

RESEARCH

Open Access

Dynamic tuning of the IEEE 802.11 distributed coordination function to derive a theoretical throughput limit

Yi-Hung Huang^{1*} and Chao-Yu Kuo²

Abstract

IEEE 802.11 is the most popular and widely used standard for wireless local area network communication. It has attracted countless numbers of studies devoted to improving the performance of the standard in many ways. In this article, we performed theoretical analyses for providing a solution to the maximum throughput problem for the IEEE 802.11 distributed coordination function, and an algorithm using a binary cubic equation for obtaining a much closer approximation of the optimal solution than previous algorithms. Moreover, by studying and analyzing the characteristics of the proposed algorithm, we found that the effects of backoff counter consecutive freeze process could be neglected or even disregarded. Using the NS2 network simulator, we not only showed that the proposed theoretical analysis complied with the simulated results, but also verified that the proposed approach outperformed others in achieving a much closer approximation to the optimal solution.

Keywords: IEEE 802.11, distributed coordination function, performance analysis

1. Introduction

Advances in wireless communication technology have increased the demand for wireless networks. The IEEE 802.11 standard [1] defines the specifications for medium access control (MAC) and the physical layers in a wireless local area network (WLAN). The IEEE 802.11 standard provides two mechanisms for the MAC protocol: the point coordination function (PCF) and the distributed coordination function (DCF). The PCF utilizes a basic access mechanism that supports contention-free services. Therefore, the PCF requires a base station that coordinates channel access among nodes. On the other hand, the DCF utilizes an access mechanism that supports contention-based services. The DCF access mechanism dictates that all the nodes should randomly access channels using the carrier sense multiple access/collision avoidance (CSMA/CA) mechanism. This mechanism employs the acknowledgment (ACK) feature to detect transmission failures. In other words, if an ACK response is not received, it is assumed that packet transmission has

failed. The nodes wait for an interframe space (IFS), and then invoke the binary exponential Backoff algorithm, which uses a uniform random distribution called a contention window (CW) to generate a random Backoff value within the range of $[0, CW - 1]$.

In this study, the initial value of CW is set to CW_{\min} (the minimum CW). Subsequently, the CW value is doubled when packet transmission fails. For a node to obtain a Backoff value, it must first determine whether the channel is in use. If the channel is not busy, then the Backoff value decreases by 1 in every time slot and the node transmits the data when the Backoff value reaches zero. However, if the channel is busy, then the Backoff counter freezes. When the channel is in an idle state, it waits for a DCF IFS (DIFS) time period after which the Backoff value begins to decrease again. If the packet transmission continues to fail, then the CW value increases to CW_{\max} (the maximum CW); when the node receives an ACK packet, CW is reset to CW_{\min} . If a node receives an error packet, it must wait for an extended IFS (EIFS) time period. Then, the node determines again whether the channel is in an idle state. If it is, then after a DIFS time, the Backoff value decreases by 1 after each idle slot.

* Correspondence: ehhwang@mail.ntcu.edu.tw

¹Department of Mathematics Education, National Taichung University of Education, Taichung 40306, Taiwan

Full list of author information is available at the end of the article

Currently, the methods of improving Backoff performance can be divided into two categories: (1) adjusting the CW size according to the number of times that collisions have occurred [1-3], and (2) dynamically adjusting the CW size by detecting changes in the network environment [4-7]. In the first type of method, the adjustment of CW size only occurs after a collision; the consequences are that the cost for collision must first be paid before the method can find the most appropriate CW size, and that this entire process is repeated when the data are transferred successfully. In contrast, the second type of method immediately adjusts to the most appropriate CW size when network environment changes are detected. Therefore, such a method has the ability to find the appropriate CW size without the cost of collision, and clearly outperforms the first type of method in many ways. For the reasons mentioned above, this article proposed an algorithm called the dynamic contention window (DCW) algorithm by adopting the second approach. Unlike other algorithms, DCW uses a binary cubic equation that has the ability to quickly and efficiently calculate the most appropriate CW size according to the network environment.

The rest of this article is organized as follows. Previous related work is presented in Section 2. Various theoretical analyses are performed in Section 3. The proposed DCW algorithm and consecutive freeze process (CFP) analysis are presented in Section 4. Simulations and performance evaluations of our proposed algorithm are conducted in Section 5. Finally, we conclude our work in Section 6

2. Related work

To the best of our knowledge, most studies on performance analysis of IEEE 802.11 MAC protocols use the results presented in [4] for their theories or discussions [8-13].

The following equation is derived from the analytic work in [4]:

$$E[\text{Coll}] \times E[N_c] = (E[N_c] + 1) \times E[\text{Idle}] \quad (1)$$

Here, the $E[\text{Coll}]$ is the expected value of collision time; $E[N_c]$ is the expected value of the number of nodes that are involved in a collision; and $E[\text{Idle}]$ is the expected value of idle time. As long as the actual values for $E[\text{Coll}]$ and $E[N_c]$ can be measured and substituted into (1), then the value of $E[\text{Idle}]$ can be solved.

$$E[\text{Idle}] = \frac{(1-p)^M}{1-(1-p)^M} \times t_{\text{slot}} \quad (2)$$

where M is the total number of nodes in the network, the t_{slot} the total duration spent in a time slot, and p is

the Backoff value sampled from a geometric distribution with parameter p . First, we use (1) to solve $E[\text{Idle}]$, which we then substitute into (2) to solve p . Using this value of p , we can derive the value of parameter p for the geometric distribution from the Backoff value of maximum throughput.

When solving (1) and (2), an optimal solution cannot be solved directly; instead, a numerical method is needed to approximate the optimal solution. Therefore, the effectiveness of this method is fully dependent on how fast the numerical method can find the approximation of the optimal solution. Additionally, ref. [4] assumes that the values of $E[\text{Coll}]$ and $E[N_c]$ in (1) and (2) can be derived by measuring the network condition. Unfortunately, this is not entirely true in practice. There is no collision detection capability due to the characteristics of the wireless networks.

Based on the solutions of (1) and (2) and by observing the solution while solving the value of p , the values of $E[N_c]$ and $E[\text{Coll}]$ mostly remain constant [4]. Therefore, (1) can be further simplified as follows:

$$E[\text{Coll}] = \Phi(\text{Idle}, N_c) = \frac{(E[N_c] + 1) \cdot E[\text{Idle}] \cdot t_{\text{slot}}}{E[N_c]} \quad (3)$$

where $\Phi(\text{Idle}, N_c)$ is a constant. Although (3) can be used to replace (1), a numerical method is still needed for this equation to approximate the optimal solution. The performance of this approach is fully dependent on the efficiency of the numerical method when finding the approximation of the optimal solution. Therefore, to save the time consumed by the numerical method, we propose a binary cubic equation with the ability to obtain a much closer approximation of the optimal solution in less time.

3. Analysis of proposed method

In this study, we assume that (1) each node is in a saturated condition (i.e., always having a packet to transmit) and (2) the channel is error-free. Packet loss is caused solely by collisions in the process of packet transmission. The hidden terminal problem is not considered in this article.

3.1. Analysis of collision probability

First, we divided the timeline into discrete time slots, where the probability of transmission for each time slot is equal to τ , in accord with [14,15]. Therefore, $\tau = 2/E[\text{CW}]$, where $E[\text{CW}]$ is the expected value of the CW. Suppose that there are M nodes in the network, where τ_x ($x = 1, 2, \dots, M$) is the probability of transmission for node x in each time slot, ACK_x ($x = 1, 2, \dots, M$) is the number of ACK packets successfully received by each node, and Coll_x ($x = 1, 2, \dots, M$) is the number of packets

that do not receive ACK. The collision probability can then be defined as follows:

$$\text{Collision probability} = 1 - \frac{\sum_{x=1}^M \text{ACK}_x}{\sum_{x=1}^M (\text{ACK}_x + \text{Coll}_x)} \quad (4)$$

Each time slot can be classified into three states: idle (no data transmission), successfully transmitted, and collision. Therefore, the probability of each state can be calculated as follows.

The probability of an idle time slot is calculated as (5) and referred to as P_i :

$$\prod_{x=1}^M (1 - \tau_x) = P_i \quad (5)$$

The probability of a successfully transmitted time slot is calculated as (6) and referred to as P_s :

$$\sum_{x=1}^M \left(\tau_x \prod_{y=1, y \neq x}^M (1 - \tau_y) \right) = P_s \quad (6)$$

The probability of a collision time slot is calculated as (7) and referred to as P_c :

$$P_c = 1 - P_i - P_s \quad (7)$$

Because collision only occurs when there are at least two nodes simultaneously transmitting data in a single time slot. Therefore, we define k as the average number of nodes involved in a collision, where k can be calculated as follows:

$$k = 2 \times \frac{\sum_{x1 \neq x2} \left(\tau_{x1} \tau_{x2} \prod_{y=1, y \neq x1, y \neq x2}^M (1 - \tau_y) \right)}{1 - P_i - P_s} + 3 \times \frac{\sum_{\substack{x1 \neq x2, \\ x1 \neq x3, \\ x2 \neq x3}} \left(\tau_{x1} \tau_{x2} \tau_{x3} \prod_{y=1, y \neq x1, y \neq x2, y \neq x3}^M (1 - \tau_y) \right)}{1 - P_i - P_s} + \dots + \frac{M \times \prod_{x=1}^M \tau_x}{1 - P_i - P_s} \quad (8)$$

Therefore, the value of Coll_x for all the nodes involved in a simultaneous data transmission is incremented by 1. Hence, when a collision occurs in a time slot, it is necessary to calculate the average number of nodes involved in the collision to facilitate calculation of the collision probability P_c . In addition, the collision probability in (4) can be rewritten using (5)-(8):

$$\begin{aligned} \text{Collision probability} &= 1 - \frac{\sum_{x=1}^M \text{ACK}_x}{\sum_{x=1}^M (\text{ACK}_x + \text{Coll}_x)} \\ &= 1 - \frac{\sum_{x=1}^M \text{ACK}_x / \text{total_slot}}{\sum_{x=1}^M (\text{ACK}_x + \text{Coll}_x) / \text{total_slot}} \\ &= 1 - \frac{P_s}{P_s + P_c \times k} = 1 - \frac{P_s}{P_s + (1 - P_s - P_i) \times k} \end{aligned} \quad (9)$$

By dividing the numerator and denominator, respectively, by the total number of time slots in the network (referred as total_slot), (4) can be represented by the probabilities P_s , P_i , and P_c . In other words, if there are three simultaneous data transmissions, the time slots will collide and each of the three nodes involved in this collision will increase their Coll_x value by 1. However, P_c represents the collision probability that occurs in a time slot. Therefore, when a collision occurs in a time slot, we calculate the value of k , which represents the average number of nodes involved in simultaneous data transmissions during that time. Therefore,

$$P_c \times k = \frac{\sum_{x=1}^M (\text{Coll}_x)}{\text{total_slot}}$$

When the system converges into a stable state, we assume that $\tau_1 = \tau_2 = \tau_3 = \dots = \tau_M = \tau$. Hence, we can rewrite (5), (6), and (7) as follows.

The probability of an idle time slot is calculated as (10) and referred to as P_i :

$$\prod_{x=1}^M (1 - \tau_x) = P_i = (1 - \tau)^M \quad (10)$$

The probability of a successfully transmitted time slot is calculated as (11) and referred to as P_s :

$$\sum_{x=1}^M \left(\tau_x \prod_{y=1, y \neq x}^M (1 - \tau_y) \right) = P_s = M \times \tau \times (1 - \tau)^{M-1} \quad (11)$$

The average number of nodes involved in simultaneous data transmission during a collision is given as follows:

$$\begin{aligned} k &= \frac{\sum_{i=2}^M [i \times \binom{M}{i} (\tau^i \times (1 - \tau)^{M-i})]}{1 - P_i - P_s} \\ &= \frac{\sum_{i=2}^M [i \times \binom{M}{i} (\tau^i \times (1 - \tau)^{M-i})] + M \times \tau \times (1 - \tau)^{M-1} - M \times \tau \times (1 - \tau)^{M-1}}{1 - P_i - P_s} \\ &= \frac{\sum_{i=1}^M [i \times \binom{M}{i} (\tau^i \times (1 - \tau)^{M-i})] - M \times \tau \times (1 - \tau)^{M-1}}{1 - P_i - P_s} \\ &= \frac{M \times \tau - M \times \tau \times (1 - \tau)^{M-1}}{1 - P_i - P_s} \end{aligned} \quad (12)$$

$\sum_{i=1}^M [i \times \binom{M}{i} (\tau^i \times (1 - \tau)^{M-i})]$ is the expected value of the binomial distribution. Therefore, it is equal to $M \times \tau$. We then substitute (10), (11), and (12) into (9) and derive the collision probability as follows:

$$\begin{aligned} \text{Collision probability} &= 1 - P_s / [P_s + (1 - P_i - P_s) \times k] \\ &= 1 - \frac{M \times \tau \times (1 - \tau)^{M-1}}{M \times \tau \times (1 - \tau)^{M-1} + (1 - P_i - P_s) \times \frac{M \times \tau - M \times \tau \times (1 - \tau)^{M-1}}{1 - P_i - P_s}} \\ &= 1 - (1 - \tau)^{M-1} \end{aligned} \quad (13)$$

In (13), the collision probability can be calculated using the total number of nodes, M , and the probability

of transmission, τ , during a time slot under conditions of a fully utilized throughput environment. The result of (13) also shows that it is easy to calculate and analyze the collision probability.

3.2. Analysis of maximum throughput

According to [14], throughput is defined as follows:

$$\text{throughput} = \frac{P_s \times \text{payload}}{P_i \times t_{\text{slot}} + P_s \times t_{\text{success}} + P_c \times t_{\text{coll}}} \quad (14)$$

where payload is the time spent to transmit data, t_{slot} is an idle slot time (aSlotTime), and t_{success} is the time spent to transmit a packet successfully. Notably, $t_{\text{success}} = \text{DATA} + \text{SIFS} + \text{ACK} + \text{DIFS}$ when the algorithm does not utilize the RTS/CTS method. Furthermore, t_{coll} is the time spent during packet collision, and $t_{\text{coll}} = \text{DATA}_{\text{max}} + \text{DIFS}$ when the algorithm does not utilize the RTS/CTS method. DATA_{max} is the maximum waiting time when a packet collision occurs. Most other studies have assumed that DATA_{max} is equal to DATA. However, according to the IEEE 802.11 standard presented in [1], nodes involved in a collision must wait for one more EIFS in addition to DATA. Hence, this article assumes that $\text{DATA}_{\text{max}} = \text{DATA} + \text{EIFS} \times (M - k)/M$, where M is the total number of nodes in the network and k is the average number of nodes involved in simultaneous data transmissions. This assumption makes it clear that there are k nodes on an average that are busy transmitting data, and that the transmission nodes are unable to receive any other packets from other nodes. Therefore, because the transmission nodes need not wait for another EIFS, the number of transmission nodes must be deducted from the equation.

By substituting (7), (10), and (11) into (14) and simplifying, we get the following:

$$\text{throughput} = \frac{M \times \tau \times \text{payload}}{\frac{t_{\text{coll}}}{(1-\tau)^{M-1}} + (1-\tau) \times (t_{\text{slot}} - t_{\text{coll}}) + M \times \tau \times (t_{\text{success}} - t_{\text{coll}})} \quad (15)$$

To solve the maximum throughput, we differentiate τ in (15) (as shown in Appendix A). To reduce the complexity of solving the maximum throughput, we assume that t_{coll} is a constant. This gives us the equation below:

$$\frac{1 - M \times \tau}{(1 - \tau)^M} = 1 - \frac{t_{\text{slot}}}{t_{\text{coll}}} \quad (16)$$

The right-hand side of (16) assumes a value between 0 and 1 because $t_{\text{coll}} > t_{\text{slot}}$. The left-hand side of (16) is equal to 1 when $\tau = 0$; however, it is 0 if $\tau = 1/M$. When $0 < \tau < 1/M$, the left-hand side becomes a decreasing function that varies between 1 and 0. Therefore, the optimal solution for τ can be obtained.

The Abel-Ruffini theorem (also known as Abel's impossibility theorem) [16] states that there is no general algebraic solution to polynomial equations of the fifth degree or higher. For this reason, an algebraic solution is impossible with (16) when M is greater than 5. Therefore, with the network parameters provided in Table 1 we adopt a numerical method to solve (16) while all nodes in the network are transmitting constant length data packets, and observe the relation between the CW ($2/\tau$) and M .

In Figure 1, the x -axis represents the number of nodes, the y -axis represents the CW ($2/\tau$), and the numbers 500, 1500, and 2312 represent constant packet sizes in bytes. Figure 1 clearly shows that the relationship between the CW ($2/\tau$) and M is linear (other packet size have the same linear relationship). Therefore, we conducted regression analyses to solve the relationship between the CW ($2/\tau$) and M , the results of which are shown in Table 2. (The results of different packet sizes are illustrated in Appendix C.)

In Table 2 the value of R^2 is almost equal to 1. This verifies that the result of the regression analysis is almost consistent with the solution of (16).

The results of Appendix C indicate that different packet sizes lead to different results for regression analysis. Figure 2 shows the relationship of the packet size to the first-degree coefficient and the constant in Appendix C.

In Figure 2, the x -axis represents the length of the packet, and the y -axis represents the first-degree coefficient and the constant. The curves for the coefficient of M and constant represent the first-degree coefficients and the constant, respectively. The results of Figure 2 clearly show that the packet length is linearly related to both the first-degree coefficient and the constant term. Therefore, by applying quadratic regression analysis to Figure 2, we obtained the coefficients of determination for the coefficients of M and the constant term in Table 3.

As shown in the table, the R^2 values are both greater than 0.975. This implies that the quadratic fit to the results in Figure 2 is good. In order to obtain the CW size of maximum throughput, we must first substitute

Table 1 WLAN parameters

DataRate	11 Mbps
PCLPDataRate	1 Mbps
BasicRate	1 Mbps
Slot time	20 μ s
SIFS	10 μ s
DIFS	50 μ s
EIFS	SIFS + DIFS + (ACK length)/basic rate
PHYHeader	192 bits
MACHeader	224 bits
ACK length	112 bits + PHYHeader

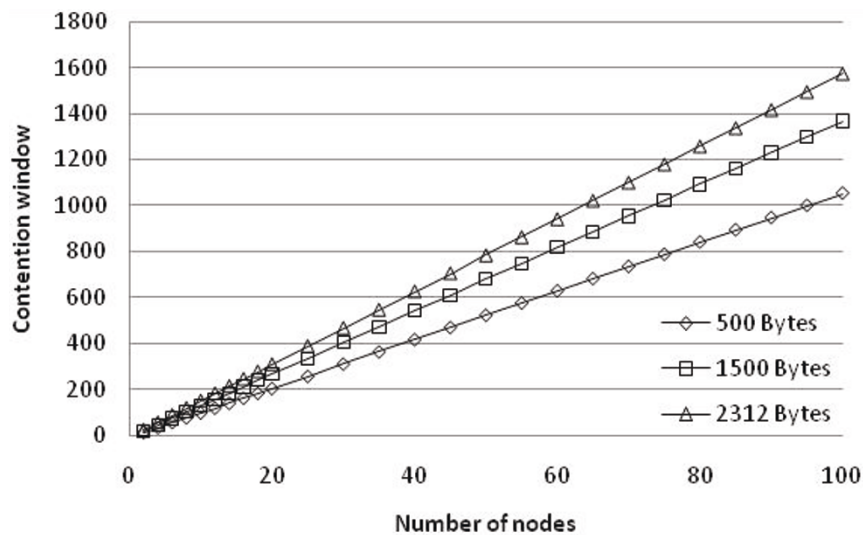


Figure 1 CW as a function of the number of nodes.

the packet length into Table 3 to solve the number of nodes and the coefficients of the linear equation of the CW, and then substitute the number of nodes into the target linear equation. We thereby obtain the CW size of maximum throughput. Therefore, in the next section, we propose the DCW algorithm.

4. DCW algorithm

In order to achieve maximum throughput with the analyses from Appendix C and Table 3 with the network environment parameters presented in Table 1 we combined the equations from Appendix C and Table 3 into (17). Here, the size of the CW is strongly related to the number of nodes (M) and the packet length (X).

$$\begin{aligned}
 \text{CW} = & \\
 & (-3.71095 \times 10^{-7}X^2 + 3.9512 \times 10^{-3}X + 8.6886)M \quad (17) \\
 & + 1.32129 \times 10^{-7}X^2 + 4.1818 \times 10^{-4}X + 7.8933
 \end{aligned}$$

In (17), we only need to substitute the packet length into X and the number of nodes into M to obtain the CW size of maximum throughput.

The proposed DCW algorithm (DCW) is shown in Figure 3.

When transmitting data using DCW, it is called upon to obtain the Backoff value regardless of the success or failure (data retransmission required) of data

transmission. After each aSlotTime, the Backoff Time reduces an aSlotTime, and if the medium is busy during an aSlotTime, the reduction of an aSlotTime from Backoff Time is stopped until the medium is once again idle. When the medium becomes idle, it must wait for a DIFS time before continuing the countdown of the Backoff Time. The countdown goes on until the Backoff Time reaches 0, after which data transmission begins.

There have already been many methods proposed by other authors to estimate the number of nodes in the network [17,18], by applying these methods in the proposed DCW. DCW is then suitable for network environments in which the number of nodes changes dynamically.

4.1. Backoff counter CFP

In IEEE 802.11, when a node completes its data transmission, it obtains another Backoff value via the Backoff procedure before it starts another round of data transmission. If the Backoff value obtained from the Backoff procedure is 0, it indicates that the node that has a Backoff value of 0 may transmit data packets immediately without reducing its value, while it also indicates that other nodes may not reduce their Backoff values. However, a transiting node that consecutively obtains a Backoff value of 0 will result in all other nodes to freeze their reduction in the Backoff procedure. This type of phenomenon is referred to as a Backoff Counter CFP [15,19].

The occurrence probability of CFP is determined by the number of nodes and the CW size. There are two conditions in CFP. First, if a node successfully transmits its data packets, the CFP occurrence probability is 1/

Table 2 Regression analysis

Packet size (bytes)	Regression analysis ($2/\tau$)	Coefficient of determination (R^2)
500	$10.6M-8.0068$	1
1500	$13.762M-8.9413$	1
2312	$15.847M-9.3857$	1

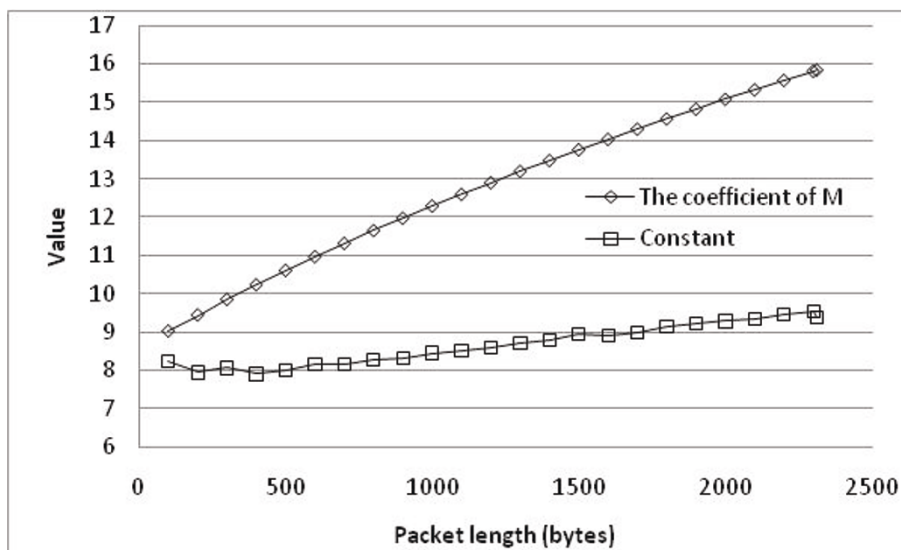


Figure 2 Coefficient of M and constant as a function of packet length.

CW because only one node may obtain a Backoff value of 0 (condition (1)). Second, if multiple nodes transmit data packets simultaneously, the average CFP occurrence probability is $1 - (1 - 1/CW)^k$ (when a collision occurs, k is the average number of nodes that participates in the collision) (condition (2)). This is because there are k nodes on average trying to transmit data packets simultaneously, and hence k nodes on average may obtain a Backoff value of 0.

The DCW did not take the phenomenon of CFP into consideration during the analysis process. This is because the occurrence probability of CFP is very low in DCW; in particular, when the number of nodes increases, the occurrence probability of CFP declines. In DCW, $CW = C1 * M + C2$, where M is the number of nodes, $C1$ and $C2$ are constants ($8.5 < C1 < 16.3$; $-9.8 < C2 < -7.9$), and the range of packet lengths is 1-2312 bytes. Therefore, when the number of nodes increases, the CW also increases in concert. Yet the occurrence probability of CFP in condition (1) decreases when the number of nodes increases. As for the occurrence probability of CFP in condition (2), the k nodes on average that participate in a collision must be used for the analysis. Thus, we set the value of M in (9) to be a value that approaches infinity, and then check whether k approaches a constant value.

$$\lim_{M \rightarrow \infty} \frac{M \times \tau - M \times \tau \times (1 - \tau)^{M-1}}{1 - P_1 - P_s} = \frac{\frac{2}{C_1} - \frac{2}{C_1} \frac{1}{e^{-C_1}}}{1 - e^{-C_1} - \frac{2}{C_1} \frac{2}{e^{-C_1}}} \quad (18)$$

In (18), e denotes a natural number. The detailed proof is available in Appendix B. From (18), the value of k approaches a constant value as the number of nodes increases.

Figure 4 uses the network environment parameters presented in Table 1; x -axis represents the number of nodes, y -axis represents the k value, and the numbers 500, 1500, and 2312 represent packet sizes in bytes. The results of different packet lengths are presented in Appendix C.

From the results obtained from Appendix C and Figure 4, the range for k is 2-2.081. This is because shorter the data packet length, the larger the value of k . However, it is impossible to have a data packet less than 1 byte. Therefore, the maximum value of k is the same as the data packet length of 1 byte. Therefore, the occurrence probability of CFP in condition (2) is less than $1 - (1 - 1/CW)^{2.81}$, although CW increases as the number of

Table 3 Coefficient of M and constant

	Regression analysis (X : packet length)	Coefficient of determination (R^2)
Coefficient of M	$-3.71095 \times 10^{-7}X^2 + 3.9512 \times 10^{-3}X + 8.6886$	0.9998
Constant	$1.32129 \times 10^{-7}X^2 + 4.1818 \times 10^{-4}X + 7.8933$	0.9751

```

backofftime()
// M: number of nodes
// X: packet length
// C1: the coefficient of M
// C2: the coefficient of constant
// CW: contention window
// aSlotTime: the value of the correspondingly named PHY characteristic


$$C1 = -3.71095 \times 10^{-7} X^2 + 3.9512 \times 10^{-3} X + 8.6886$$


$$C2 = 1.32129 \times 10^{-7} X^2 + 4.1818 \times 10^{-4} X + 7.8933$$


$$CW = C1 \times M + C2$$


// {0, 1, 2, ..., CW-1} Randomly selected integer value
backoff_value = Uniform (CW)

return backoff_value * aSlotTime
    
```

Figure 3 DCW algorithm.

nodes increases where the occurrence probability of CFP in condition (2) becomes increasingly smaller.

From the above analysis on the occurrence probability of CFP, the occurrence probability of CFP is very low when using the DCW method. This also implies that in situations where the number of nodes propagates, the impact of CFP may be neglected.

5. Simulations

5.1. Environmental settings

In this article, we use NS2 [20] as the simulation tool and use the network environment parameters presented in Table 1 as simulation parameters. Each simulation runs for 100 simulated seconds. The simulation uses the normalized throughput indicated in (14) and the

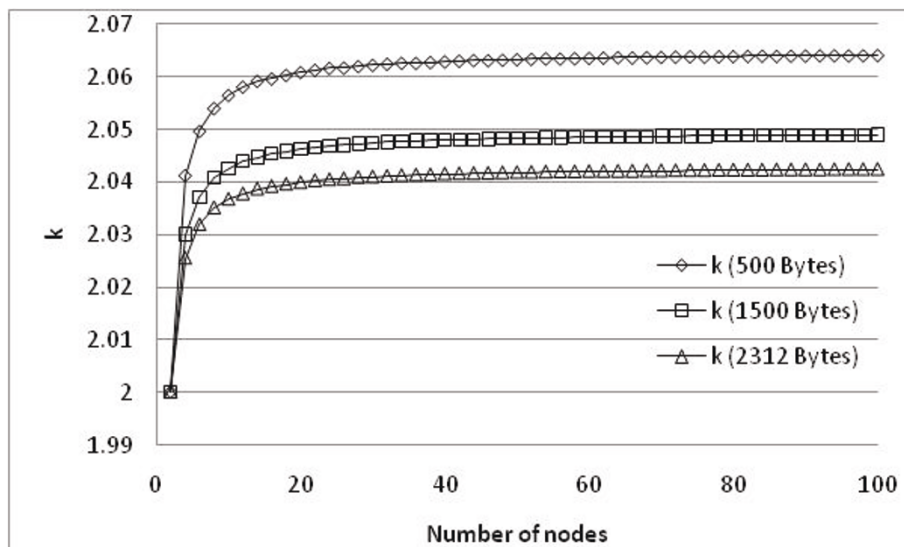


Figure 4 Dependence of k on the number of nodes.

collision probability indicated in (4) as performance indicators.

5.2. Different data packet lengths

We are currently using DCW and different data packet lengths to verify the analyses of (9) and (14).

In Figure 5, the x -axis represents the number of nodes, y -axis represents the collision probability, and the numbers 500, 1500, and 2312 are the numerical results for the corresponding packet sizes in bytes substituted into (9), and 500, 1500, and 2312 are the simulation results for the corresponding packet sizes in bytes. The different data packet lengths show similar results, as shown in Figure 5. CW is an integer, and it is essential to round $2/\tau$ off to an integer. For this reason, the analysis results show non-smooth characteristics. The results of Figure 5 show that the simulated and analytical results are very close.

Using the t distribution and under 99% confidence level, the experiments sampling error for 500, 1500 and 2312 bytes is within $\pm 0.141\%$, $\pm 0.193\%$, and $\pm 0.302\%$, respectively.

In Figure 6, the x -axis represents the number of nodes, y -axis represents the normalized throughput, and the numbers 500, 1500, and 2312 are the numerical results for the corresponding packet sizes in bytes substituted into (14), and 500, 1500, and 2312 are the simulated results using the corresponding packet sizes in bytes. The different data packet lengths show similar results, as shown in Figure 6. CW is an integer, and it is essential to round $2/\tau$ off to an integer. For this reason, the analysis results also show non-smooth characteristics. The results of Figure 6 show that the simulated and analytical results are very close.

Using the t distribution and under 99% confidence level, the experiments sampling error for 500, 1500, and 2312 bytes is within $\pm 0.656\%$, $\pm 0.559\%$, and $\pm 0.733\%$, respectively.

5.3. Comparison between different algorithms

We compare DCW with other algorithms to verify that the DCW is able to provide a relatively close approximation to the maximum throughput. In order to show the differences in IEEE 802.11+ [4] and DCW, the scale for the y -axis in Figure 7a, c, e (DCF versus DCW) as well as Figure 7b, d, f (IEEE 802.11+ versus DCW) is different. IEEE 802.11+ is very close to the maximum throughput; however, by narrowing the distance between the y -axis scale spans, we can clearly see that the DCW provides an even closer approximation to the maximum throughput than IEEE 802.11+.

In Figure 7, the x - and y -axes represent the number of nodes and the collision probability, respectively. The DCF curve uses the standard IEEE 802.11 algorithm, and the curve for IEEE 802.11+ uses the algorithm presented in [4] where the number of nodes is known. Similar results are obtained for different packet lengths, as shown in Figure 7. From these results, the collision probability is lower in DCW than in the other two algorithms. The DCF shows lower collision probability only when the number of nodes is between 2 and 4.

Using the t distribution and under 99% confidence level, the DCF experiments sampling error for 500, 1500, and 2312 bytes is within $\pm 0.134\%$, $\pm 0.265\%$, and $\pm 0.444\%$, respectively, and the IEEE 802.11+ experiments sampling error for 500, 1500, and 2312 bytes is within $\pm 0.154\%$, $\pm 0.248\%$, and $\pm 0.291\%$, respectively.

In Figure 8, the x - and y -axes represent the number of nodes and the normalized throughput, respectively. The

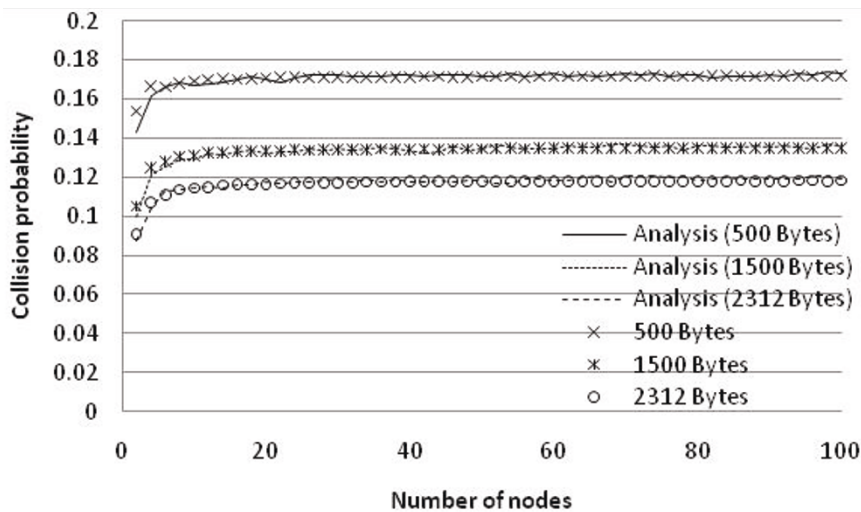


Figure 5 Collision probability.

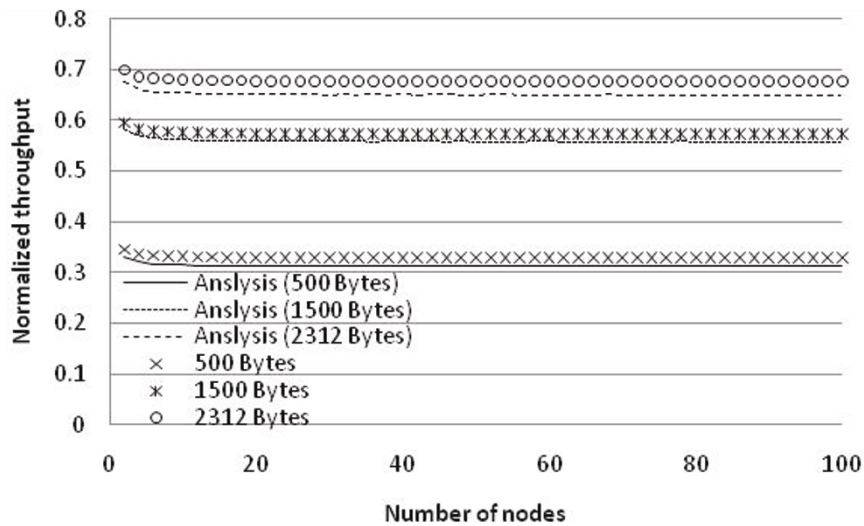
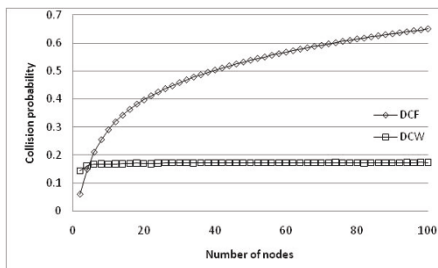
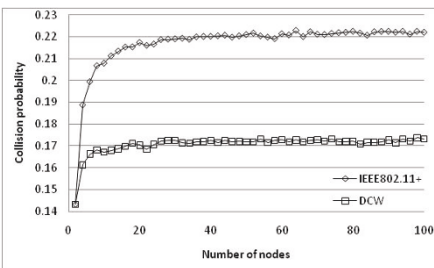


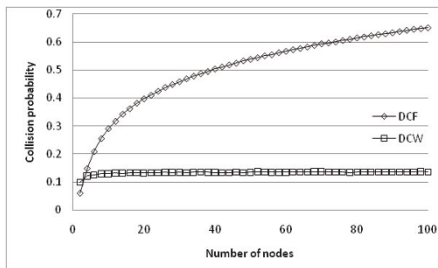
Figure 6 Normalized throughput.



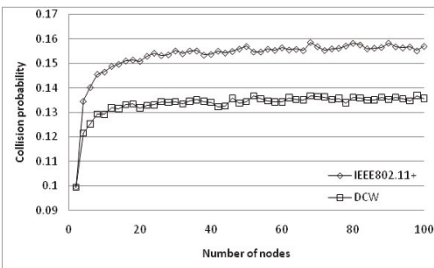
(a) 500 Bytes: DCF vs. DCW



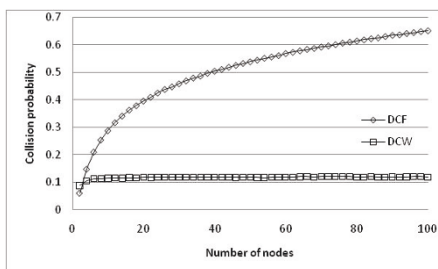
(b) 500 Bytes: IEEE 802.11⁺ vs. DCW



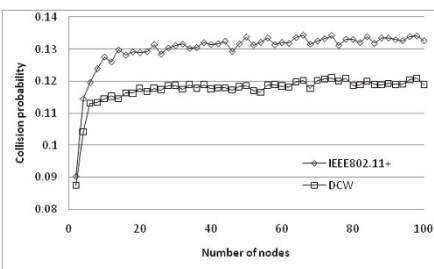
(c) 1500 Bytes: DCF vs. DCW



(d) 1500 Bytes: IEEE 802.11⁺ vs. DCW



(e) 2312 Bytes: DCF vs. DCW



(f) 2312 Bytes: IEEE 802.11⁺ vs. DCW

Figure 7 Collision probability. (a) 500 bytes: DCF versus DCW. (b) 500 bytes: IEEE 802.11⁺ versus DCW. (c) 1500 bytes: DCF versus DCW. (d) 1500 bytes: IEEE 802.11⁺ versus DCW. (e) 2312 bytes: DCF versus DCW. (f) 2312 bytes: IEEE 802.11⁺ versus DCW.

DCF curve uses the standard IEEE 802.11 algorithm, and the curve for IEEE 802.11⁺ uses the algorithm presented in [4] where the number of nodes is known. Similar results are obtained for the different packet lengths, as shown in Figure 8. From these results, the normalized throughput is higher in DCW than in the other two algorithms. Although IEEE 802.11⁺ and DCW show similar results when the packet lengths are 1500 and 2312 bytes, DCW still shows relatively high normalized throughput.

Using the *t* distribution and under 99% confidence level, the DCF experiments sampling error for 500, 1500, and 2312 bytes is within $\pm 0.413\%$, $\pm 0.556\%$, and $\pm 0.768\%$, respectively, and the IEEE 802.11⁺ experiments sampling error for 500, 1500, and 2312 bytes is within $\pm 0.664\%$, $\pm 0.669\%$, and $\pm 0.678\%$, respectively.

6. Conclusions

In this article, we take the influence of EIFS into consideration whereas previous literatures did not. In doing so, we are able to provide analysis results that are much

closer to simulated results. An observation of the results clearly indicates that the influence of EIFS should not be ignored. Moreover, this article also proposes an algorithm that is distinct from others that only use numerical methods. This algorithm is able to find the CW size of maximum throughput immediately by substituting the packet length and number of nodes into a binary cubic equation. From the mathematical analyses provided in this article, it is shown that the influence of CFP is extremely small or even negligible using proposed algorithm.

For studies in the near future, other parameters of network environments can be considered for multidimensional experiments. For example, different values for DataRate can be used for a more realistic wireless network environment.

Appendix A

The inference process of maximum throughput is differentiated by τ in (15).

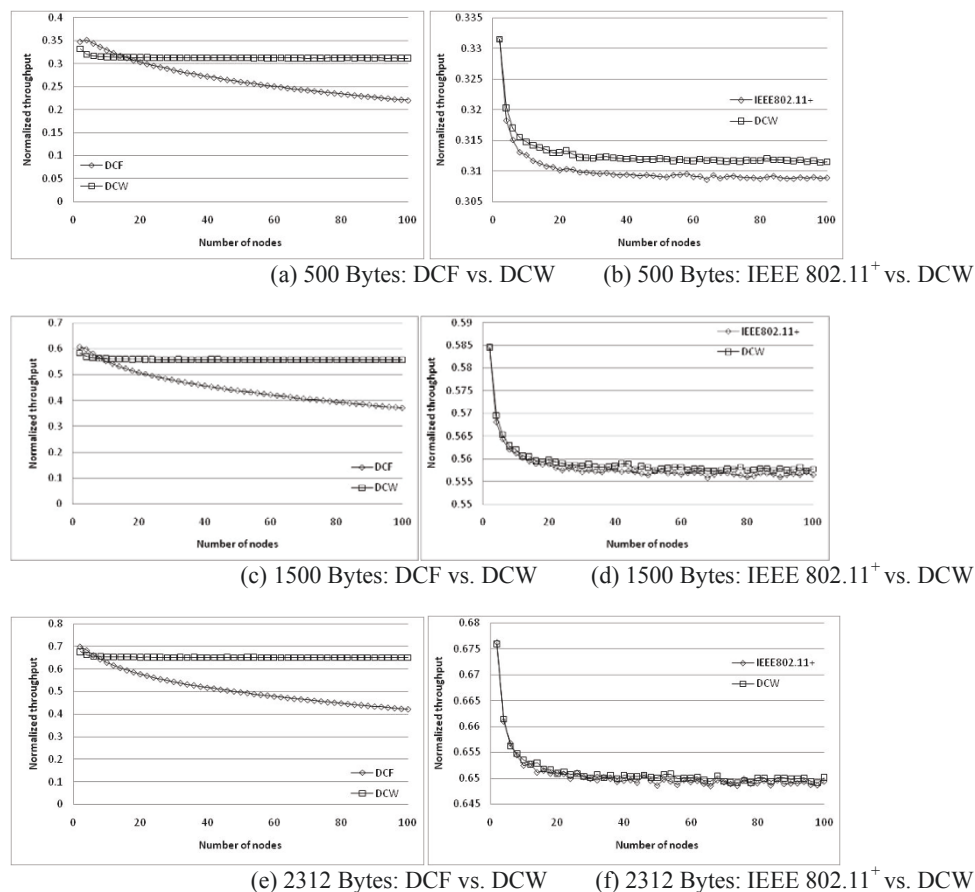


Figure 8 Normalized throughput. (a) 500 bytes: DCF versus DCW. **(b)** 500 bytes: IEEE 802.11⁺ versus DCW. **(c)** 1500 bytes: DCF versus DCW. **(d)** 1500 bytes: IEEE 802.11⁺ versus DCW. **(e)** 2312 bytes: DCF versus DCW. **(f)** 2312 bytes: IEEE 802.11⁺ versus DCW.

Table 4 Regression analysis results and limits of k

Packet size (100-500 bytes)	Regression analysis ($2/\tau$)	Coefficient of determination (R^2)	$\lim_{M \rightarrow \infty} k$
1	8.5684M-8.1523	1	2.080878232
100	9.0133M-8.2238	1	2.076739841
500	10.6M-8.0068	1	2.06489532
1000	12.289M-8.4472	1	2.055736278
1500	13.762M-8.9413	1	2.049627138
2000	15.084M-9.2797	1	2.045182161
2312	15.847M-9.3857	1	2.042961214

\therefore When $\frac{t_{\text{Coll}}}{(1-\tau)^{M-1}} + (t_{\text{slot}} - t_{\text{Coll}}) - \tau \times (M-1) \times \frac{t_{\text{Coll}}}{(1-\tau)^M} = 0$, the throughput assumes an extreme value

$$\therefore \frac{t_{\text{Coll}}}{(1-\tau)^M} ((1-\tau) - \tau \times (M-1)) = t_{\text{Coll}} - t_{\text{slot}}$$

$$\Rightarrow \frac{t_{\text{Coll}}}{(1-\tau)^M} (1 - \tau \times M) = t_{\text{Coll}} - t_{\text{slot}}$$

$$\Rightarrow \frac{(1-\tau \times M)}{(1-\tau)^M} = 1 - \frac{t_{\text{slot}}}{t_{\text{Coll}}} \text{ is proven.}$$

Appendix B

When collision occurs, the average number of nodes that participate in a collision is k . When the number of nodes approaches infinity and DCW is used, k approaches a constant value. From the results of (17), when the packet length is fixed,

$$CW = C_1 \times M + C_2, \quad (B1)$$

where CW is the contention window, C_1 and C_2 are constants, and M is the number of nodes. Thus, the probability of transmission τ is

$$\tau = 2/E[CW] = 2/CW = 2/(C_1 \times M + C_2) \quad (B2)$$

When the number of nodes M approaches infinity,

$$\begin{aligned} \lim_{M \rightarrow \infty} (1-\tau)^M &= \lim_{M \rightarrow \infty} \left(1 - \frac{2}{C_1 \times M + C_2}\right)^M \\ &= \lim_{M \rightarrow \infty} \left(1 - \frac{2}{C_1 \times M + C_2}\right)^{\left(\frac{-C_1 \times M - C_2}{2}\right) \left(\frac{2}{-C_1 - \frac{C_2}{M}}\right)} \\ &= e^{\lim_{M \rightarrow \infty} \left(\frac{2}{-C_1 - \frac{C_2}{M}}\right)} \\ &= e^{\frac{-2}{C_1}} \end{aligned} \quad (B3)$$

From (10), $P_i = (1-\tau)^M$, when M approaches infinity, P_i approaches $e^{\frac{-2}{C_1}}$.

$$\begin{aligned} \lim_{M \rightarrow \infty} M \times \tau \times (1-\tau)^{M-1} &= \lim_{M \rightarrow \infty} M \times \frac{2}{C_1 \times M + C_2} \times \left(1 - \frac{2}{C_1 \times M + C_2}\right)^{M-1} \\ &= \lim_{M \rightarrow \infty} \frac{2}{C_1 + \frac{C_2}{M}} \times \lim_{M \rightarrow \infty} \left(1 - \frac{2}{C_1 \times M + C_2}\right)^{\left(\frac{-C_1 \times M - C_2}{2}\right) \left(\frac{2 - \frac{2}{M}}{-C_1 - \frac{C_2}{M}}\right)} \\ &= \frac{2}{C_1} \times e^{\lim_{M \rightarrow \infty} \left(\frac{2 - \frac{2}{M}}{-C_1 - \frac{C_2}{M}}\right)} \\ &= \frac{2}{C_1} e^{\frac{-2}{C_1}} \end{aligned} \quad (B4)$$

From (11), $P_s = M \times \tau \times (1-\tau)^{M-1}$, when M approaches infinity, P_s approaches $\frac{2}{C_1} e^{\frac{-2}{C_1}}$. From (12),

$k = \frac{M \times \tau - M \times \tau \times (1-\tau)^{M-1}}{1 - P_i - P_s}$, when M approaches infinity, we obtain the following using (B2), (B3), and (B4):

$$\begin{aligned} \lim_{M \rightarrow \infty} \frac{M \times \tau - M \times \tau \times (1-\tau)^{M-1}}{1 - P_i - P_s} &= \lim_{M \rightarrow \infty} \frac{M \times \frac{2}{C_1 \times M + C_2} - P_s}{1 - P_i - P_s} \\ &= \lim_{M \rightarrow \infty} \frac{\frac{2}{(C_1 + C_2/M)} - P_s}{1 - P_i - P_s} \\ &= \frac{\frac{2}{C_1} - \frac{2}{C_1} e^{-C_1}}{1 - e^{-C_1} - \frac{2}{C_1} e^{-C_1}} \end{aligned} \quad (B5)$$

From the results of (B5), when the number of nodes approaches infinity and DCW is used, k approaches a constant value.

Appendix C

Equation 16 uses the network parameters provided in Table 1 and packet size in the range of 1-2312 bytes. When solving the relationship between contention window ($2/\tau$) and M , we obtain the results presented in table 4 using regression analysis, where $\lim_{M \rightarrow \infty} k$ is obtained by (B5):

Acknowledgements

This study was supported in part by the National Science Council, Taiwan, under Grant NSC99-2221-E-142-007.

Author details

¹Department of Mathematics Education, National Taichung University of Education, Taichung 40306, Taiwan ²Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan

Competing interests

The authors declare that they have no competing interests.

Received: 15 April 2011 Accepted: 20 September 2011

Published: 20 September 2011

References

1. IEEE Computer Society, IEEE standard 802.11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications (New York, 1999)
2. H Minooei, H Nojumi, Performance evaluation of a new backoff method for IEEE 802.11. *Comput Commun.* **30**(18), 3698–3704 (2007). doi:10.1016/j.comcom.2007.07.003
3. L Yun, L Ke-Ping, Z Wel-Liang, C Qian-Bin, A novel random backoff algorithm to enhance the performance of IEEE 802.11 DCF. *Wirel Personal Commun.* **36**(1), 29–44 (2006). doi:10.1007/s11277-006-6176-8
4. F Cali, M Conti, E Gregori, Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans Netw.* **8**(6), 785–799 (2000). doi:10.1109/90.893874
5. R Bruno, M Conti, E Gregori, A simple protocol for the dynamic tuning of the backoff mechanism in IEEE 802.11 networks. *Comput Netw.* **37**(1), 33–44 (2001). doi:10.1016/S1389-1286(01)00197-9
6. Y Chetoui, N Bouabdallah, Adjustment mechanism for the IEEE 802.11 contention window: an efficient bandwidth sharing scheme. *Comput Commun.* **30**(13), 2686–2695 (2007). doi:10.1016/j.comcom.2007.06.006
7. B-J Kwak, N-O Song, LE Miller, Performance analysis of exponential backoff. *IEEE/ACM Trans Netw.* **13**(2), 343–355 (2005)
8. L Bononi, M Conti, E Gregori, Runtime optimization of IEEE 802.11 wireless LANs performance. *IEEE Trans Parallel Distrib Syst.* **14**(12), 1–15 (2003)
9. R Bruno, M Conti, E Gregori, Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs. *IEEE Trans Mobile Comput.* **1**(1), 10–31 (2002). doi:10.1109/TMC.2002.1011056
10. F Cali, M Conti, E Gregori, IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. *IEEE J Sel Areas Commun.* **18**(9), 1774–1786 (2000). doi:10.1109/49.872963
11. Y Kwon, Y Fang, H Latchman, A novel MAC protocol with fast collision resolution for wireless LANs, in *Proceedings of the IEEE INFOCOM'03*, **2**, 853–862 (2003)
12. T-H Lee, A Marshall, B Zhou, An optimization scheme for energy efficient ad-hoc wireless networks operating in error-prone channel conditions, in *Proceedings of the World Congress on Engineering 2009*, vol. 1. (London, UK, 2009). WCE 2009, July 1-3
13. AHM Rad, J Huang, M Chiang, WWS Wong, Utility-optimal random access without message passing. *IEEE Trans Wirel Commun.* **8**(3), 1073–1079 (2009)
14. G Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J Sel Areas Commun.* **18**(3), 535–547 (2000). doi:10.1109/49.840210
15. G Bianchi, I Tinnirello, Remarks on IEEE 802.11 DCF performance analysis. *IEEE Commun Lett.* **9**(8), 765–767 (2005). doi:10.1109/LCOMM.2005.1496609
16. RG Ayoub, Paolo Ruffini's contributions to the quintic. *Arch History Exact Sci.* **23**(3), 253–277 (1980). doi:10.1007/BF00357046
17. AL Toledo, T Vercauteren, X Wang, Adaptive optimization of IEEE 802.11 DCF based on Bayesian estimation of the number of competing terminals. *IEEE Trans Mobile Comput.* **5**(9), 1283–1296 (2006)
18. G Bianchi, I Tinnirello, Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network. in *Proc Infocom 2003*, **2**, 844–852, (March 2003)
19. CH Foh, JW Tantra, Comments on IEEE 802.11 saturation throughput analysis with freezing of backoff counters. *IEEE Commun Lett.* **9**(2), 130–132 (2005). doi:10.1109/LCOMM.2005.02008
20. Network Simulator <http://www.isi.edu/nsnam/ns/>

doi:10.1186/1687-1499-2011-105

Cite this article as: Huang and Kuo: Dynamic tuning of the IEEE 802.11 distributed coordination function to derive a theoretical throughput limit. *EURASIP Journal on Wireless Communications and Networking* 2011 2011:105.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com