presented are $1/\alpha = 1.0666$ and $\epsilon = 6.0$.

The above procedure of adjusting the CW is performed periodically after the elapse of a number of transmissions given by $maxtrans$. Initially, $maxtrans = 5$, but when the above procedure is done, $maxtrans$ might be changed if the number of idle slots is far off from n_i^{target} so as to speed up the convergence. Specifically, $if(|n_i^{target} - n_i| \geq \beta)$, then $maxtrans \leftarrow 5$; this value is considered small. Otherwise, $maxtrans \leftarrow CW/\gamma$; this value will keep $maxtrans$ proportional to the number of active stations and will be roughly equal to $3n$ (n is the number of active stations) in IEEE 802.11b as explained in [13]. The optimal parameters are given as $\beta = 0.75$ and $\gamma = 4$. We use the same parameters when we compare our work to Idle Sense in the simulation.

2.4 Other Approaches

Other proposed approaches based on jams are in [14], [15], [16], [17], [18], [7]. Other proposed approaches to optimize the CW schemes are in [19], [20], [21], [22], [23], [24]. In this paper, we compare our scheme to the standard's DCF, PREMA, k-EC, and Idle Sense.

3 Medium Access Control (MAC) with Constant-Time Contention

This section presents the proposed MAC scheme, CONTI, which attempts to resolve the contention using a constant number of slots. We start by defining the terms that are used in our work.

3.1 Notations

- The number of stations in the WLAN cell is given by n.
- The contention is resolved over a number of slots given by k, and the contention slots are labeled $\{s_1, \ldots, s_k\}$.
- During a contention slot, a station either transmits a pulse, called *signal 1* or listens to the channel, called *signal 0*.
- The probability vector used by the stations to decide whether to transmit a pulse or listen is given by $p : \{p_1, \ldots, p_k\}$. A station will choose signal 1 during slot s_i with probability p_i. Otherwise, a signal 0 is chosen with a probability $1 - p_i$.
- The number of remaining stations in the contention at the end of the slots is designated by the vector $r : \{r_0, \ldots, r_k\}$. So, $r_0 = n$ stations start the contention, and r_i stations remain in the contention at the end of slot s_i.
- An instance of CONTI is characterized by its parameters, the number of slots k and the probability vector p. Thus, an instance of the scheme, S, is designated by $S(k,p)$.

3.2 Contention

The contention of n stations is resolved using CONTI over k contention slots. Each of the stations uses the same probability vector p. All of the stations go through the following procedure. Before a contention slot s_i, a station chooses signal 1 with probability p_i or signal 0 with probability $1 - p_i$.

During a contention slot, the station will transmit a pulse on the channel if it has signal 1. Otherwise, the station will listen to the channel. The pulse that is transmitted doesn't need to contain information. Rather, its presence on the channel indicates to other stations that some stations have chosen a signal 1. A station that is listening and hears the presence of a signal on the channel is said to be preempted, and this station doesn't contend anymore in this contention. But if a station with signal 0 doesn't hear a signal, it stays in the contention. If the station has signal 1, it transmits the pulse and moves to the next contention slot. At the end of the last slot, a station transmits its data frame if it has not been preempted.

During a contention slot, it is better to eliminate the largest number of stations possible. This means that the contention resolution is occurring quickly and the amount of time spent on contention resolution is minimized. Before slot s_i, there are r_{i-1} stations. At the end of slot s_i, there are r_i stations that are remaining in the contention. Thus, slot s_i has eliminated $(r_{i-1} - r_i)$ stations from the contention, which we seek to maximize.

With CONTI, it is possible that no stations are eliminated during a contention slot. This happens if all the stations choose signal 1. Then, no station is preempted. It also happens if all the stations choose a signal 0. If this event happens, then the following slots will continue the contention. But if it happens in the last slot and there are more than one station remaining, there will be a collision.

For an efficient contention resolution, the probability choices should be optimized to minimize the collision rate. The number of slots should also be minimized so that the time spent in the contention is reduced. For that purpose, the values of k and the vector p are optimized in the next section.

Finally, we add a stipulation that ensure compatibility with the Inter-Frame Spacing used in wireless networks, such as DIFS in the standard. In CONTI, there might be a few consecutive slots where all of the stations choose signal 0. Thus, a station that had already retired from contention should not count this silent time in its IFS timer. Thus, we require a station that has retired to stop its IFS timer until the contention is finished. Since the station knows the number of slots, k, a priori, it can do that.

3.3 Example

An example on the contention resolution using CONTI is presented in Fig. 1. There are six stations. In the first slot, stations 2, 4, and 5 choose signal 1 and preempt stations 1, 3, and 6. Thus, stations 1, 3, and 6 don't contend anymore in this round. The graph on the left side of Fig. 1 shows the signals, while the graph on the right side depicts the jams. In the second slot, stations 2, 4, and 5 choose signal 0 and no station is preempted. All the stations move to the third slot. In the third slot, stations 2 and 4 preempt station 5. Finally,

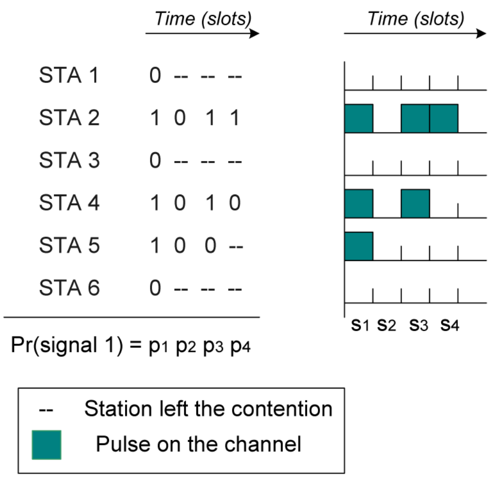

Fig. 1. Contention resolution using CONTI.

in the last slot, station 2 preempts station 4. Then, station 2 is able to transmit a data frame. In this example, $n = 6$, $k = 4$ and the vector r is $\{6, 3, 3, 2, 1\}$.

3.4 Pseudocode

The contention resolution using CONTI is specified in the pseudocode in Algorithm 1. The pseudocode describes the operation of a CONTI instance $S(k, p)$. In Algorithm 1, the state variable $retire$ indicates if the station has been preempted, when $retire = 1$, or if the station is still in the contention, when $retire = 0$.

Algorithm 1. Contention Resolution with CONTI

$retire := 0$
$i := 1$
while $(i \le k)$ **do**
  if $retire = 1$ /* Station has been preempted */
    $defer(t_{slot})$
  else if $retire = 0$ /* Station in contention */
    $proba := Uniform(0, 1)$ /* Choose signal 1 or 0 */
    if $proba < p_i$
      $signal := 1$
    else $signal := 0$
    if $signal = 1$ /* Station with signal 1 */
      $pulse(t_{slot})$
    else if $signal = 0$ /* Station with signal 0 */
      $listen(t_{slot})$
      if $pulseDetected(t_{slot}) = true$
        $retire := 1$
  $i := i + 1$

The function $defer(t)$ makes the station remain idle for a duration of t. The function $pulse(t)$ makes the station transmit a pulse for a duration of t. Finally, the function $pulseDetect(t)$ makes the stations detect if there has been a pulse on the channel during a time slot.

3.5 Why Do We Need to Optimize the Parameters?

In this section, we present an example that shows how the parameters affect the contention resolution in CONTI. In other approaches in the literature that use similar probabilistic jams [14], the probabilities are chosen randomly. This example shows that we can have performance gain by optimizing the parameters. In this example, we compare two cases. First, we consider intuitively chosen values of $p_i = 0.5$ for all the slots. Then, we show another case with more appropriate parameter choices.

TABLE 1
Percentage Collision Rate $(p_i = 0.5, (1 \le i \le k))$

k	2	3	4	5	6	7	8
$n = 10$	80.73	50.95	28.33	14.89	7.62	3.86	1.94
$n = 25$	99.37	87.00	59.21	34.21	18.31	9.46	4.80

In the case of $p_i = 0.5$ for $1 \le i \le k$, we show how the collision rate is affected by the number of slots used. The result is shown in Table 1 for the number of stations 10 and 25. With five contention slots, the collision rate[1] of 10 stations is 14.89 percent and with 25 stations it is 34.21 percent. Also, as one might think intuitively, the result shows that an increase in the number of slots reduces the collision rate.

Now we consider the use of CONTI with five slots and with the following probability vector[2] $p : \{0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$. For this vector, the collision rate for 10 stations is 7.59 percent compared to 14.89 percent with the intuitive choice of $p_i = 0.5$. This is a reduction by almost a factor of 2. For 25 stations, the collision rate is 13.65 percent compared to 34.21 percent using the intuitive choice. This shows that optimized parameters have a significant effect on the performance of contention resolution with CONTI.

4 Parameters: Bounded Collision Rate

This section presents an analysis to find the optimal parameters of CONTI based on a bounded collision rate. The analysis also shows that for a wide range of n, we can use the same parameters, while remaining close to the optimal performance. Hence, the stations don't need to know or approximate n.

This section has the following contents. First, the definition of optimal solution is presented and then we show the algorithmic concepts that we use to get the parameters. Then, the collision rate is found analytically, which will be used to find the parameters. Then, k and p are found for a given number of stations, n. Finally, we show that there are certain k and p that can be used efficiently, independently of n for a range of realistic cases.

4.1 Defining the Optimal Solution

The optimal solution with CONTI can be defined in more than one way. If we seek to minimize the collision rate, say to make it equal to zero, we can do that but we risk making the number of slots too large. On the other hand, if we use a small number of slots to avoid wasting time in the contention, we might end up with a large collision rate that adversely affects the throughput. Thus, we use a definition that is a trade-off between these two factors. Our definition of optimal solution is based on the idea that we

1. The table shows analytic result of the collision rate. The expression of the collision rate is presented in Section 4.
2. We show how to derive the numbers in this vector in Section 4.

can tolerate a small collision rate. Then, we need to find the minimum number of slots and the corresponding probabilities such that the collision rate is bounded.

Definition 1 (Optimal Solution). *The minimum value of k and a probability vector p should be found such that the collision rate p_{coll} is bounded by a value given in p^*_{coll}.*

4.2 Solution Approach

In a problem satisfying the *optimal substructure property* [25] and the *greedy-choice property*, a greedy solution yields an optimal solution. In the greedy solution, there are several choices of parameters (different p-values at different contention slots) and the best choices are found at each step. For CONTI, that means the value of p that minimizes the number of stations remaining in contention are found in each contention slot. It remains to show that CONTI satisfies the two properties mentioned above.

A problem is said to have the *optimal substructure property* if an optimal solution to the problem contains within it optimal solutions to subproblems. This is proved for the case of CONTI by contradiction. Consider an optimal solution $S_\alpha(k_\alpha, p_\alpha)$ that satisfies the upperbound condition on the collision rate p^*_{coll}. If at any slot s_i, the choice of p_i in S_α doesn't minimize the number of remaining stations (r_i), then p_i can be replaced with another optimal value p_i^{opt} that minimizes r_i. Then, k_α might be reduced or a smaller value for p^α_{coll} is obtained, which implies that S_α is not an optimal solution.

By definition, a problem has the *greedy-choice property* if a greedy choice that is already made combined with an optimal solution to the remaining subproblem yields an optimal solution. To show this property for CONTI, consider a case of contention resolution where a greedy choice is made at the first slot s_1. The greedy choice minimizes the number of remaining stations (r_1) at the end of the slot. An optimal solution for the subproblem with r_1 stations and $k-1$ slots, combined with the greedy solution at the first slot yields an optimal solution because it contains the minimum k value.

4.3 The Collision Rate

Let the term $\sigma(n,k,p)$ be the probability that the instance of CONTI, $S(k,p)$, resolves the contention successfully for n stations.

Next, the probability of preempting stations over one slot is defined. Consider a contention slot s_i, where $r_{i-1} = u$ stations start the contention and $r_i = v$ stations remain at the end of the slot. Let the probability of this event be designated by $\tau_{u,v}(p_i)$. Its expression is the following:

$$\tau_{u,v}(p_i) = \begin{cases} \binom{u}{v}.(p_i)^v.(1-p_i)^{u-v}, & 1 \le v \le u-1, \\ (p_i)^u + (1-p_i)^u, & v = u. \end{cases} \tag{1}$$

In the first case, v out of u stations remain at the end of the slot. In the second case, all of the u stations remain at the end of the slot. This happens if all the stations choose the same signal, whether it is signal 1 or signal 0.

Before we can give the analytic expression of the collision rate, we give the following definition of the subvector of p.

Definition 2. *For a vector $p : \{p_1, p_2, \ldots, p_k\}$ with k elements, the term $\pi_i (1 \le i \le k)$ defines the subvector of p with $k-i+1$ elements given by $\pi_i : \{p_i, p_{i+1}, \ldots, p_k\}$, so it is a suffix subvector.*

Finally, the collision rate of CONTI is given by the following theorem:

Theorem 1. *The probability that scheme $S(k,p)$ resolves the contention successfully for n stations is given by:*

$$\sigma(n,k,p) = \Sigma_{i=0}^{n-1}[\tau_{n,n-i}(p_1).\sigma(n-i,k-1,\pi_2)]. \tag{2}$$

The trivial cases for the expression of σ are the following:

$$\sigma(1, i, \pi_{(k-i+1)}) = 1, \qquad 1 \le i \le k, \tag{3}$$

$$\sigma(v, 1, \pi_k) = \tau_{v,1}(p_k), \qquad v \ge 1. \tag{4}$$

Proof. First, notice that π_1 designates the same vector as p. Then, $\sigma(n,k,p)$ can be rewritten as $\sigma(n,k,\pi_1)$. After the elapse of one slot, the number of stations is reduced from $r_0 = n$ to r_1, where $r_0 \ge r_1$. The remaining problem is the contention resolution of r_1 stations in $k-1$ slots using the vector π_2. This subproblem is solved successfully with a probability given by $\sigma(r_1, k-1, \pi_2)$.

During the first slot, the number of stations that are preempted is between 0 and $n-1$. Each of these events occur with a probability of $\tau_{n,n}(p_1), \ldots, \tau_{n,1}(p_1)$, respectively. Thus, $\sigma(n,k,p)$ is equal to the following expression:

$$\begin{aligned} \sigma(n,k,\pi_1) = {} & \tau_{n,n}(p_1).\sigma(n,k-1,\pi_2) \\ & + \tau_{n,n-1}(p_1).\sigma(n-1,k-1,\pi_2) \\ & + \cdots + \\ & \tau_{n,1}(p_1).\sigma(1,k-1,\pi_2). \end{aligned} \tag{5}$$

The trivial cases for the expression of σ are given in (3) and (4). Equation (3) represents the case where there is one station left and there are one or more slots. In this case, the contention is resolved successfully with probability 1. The case in (4) represents the scenario when there is one slot left and there are one or more stations contending. In this case, the contention is resolved correctly if all the stations are preempted except one. This happens with a probability given by $\tau_{v,1}(p_k)$. □

4.4 Finding the Probability Choices

We need to find the values in the vector p that resolve the contention the quickest. Over a slot s_i, the number of stations should be reduced from r_{i-1} to r_i, with having r_i minimized.

One observation about the probability values p_i is the following. In the earlier slots, the value of p_i should be small. This will allow few stations to choose signal 1 and hopefully preempt most of the other stations. Then, the contention resolution proceeds fast. At the later slots, there should be few remaining stations and then p_i should have a larger value so that at least one station will choose a signal 1 and preempt the other stations. The values of p_i drawn from the analysis agree with this observation in that the values in p are nondecreasing. This trend of increasing probability values as the slots progress is also observed in the analysis of the Sift MAC scheme [24]. However, Sift differs from CONTI in that it is based on Contention Windows, not jamming slots, and it has the goal of supporting event-driven sensor networks, not wireless LANs.

TABLE 2
Contention over One Slot (CDF on v)

u	p_i^*	$E[v]$	v^*	$\Pr(v \leq v^*) = CDF(v^*)$
2	0.5	1	1	0.5
3	0.4314	1	2	0.735
4	0.4245	2	3	0.857
5	0.36715	2	4	0.891
10	0.2563	3	5	0.925
20	0.14955	3	9	0.960
30	0.11945	4	10	0.977
40	0.09425	4	10	0.980
50	0.0783	4	10	0.981
60	0.0673	4	10	0.982
70	0.064	5	10	0.985
80	0.05705	5	10	0.985
90	0.0516	5	10	0.984
100	0.04715	5	10	0.984

TABLE 3
The Parameters and the Collision Rate

n	k	p	p_{coll}
2	5	$\{0.5, 0.5, 0.5, 0.5, 0.5\}$	3.12%
3	5	$\{0.4314, 0.4314, 0.5, 0.5, 0.5\}$	4.32%
4	5	$\{0.4245, 0.4245, 0.4314, 0.5, 0.5\}$	5.33%
5	5	$\{0.36715, 0.4245, 0.4314, 0.5, 0.5\}$	5.88%
10	6	$\{0.2563, 0.36715, 0.4245, 0.4314, 0.5, 0.5\}$	3.85%
20	6	$\{0.14955, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.52%
30	6	$\{0.11945, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.53%
40	6	$\{0.09425, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.40%
50	6	$\{0.0783, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.28%
60	6	$\{0.0673, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.21%
70	6	$\{0.064, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.20%
80	6	$\{0.05705, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.26%
90	6	$\{0.0516, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.35%
100	6	$\{0.04715, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$	5.48%

Consider a slot s_i with u contending stations. The probability that v stations remain at the end of the slot is given by $\tau_{u,v}(p_i)$ in (1). The expected value of v is given by the following equation:

$$E[v] = \left[\sum_{v=1}^{u} v.\binom{u}{v}.(p_i)^v.(1-p_i)^{u-v}\right] + u.(1-p_i)^u. \quad (6)$$

For a given value of u, (6) is solved numerically to determine p_i that minimizes v. This value of p_i, denoted by p_i^* reduces the number of stations the quickest over one slot. The corresponding values of u, p_i and $E[v]$ are presented in the three leftmost columns of Table 2. An entry in the table is read as follows: with $u = 100$ stations at the beginning of a slot, using $p_i = 0.04715$ provides an expected number of remaining stations equal to $v = 5$.

4.5 CDF on the Number of Remaining Stations

The information in the three leftmost columns of Table 2 shows the value of p_i that minimizes the expected value of v. However, this information doesn't show the probability that this event ($E[v]$ is minimized) will happen. For the entry with $u = 100$, the event $v = 5$ will occur with a probability of $\tau_{100,5}(0.04715) = 0.178$. Thus, we will have to use the Cumulative Distribution Function (CDF) to have a higher confidence on how the number of stations is reduced over one contention slot.

Let the CDF on v be defined by the function $\theta_{u,v}(p_i)$. This function designates the probability that, starting with u stations, the number of stations at the end of the slot will be less or equal to v. The value of $\theta_{u,v}(p_i)$ is the following:

$$CDF(v) = \theta_{u,v}(p_i) = \sum_{j=1}^{v} \tau_{u,j}(p_i). \quad (7)$$

Using the CDF, a bound (in a probabilistic sense[3]) on the number of stations that survive a contention slot is given in Table 2. At the start of the slot, there are u stations. The column labeled v^* designates the upperbound, in probabilistic terms, on the number of stations that will survive the contention. The last column, labeled $CDF(v^*)$, indicates the probability of this event happening, that is, $\Pr(v \leq v^*)$. As an example, the last entry is interpreted as: starting with $u = 100$ stations, there will remain $v \leq 10$ stations at the end of the slot with a probability of 0.984.

The information in Table 2 is used to derive the number of slots k and the probability vector p that are needed to resolve the contention of n stations. The collision rate will be measured by the analytic expression given in (2).

4.6 Finding the Parameters Given the Number of Stations

Now the number of slots and the probability vector can be found for a given number of stations using Table 2. In the definition of the optimal solution, it was required to minimize the number of slots k and find a corresponding vector p so that the collision rate is smaller than an upperbound, $p_{coll} < p^*_{coll}$. In this paper, we use $p^*_{coll} = 6\%$, which is considered the upperbound on the collision rate that is tolerable. The value of $p^*_{coll} = 6\%$ was chosen based on simulation results.

The parameters for $n = 100$ stations are found as follows. In the first slot, $p_1 = 0.04715$ is used, which yields a number of remaining stations $r_1 \leq 10$, with a high probability (equal to 0.984 from Table 2). In the second slot, we assume that there are 10 stations and then $p_2 = 0.2563$. This leads to $r_2 \leq 5$ with a high probability. Similarly, the number of remaining stations progresses as $r_3 = 4, r_4 = 3, r_5 = 2$, and $r_6 = 1$. The remaining probability choices are $p_3 = 0.36715$, $p_4 = 0.4245, p_5 = 0.4314$, and $p_6 = 0.5$. In total, it takes six slots to resolve the contention when starting with 100 stations. The collision rate measured by the analytic expression is $p_{coll} = 5.48\%$.

The parameters for other values of n are presented in Table 3. There are cases when the CDF values from Table 2 don't correspond to a high certainty. For example, starting with two stations, the best possible event is to have one station remaining. This happens with a probability of 0.5. To satisfy the upperbound on the collision rate of 6 percent, the same value of $p_i = 0.5$ is used over several slots. For the case of $n = 2$, there are five slots each using $p_i = 0.5$. The collision rate is $p_{coll} = 3.12\%$.

4.7 Constant-Time Contention Resolution

In Table 3, the parameters for CONTI were optimized for a given number of stations. This was done independently for

3. In the rest of this section, the meaning of bound *in a probabilistic sense* applies to v^*.

TABLE 4
Collision Rate Comparison

n	k	p_{coll}	k°	p°_{coll}
2	5	3.12%	6	3.92%
3	5	4.32%	6	4.00%
4	5	5.33%	6	4.19%
5	5	5.88%	6	4.37%
10	6	3.85%	6	5.02%
20	6	5.52%	6	5.52%
30	6	5.53%	6	5.53%
40	6	5.40%	6	5.40%
50	6	5.28%	6	5.28%
60	6	5.21%	6	5.21%
70	6	5.20%	6	5.20%
80	6	5.26%	6	5.26%
90	6	5.35%	6	5.35%
100	6	5.48%	6	5.48%

each value of n. However, there are several cases where $20 < n < 100$ that obtained the same number of slots and the same probability values. For the cases where $n < 20$, the number of slots is 5, which is close to the other cases.

Next, the same parameters are used for all the cases of n. The parameters chosen are the ones used for $n = 100$. Accordingly, the scheme $S^{\circ}(k^{\circ}, p^{\circ})$ is defined where $k^{\circ} = 6$ and $p^{\circ} = \{0.04715, 0.2563, 0.36715, 0.4245, 0.4314, 0.5\}$. The collision rate of S° is measured and compared to the case with the previous parameters that are found in Table 3. The comparison results are presented in Table 4. The collision rate varies slightly and it remains below 6 percent for all cases. For some cases, the collision rate is lower for S° but this is because six slots are used, instead of five. Using this observation, it is possible to use the same parameters, as in S°, to resolve the contention independently of the number of stations. This makes it easier to do contention using CONTI as the number of the stations doesn't need to be known and all the stations use the same parameters all the time.

5 Parameters: Maximized Throughput

In this section, we find the optimal parameters for CONTI, k and p, that maximize the throughput.

5.1 Time Utilization

The medium access with CONTI is shown in Fig. 2. In the first attempt, a frame is transmitted successfully and an ACK is received in reply. However, the second attempt is a collision.

Hence, the time utilization of CONTI, designated by ρ, is found as the following:

$$\rho = \frac{p_s.t_{data}}{p_s.T_{successful} + p_c.T_{collision}}, \tag{8}$$

where t_{data} is the time to transmit the data frame. The probabilities of success and collision are given by p_s and p_c, respectively. $T_{successful}$ is the time for a successful transmission cycle, and $T_{collision}$ is the time consumed by a collision cycle, which are given as follows:

$$T_{successful} = t_{difs} + k.t_{slot} + t_{data} + t_{sifs} + t_{ack} \tag{9}$$

and

$$T_{collision} = t_{difs} + k.t_{slot} + t_{data}. \tag{10}$$

To be able to find the time utilization, we need to know the expression of the successful transmission event, p_s, which is given in Section 4.3 in (2).

5.2 Finding the Parameters

According to the above, if we know the number of stations, n, and the data transmission time, t_{data}, we can find the optimal parameters which are the number of slots, k, and the probability vector, p, that contains k elements. The optimal parameters maximize the time utilization of the access scheme.

We are interested in a number of scenarios in the Wireless LAN environment. Each scenario has a number of contending stations and a given frame size. Thus, we optimize the parameters having in mind all of the realistic scenarios in a WLAN environment. We consider the number of stations to be: 2, 5, 7, 10, 15, and 25. The frame sizes that we consider, in bytes, are 250, 600, 950, 1,300, 1,650, 2,000, and the maximum frame size, 2,346. The transmission rate is 11 Mbps.

Let j be the number of scenarios that vary the number of stations (here $j = 6$). And let k be the number of scenarios that vary the frame size (here $k = 7$). Hence, we have 42 scenarios, one for each value of n corresponding to a frame size. For a certain scenario with n stations and a given frame size, the

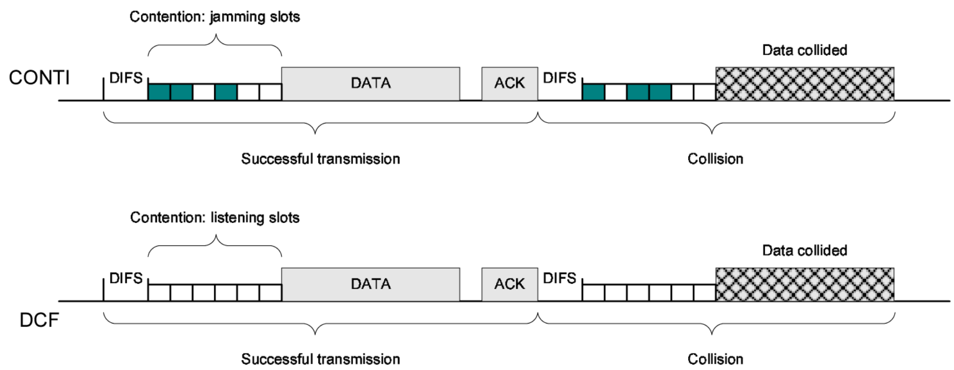

Fig. 2. Transmission cycle of CONTI and DCF: a successful transmission and a collision.

time utilization is designated by $\rho^n_{FrameSize}$. Hence, we need to maximize the average throughput over all the scenarios given by: $\frac{1}{j.k}\sum_j \sum_k \rho^n_{FrameSize}$.

The parameters k and p are found by looking in the search space:

$$\begin{cases} k \geq 1 \\ 0 \leq p_1 \leq 1 \\ 0 \leq p_2 \leq 1 \\ \dots \\ 0 \leq p_k \leq 1 \end{cases} \tag{11}$$

and the objective function is to maximize the average time utilization:

$$\max\left(\frac{1}{j.k}\sum_j \sum_k \rho^n_{FrameSize}\right). \tag{12}$$

The optimal parameters obtained are $k = 7$ slots and the probabilities are $p = \{0.18, 0.31, 0.40, 0.48, 0.48, 0.49, 0.49\}$. For these parameters, the average time utilization over all the scenarios is 65.27 percent. These are the parameters that we use in the simulation results as elaborated in the discussion below.

5.3 Discussion

We have presented two methods to find the parameters of CONTI. The method in Section 4 was based on bounding the collision rate and using an algorithmic approach to finding the parameters k and p. On the other hand, the method in this section is based on maximizing the throughput and searching in the probability space to find the parameters. We compare these two methods as follows:

- The method in Section 4, which uses an algorithmic method to find the parameters, requires less computation time since it optimizes one slot and then moves to the next one. However, the optimization method presented in this section requires more computation time since it considers all of the combinations of p_i values over the slots.
- For the WLAN scenarios that we consider, the computation time is tractable. Hence, we use the results that were produced in this section.
- A comparison of the two methods in the simulation shows almost identical results. Hence, in the simulation we only use the results of the method in this section.

6 Characterization of the Time Utilization

This section presents an analysis on the throughputs of CONTI and DCF. It also shows the effect of the slot length on the relative performance between CONTI and DCF. A more comprehensive evaluation is shown between CONTI, DCF, PREMA, k-EC, and Idle Sense in Section 7 by simulation.

The access schemes of CONTI and DCF are shown in the illustration in Fig. 2. While both of the schemes wait for the initial DIFS interframe space, the contention of CONTI uses the pulses, while the contention of DCF uses the backoff countdown. What follows the contention is similar for the two cases which is the data transmission, the wait of SIFS interframe space, and the transmission of the ACK frame upon successful transmission. In the case of a collision, an ACK frame will not be received.

The characterization of the time utilization between CONTI and DCF can be written as the following:

$$gain_{conti} = \frac{\rho_{conti}}{\rho_{dcf}}. \tag{13}$$

By observation from Fig. 2, and using the derivation similar to (8), we have:

$$gain_{conti} = \frac{p_s^{conti}}{p_s^{dcf}} \times \frac{p_s^{dcf}.T^{dcf}_{successful} + p_c^{dcf}.T^{dcf}_{collision}}{p_s^{conti}.T^{conti}_{successful} + p_c^{conti}.T^{conti}_{collision}}. \tag{14}$$

The expressions for $T^{conti}_{successful}$ and $T^{conti}_{collision}$ are given in (9) and (10), respectively.

The expressions for $T^{dcf}_{successful}$ and $T^{dcf}_{collision}$ are the following:

$$T^{dcf}_{successful} = t_{difs} + t_{contention} + t_{data} + t_{sifs} + t_{ack} \tag{15}$$

and

$$T^{dcf}_{collision} = t_{difs} + t_{contention} + t_{data}, \tag{16}$$

where $t_{contention}$ designates the average number of slots spent in a DCF contention.

6.1 Probability of a Successful Transmission

Since CONTI and DCF employ two different mechanisms for the contention resolution, it is obvious that they have different expressions for the probability of a successful transmission.

The characterization of the collision event in DCF was presented in [2]. Accordingly, the probability that a station transmits in a slot, p_{tr}, and the probability that a station has a successful transmission given a transmission attempt, p_{cond}, are given as follows:

$$p_{tr} = 1 - (1-\tau)^n, \tag{17}$$

$$p_{cond} = \frac{n\tau(1-\tau)^{n-1}}{p_{tr}}, \tag{18}$$

where n is the number of contending stations and τ and p are given as follows:

$$\tau = \frac{2(1-2p)}{(1-2p)(CW_{min}+1) + pCW_{min}(1-(2p)^m)}, \tag{19}$$

$$p = 1 - (1-\tau)^{n-1}. \tag{20}$$

The analytic results of the successful probability of transmission are presented in Table 5. The results of DCF are based on the equations above and from [2], and the results of CONTI are obtained using the equation we derived in (2). For DCF and CONTI, as shown in Table 5, the probability of a successful transmission decreases as the number of stations increases.

6.2 Effect of the Length of the Contention Slot

To understand how the duration of the contention slot affects the system performance, we look at the number of slots that is consumed in each access to the channel. In CONTI, the parametrization has been studied for using

TABLE 5
Analytic Results of Successful Transmission

No. of Stations	10	20	30	50	75
DCF	0.838	0.775	0.737	0.688	0.657
CONTI	0.982	0.976	0.970	0.959	0.944

seven slots. However, it is a different story in DCF. By observing the number of idle slots spent between the end of DIFS and the initial moment of the transmission, this number is different and becomes smaller when the number of the stations increases.

For example, if there are five stations contending with the values of backoff counter equal to 7, 10, 13, 16, and 19, it means in the first access, there will be seven idle slots, and thereafter, there will be three contention slots preceding each data transmission. For the evaluation of the gain in time utilization, three slots are assumed for DCF and seven slots are assumed for CONTI. Of course, CONTI is using more slots but making up in providing a lower collision rate. In this study, the data rate is 11 Mbps, the control rate is 1 Mbps, and the frame size is 1,500 bytes.

The results of the gain in time utilization are shown in Fig. 3. CONTI has a gain in time utilization in comparison to DCF. Expectedly, with a larger time slot, the gain that CONTI has starts to decrease since CONTI is consuming more slots in one access. The other observation is, with the larger number of stations, the gain of CONTI increases due to its better collision rate. The time slots that are shown range in duration from 20 μs, which is commonly used in practice in IEEE 802.11b, to larger values up to 60 μs. The value of 60 μs represents the threshold where the gain of CONTI starts to disappear, for the number of 10 stations. This is seen in Fig. 3, as the lowermost curve approaches 1 on the left-hand side.

More comprehensive evaluation between CONTI, DCF and the other schemes that we consider follow in the simulation results in Section 7.

7 Simulation Results

This section presents the simulation results. We compare the performances of CONTI, DCF, PREMA, k-EC, and Idle Sense. We consider the important measurements for the MAC such as the number of slots used, the collision rate, the throughput, the delay, and the fairness to the users.

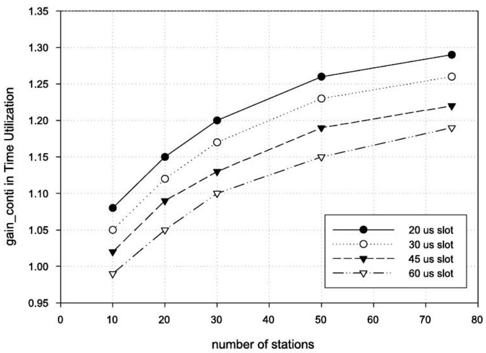

Fig. 3. Effect of the time slot duration.

TABLE 6
Physical Layer Characteristics (802.11b)

Characteristics	Value	Comments
tSlotTime	$20\mu s$	Slot time
tSIFSTime	$10\mu s$	SIFS time
tDIFSTime	$50\mu s$	DIFS = SIFS + 2 × Slot
aCWmin	15	min contention window size
aCWmax	1023	max contention window size
tPLCPOverhead	$192\mu s$	PLCP overhead

7.1 Parameters

We developed a discrete-event simulator for the MAC of wireless networks. The physical layer we consider is the 802.11b [26]. Its parameters, with the parameters of DCF are summarized in Table 6. We program the simulation code and evaluate all the schemes in the same environment and in the same manner to ensure the fairness of evaluation.

The parameters of CONTI, PREMA, k-EC, and Idle Sense are presented in Table 7. The parameters of all the other schemes are the optimal parameters from the papers in which they were proposed.

7.2 Number of Contention Slots

Each of the schemes that we compare requires a certain number of contention slots. While the number of slots spent in contention isn't the only performance indicator, having a small number of slots is generally considered as preferable. On one side, CONTI takes a constant number of seven slots and Idle Sense aims to achieve a number of slots equal to 3.91. On the other side, DCF, PREMA, and k-EC spend a varying number of slots for each contention.

Table 8 shows the average number of slots spent in a contention. In this simulation experiment, the duration time is 1,200 seconds and the frame size is 1,500 bytes. The data rate is 11 Mbps and the control rate is 1 Mbps. The number of stations changes for different simulation runs as shown in Table 8. First, CONTI takes seven slots for every contention, which is already known from the parameters.

TABLE 7
Parameters of CONTI, PREMA, k-EC, and Idle Sense

CONTI
Number of slots: $k = 7$
Probabilities: $p = \{0.18, 0.31, 0.40, 0.48, 0.48, 0.49, 0.49\}$
PREMA
Number of eliminations: $h = 4$
Geometric distribution parameter: $q = 0.5$
k-EC
Number of rounds: $k = 7$
Maximum length of a round: $m = 3$
Idle Sense
Target number of idle slots: $n_i^{target} = 3.91$
CW increase parameter: $\epsilon = 6.0$
CW decrease parameter: $1/\alpha = 1.0666$
Initial window size: $maxtrans = 5$
Window size adjustment parameters: $\beta = 0.75$, $\gamma = 4$

TABLE 8
Average Number of Slots in a Contention

Number of Stations	CONTI	DCF	PREMA	k-EC	Idle Sense
10	7	3.00	11.06	12.01	4.08
20	7	2.34	12.02	11.40	3.81
35	7	2.00	12.82	10.90	3.70
50	7	1.84	13.33	10.58	3.66
75	7	1.69	13.91	10.22	3.64
100	7	1.60	14.32	9.96	3.64

With DCF, the number of slots is reduced with more stations even though the CW size is becoming larger. This happens since the number of slots that are spent is the minimum among all the backoff counters of stations. However, there are more collisions which will be shown next. PREMA has a different trend with more slots spent when there are more stations. This happens since the longest burst prevails in PREMA. With more stations, there's more chance to have a long burst. The k-EC scheme starts at 12 slots and takes less with more stations since the earliest jam finishes an elimination round. Finally, with Idle Sense, the number of slots fluctuates around the target number of 3.91. It is slightly higher with fewer stations but slightly lower with more stations.

From this experiment, we conclude that the schemes with multiround contentions, PREMA and k-EC use the largest number of slots, average more than 10 slots in the above scenarios. On the other hand, the schemes based on the contention window use the least number of slots, average less than four slots. CONTI comes in the middle by using seven slots.

7.3 Collision Rate

This part shows the collision rates of the schemes. The simulation parameters are similar to above with 1,200 seconds of simulation time, 1,500 bytes frames, 11 Mbps of data rate, and 1 Mbps of control rate. The collision rate is shown in Fig. 4.

The number of stations varies in the same numbers as in the previous experiment. First, we notice that PREMA and k-EC have the lowest collision rates. Across all scenarios, their rate is around 1 percent. However, their number of slots used per contention was the highest. CONTI has a collision rate that starts at 1 percent for 10 stations and climbs to 7 percent for 100 stations. Then, Idle Sense has a collision rate that varied between 11 and 14 percent. Finally, DCF had the highest rate of collisions that varied from 16 to 37 percent. From the previous experiment, DCF used the least number of slots per contention, but its collision rate turned out to be much higher than the other schemes. Typically, the trend is that the higher number of slots allows reducing the collision rate.

Finally, we mention an observation between the collision rates of PREMA and k-EC. At a lower number of stations, k-EC has a smaller collision rate. However, their collision rates intersect at about $n = 35$, and for higher n, PREMA has a smaller collision rate. This trend happens since PREMA is based on a longest-burst-prevail policy.

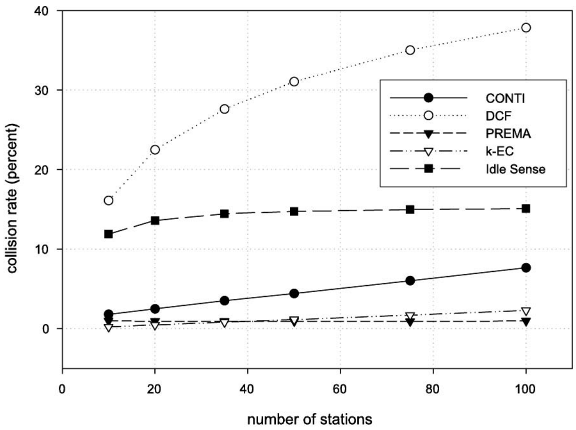

Fig. 4. Collision rate.

Therefore, for a large number of stations, there is more chance that the geometric distribution used will produce a large burst and therefore, there won't be a collision.

7.4 Throughput

The two previous experiments show the insightful measures of number of slots used and the collision rate. However, we need to understand their effect on the throughput that is achieved with the schemes. This part shows the throughput comparison of the schemes. The experiment environment is similar to above with 1,200 seconds of simulation time, 11 Mbps for data transmission, and 1 Mbps of control rate.

We would like to note that since the throughputs of several of the schemes are close to each other, we used a quite large simulation time of 20 minutes to obtain stable results. We noticed that the throughput values at this time are stable for different simulation runs. For each scenario, we made 10 simulation runs and we show the results in Figs. 5a, 5b, and 5c. On top of every column in the chart, there is an error bar at a distance d from the column top. The error is defined as the maximum variation from the average value obtained. That is, the column top is at value x, then, all the values obtained are in the interval $[x - d; x + d]$. In the figures, the error bars are almost coincident with the column top since we encountered typical errors of 0.01 to a maximum 0.11 due to the large simulation time that we used.

The throughputs in Figs. 5a, 5b, and 5c correspond to 5, 20 and 50 stations with 802.11b, respectively. Each figure has seven groups of bars, each corresponding to a frame size, ranging from 250 bytes to the maximum size of 2,346 bytes. We note the following observations:

- The first observation is that, for a number of stations, the throughput increases with the larger frame size. This happens since with larger frames, there is less time spent in contention and more time spent in transmission in the simulation time. The same simulation time of 1,200 seconds applies for all the cases.
- For a low number of stations, as in Fig. 5a, the schemes have almost identical performances, although CONTI maintains a small advantage. The

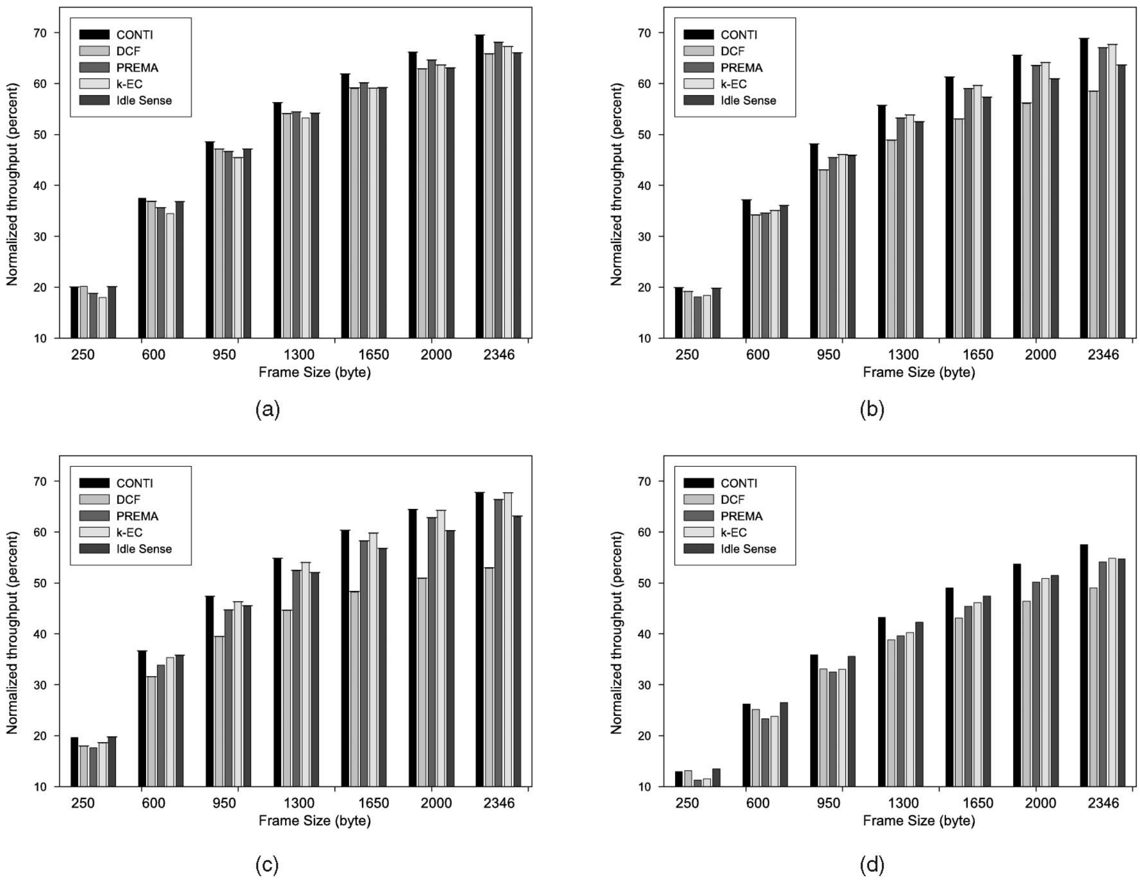

Fig. 5. Throughput with IEEE 802.11b&g. (a) Throughput of five stations with IEEE 802.11b. (b) Throughput of 20 stations with IEEE 802.11b. (c) Throughput of 50 stations with IEEE 802.11b. (d) Throughput of 20 stations with IEEE 802.11g.

error bars along with the long simulation time validate the higher performance of CONTI, although by a small margin. More importantly, Fig. 5a shows that DCF has a similar performance to the other schemes for the low number of stations. This, however, is not the case with larger number of stations where DCF's performance starts to slip.

- We also notice that for the small frame size, PREMA and k-EC are behind DCF and Idle Sense in Figs. 5a and 5b. In other words, the jamming-based schemes are behind the CW-based schemes (except for CONTI). This is because the advantages of PREMA and k-EC are their small collision rates. However, this advantage is not effective when the frame size is small. When the frame size is large, PREMA and k-EC surpass DCF and CONTI since these latter two's collision rate will waste a significant portion of time.
- Finally, we make the following observation on the throughputs of PREMA and k-EC. In Fig. 5a, PREMA has a higher throughput. However, in Figs. 5b and 5c, k-EC's throughput surpasses PREMA's. This trend happens since k-EC requires less slots with more number of stations due to its earliest-jam-prevail policy.

7.5 Throughput with IEEE 802.11g

We also show the throughput evaluation with the IEEE 802.11g physical layer in Fig. 5d. We use the maximum data rate supported by 802.11g, which is 54 Mbps. The control rate we use is 2 Mbps, since this rate is mandatory to be supported in 802.11g. The simulation is also run for 1,200 seconds. In 802.11g, even more frames will be transmitted in this time since the transmission time is shorter, and this will ensure stable results as well.

In this part, we use the following physical layer parameters, and not the ones in Table 6. The slot time is $9\ \mu$s, SIFS time is $10\ \mu$s, DIFS time is $28\ \mu$s, CWmin is 15, CWmax is 1,023, and the PLCP overhead is $41.6\ \mu$s. However, we continue using 802.11b after this part for the delay and fairness measurements.

First off, since all of the four parts in Fig. 5 are on the same scale, we notice that the normalized throughput of 802.11g is lower. This is the normalized throughput percent, however, and 802.11g still has higher throughput in Mbps due to its higher rate. This trend happens because in this simulation there is a large gap between the control rate of 2 Mbps and the data rate of 54 Mbps. As a matter of fact, 802.11g is more sensitive to overhead since the opportunity to transmit a lot of data would be lost due to higher rates [27]. From Fig. 5d, however, we observe that CONTI maintains a slightly higher throughput than the other schemes for the frame sizes of 600 bytes and larger.

7.6 Delay

In this part we evaluate the delay of the schemes. The delay is defined as the time spent from when the frame arrives at

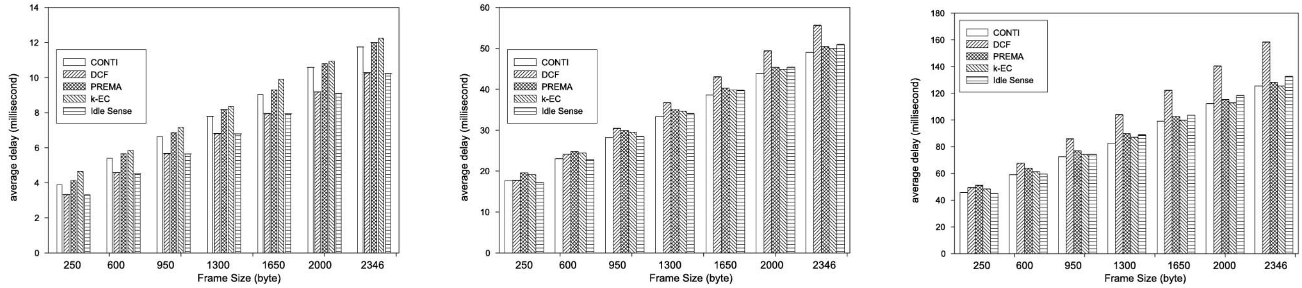

Fig. 6. Delay with IEEE 802.11b. (a) Delay of five stations. (b) Delay of 20 stations. (c) Delay of 50 stations.

the station's head of the queue to the time where it's transmitted successfully. Similar to above, the experiment to measure the delay has 1,200 seconds of simulation time, 11 Mbps of data rate, and 1 Mbps of control rate. The number of stations and the frame size are varied and the results are shown in Fig. 6. The figure also shows the error bars, although it's not well visible in most of the cases since it's small. The error was typically between 0.01 and 0.45. Every result was obtained based on 10 runs of the simulations with 1,200 seconds for each run.

With five stations in Fig. 6a, DCF and Idle Sense have the smallest delay. This is because they have the combination of a low collision rate with five stations, and also using a small number of slots. However, when there are more stations, in Figs. 6b and 6c, DCF's delay rises significantly and Idle Sense doesn't have an advantage anymore over PREMA and k-EC. For 20 and 50 stations, CONTI has a small advantage in the delay values over the other schemes.

7.7 Fairness

In this part, we evaluate the fairness of the MAC schemes. We use Jain's fairness index [28] for this measurement. To evaluate this index, we consider the number of active stations to be n, and the proportions of the transmitted frames with respect to the total number of successfully transmitted frames by each station is given by $x_1, x_2, \ldots, x_n$, respectively. The Jain's index is then given by the following:

$$f(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n.\sum_{i=1}^{n} x_i^2}. \tag{21}$$

The value of this index ranges from 0 to 1. A value that is close to zero means a low fairness between the users, while a value approaching 1 designates a high fairness. We investigate several factors that affect the fairness: the time-scale (to understand short-term fairness and long-term fairness), the frame size, and the number of the stations.

7.7.1 Short-Term versus Long-Term Fairness

To evaluate the short-term fairness, we implement the sliding window mechanism that is presented in [29]. This mechanism requires a transmission trace, which is a set of the stations' IDs, in the order in which they transmitted. First, we consider a window starting at the first element of the trace and of length w. We find Jain's index on this window. Then we slide the window (still of length w) by one element to the right and we find Jain's index again. We keep sliding the window until the right side of the window reaches the last element in the trace. We average all the index values on all the windows. This would be the average value associated with the window size w. Then, we plot the average index against the values of w.

In this scenario, we have 20 stations that are transmitting. The trace size is 5,000. We vary the window size from 20 up to 1,000 as we measure the averaged Jain's index. The results are shown in Fig. 7. Apparently from the figure, DCF has the lowest fairness among all the window sizes. CONTI, PREMA, and k-EC have a comparable fairness, while Idle Sense has a slightly higher fairness. As the time evolves into long-term, all of the schemes' fairness approaches 1. However, the difference is in the speed of convergence which is the short-term fairness.

One reason why Idle Sense has a better fairness than CONTI, PREMA, and k-EC is that Idle Sense require each station to select one random number per access. On the other hand, CONTI, PREMA, and k-EC, even though they give the same parameters to all the stations, they require the selection of multiple random numbers in one access. That might introduce more randomness in the distribution of access than with Idle Sense.

7.7.2 Fairness versus Frame Size

We measure the fairness when the frame size changes. The results are shown in Fig. 8. We observe that there is a trend that happens on a specific short-term basis. For CONTI, PREMA, k-EC, and Idle Sense, this short term is around 0.5 seconds. At this term basis, smaller frame sizes give a higher fairness. This is because there would be

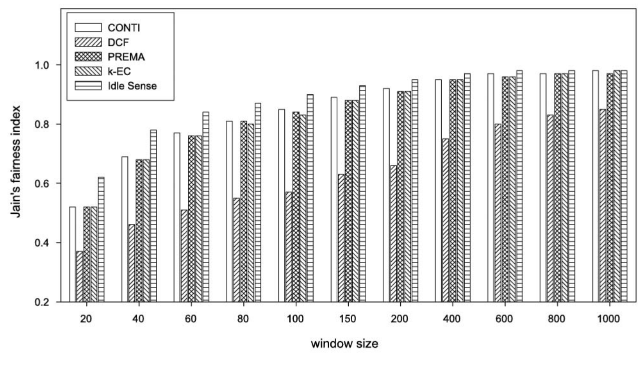

Fig. 7. Averaged Jain's index over sliding window.

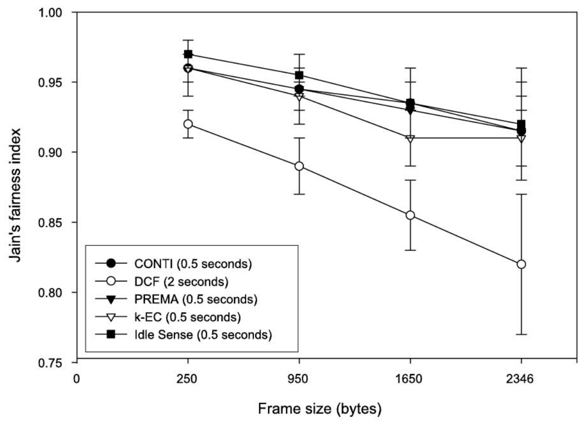

Fig. 8. Fairness versus srame size.

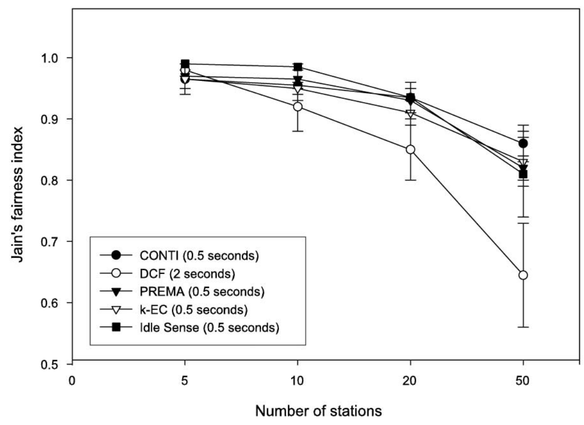

Fig. 9. Fairness versus number of stations.

more frames transmitted with the smaller frame size; hence, fairness will increase in accordance with the result of Fig. 7. At the simulation smaller than 0.5 seconds, the error bars were too large to draw any conclusion. At the simulation time that's larger than 0.5 seconds, this curve becomes too flat so that the trend disappears. It seems every scheme has a short-term interval where this trend would be observed.

For CONTI, PREMA, k-EC, and Idle Sense, this short term was 0.5 seconds. For DCF, this short term was 2 seconds, hence its result in Fig. 8 are for 2 seconds of simulation time. All of the results are based on error from 10 simulation runs.

7.7.3 Number of Stations

We measure the fairness when the number of stations changes. Similarly to the previous experiment, this effect occurs on a time scale that is around 0.5 seconds for CONTI, PREMA, k-EC, and Idle Sense, and at 2 seconds for DCF.

In this experiment, the frame size is 1,500 bytes. The result is shown in Fig. 9. Every value shows the error bar based on 10 simulation runs. The observation from this experiment is the following.

When there are more stations, it's more likely that unfairness will happen since there are more possibilities of the distribution of successful frames among the stations. This is the trend that's observed in the short term. As the simulation time increases, this trends diminishes.

Finally, we conclude the following from the fairness measurement:

1. In longer simulation time, more fairness is observed, unlike the short term where there's more difference between the schemes.
2. The frame size affects the fairness in the short term only, where the small frame size gives more fairness. This disappears in the long term.
3. The number of stations affects the fairness in the short term only, where less stations give higher fairness. This observation diminishes in the long term.

8 Conclusion

This paper presented a comparison of MAC schemes for wireless LANs. Our scheme, which attempts to resolve the contention in a constant-time (CONTI), was compared to other schemes, namely, DCF, PREMA, k-EC, and Idle Sense. First, we reviewed the related work and described the operations of a few schemes that we compare against. Then, we presented the details of CONTI and obtained its optimal parameters in two ways: an algorithmic approach and an optimization approach. Following, we presented an analysis that shows the effect of the contention slot on the throughput of CONTI. Finally, in the simulation results, we compared the performance of CONTI to other schemes. From the experiments, we started by showing the number of slots in a contention and the collision rate. Then, we showed the throughput where CONTI provided a small advantage over the schemes in the majority of the cases. We also showed the delay where DCF and Idle Sense provided the lowest delay for the small number of stations, equal to 5. However, CONTI provided the lowest delay for the medium and large size network, with 20 and 50 stations, respectively. Finally, we made a fairness comparison that showed that Idle Sense has the highest fairness, followed equally by CONTI, PREMA, and k-EC, then finally by DCF. In the last part, we showed that a smaller frame size and a smaller number of stations increase the fairness, but this trend happens temporarily on a short-term basis.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments. Especially, the authors believe that the addition of Section 5, which was made in response to the reviews, improves the quality of the paper.

References

[1] *IEEE Std 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE, 1999.
[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 3, pp. 535-547, Mar. 2000.
[3] E. Ziouva and T. Antonakopoulos, "CSMA/CA Performance under High Traffic Conditions: Throughput and Delay Analysis," *Computer Comm.,* vol. 25, no. 3, pp. 313-321, 2002.
[4] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs," *Proc. IEEE INFOCOM,* 2005.
[5] M. Carvalho and J. Garcia-Luna-Aceves, "Delay Analysis of IEEE 802.11 in Single-Hop Networks," *Proc. 11th IEEE Int'l Conf. Network Protocols (ICNP),* 2003.

[6] N.H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," *Proc. ACM MobiCom*, Aug. 2000.
[7] P. Jacquet, P. Minet, P. Muhlethaler, and N. Rivierre, "Priority and Collision Detection with Active Signaling—The Channel Access Mechanism of HIPERLAN," *Wireless Personal Comm.*, vol. 4, no. 1, pp. 11-26, Jan. 1997.
[8] J.L. Sobrinho and A.S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks," *IEEE J. Selected Areas in Comm.*, vol. 17, no. 8, pp. 1353-1368, Aug. 1999.
[9] G. Wikstrand, T. Nilsson, and M. Dougherty, "Prioritized Repeated Eliminations Multiple Access: A Novel Protocol for Wireless Networks," *Proc. IEEE INFOCOM*, pp. 1561-1569, Apr. 2008.
[10] B. Zhou, A. Marshall, and T.-H. Lee, "A k-Round Elimination Contention Scheme for WLANs," *IEEE Trans. Mobile Computing*, vol. 6, no. 11, pp. 1230-1244, Nov. 2007.
[11] Z. Abichar and J.M. Chang, "CONTI: Constant-Time Contention Resolution for WLAN Access," *Proc. Fourth Int'l IFIP-TC6 Networking Conf.*, 2005.
[12] 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," *Proc. ACM SIGCOMM*, 2005.
[13] Y. Grunenberger, M. Heusse, F. Rousseau, and A. Duda, "Experience with an Implementation of the Idle Sense Wireless Access Method," *Proc. ACM Int'l Conf. Emerging Networking Experiments and Technologies (CoNEXT '07)*, pp. 1-12, Dec. 2007.
[14] H. Wu, A. Utgikar, and N. Tzeng, "SYN-MAC: A Distributed Medium Access Control Protocol for Synchronized Wireless Networks," *Mobile Networks and Applcations*, vol. 10, no. 5, pp. 627-637, Oct. 2005.
[15] T. You, C. Yeh, and H. Hassanein, "A New Class of Collision Prevention MAC Protocols for Wireless Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC)*, 2003.
[16] C. Yeh and T. You, "A QoS MAC Protocol for Differentiated Service in Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Parallel Processing (ICPP)*, 2003.
[17] J. Stine, G. DeVeciana, K. Grace, and R. Durst, "Orchestrating Spatial Reuse in Wireless Ad Hoc Networks Using Synchronous Collision Resolution (SCR)," *J. Interconnection Networks*, vol. 3, nos. 3/4, pp. 167-195, Sept.-Dec. 2002.
[18] J. Galtier, "Analysis and Optimization of MAC with Constant Size Congestion Window for WLAN," *Proc. Second Int'l Conf. Systems and Networks Comm. (ICSNC '07)*, 2007.
[19] L. Bononi, M. Conti, and E. Gregori, "Runtime Optimization of IEEE 802.11 Wireless LANs Performance," *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 1, pp. 66-80, Jan. 2004.
[20] K.C. Tay, K. Jamieson, and H. Balakrishnan, "Collision-Minimizing CSMA and Its Applications to Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.*, vol. 22, no. 6, pp. 1048-1057, Aug. 2004.
[21] Q. Ni, I. Aad, C. Barakat, and T. Turletti, "Modeling and Analysis of Slow CW Decrease for IEEE 802.11 WLAN," *Proc. 14th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC)*, 2003.
[22] H. Wu, S. Cheng, Y. Peng, K. Long, and J. Ma, "IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement," *Proc. IEEE Int'l Conf. Comm. (ICC)*, 2002.
[23] F. Cali, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Networking*, vol. 6, no. 8, pp. 785-799, Dec. 2000.
[24] K. Jamieson, H. Balakrishnan, and Y.C. Tay, "Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks," *Proc. Third European Workshop Wireless Sensor Networks (EWSN)*, 2006.
[25] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, second ed. MIT Press, 2001.
[26] *IEEE Std 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE, 1999.
[27] Z. Abichar, J.M. Chang, and D. Qiao, "Group-Based Medium Access for Next-Generation Wireless LANs," *Proc. Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WOWMOM '06)*, pp. 35-41, 2006.
[28] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.
[29] C.E. Koksal, H. Kassab, and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," *Proc. ACM SIGMETRICS*, June 2000.

Zakhia Abichar received the BS degree (with distinction) in computer engineering from Iowa State University in August 2003. He is a PhD candidate in the Department of Electrical and Computer Engineering at Iowa State University. His research interests are in wireless networks and mobile computing. He has been doing research on IEEE 802.11 Wireless LAN networks, IEEE 802.16 WiMAX Broadband Wireless networks, and Wireless Vehicular networks. He is a student member of the IEEE.

J. Morris Chang received the PhD degree in computer engineering from North Carolina State University. He is an associate professor at Iowa State University. His industrial experience includes positions at Texas Instruments, Microelectronic Center of North Carolina and AT&T Bell Laboratories. He received the University Excellence in Teaching Award at Illinois Institute of Technology in 1999. His research interests include wireless networks, performance study of Java virtual machines (JVM), and computer architecture. Currently, he is a handling editor of the *Journal of Microprocessors and Microsystems* and the middleware & wireless networks subject area editor of *IEEE IT Professional*. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.