

RESEARCH

Open Access

Performance enhancement of IEEE 802.11 DCF using novel backoff algorithm

Chao-Yu Kuo¹, Yi-Hung Huang^{2*} and Kuan-Cheng Lin³

Abstract

In the IEEE 802.11 distributed coordination function (DCF), the binary exponential backoff algorithm selects a random backoff number from a uniform probability distribution to avoid the problem of packet collision. In this article, we present a novel backoff algorithm that uses a binominal distribution rather than a uniform distribution to determine the backoff value. Simulation results show that the proposed algorithm outperforms the original IEEE 802.11 DCF algorithm.

Keywords: Binary exponential backoff, IEEE 802.11, Carrier sense multiple access/collision avoidance (CSMA/CA)

Introduction

Advances in wireless communication technology have increased the demand for wireless networks. The IEEE 802.11 standard defines the specifications for medium access control (MAC) and the physical layers in a wireless local area network. 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 mechanism. This mechanism employs the acknowledgment (ACK) feature to detect transmission failures. In other words, if an ACK response is not received, it implies that packet transmission has failed. Nodes will wait for an interframe space (IFS), and then, they will invoke the binary exponential backoff (BEB) algorithm [1] that uses an uniform random distribution called contention window (cw) size 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 contention window) and the “cw” value will be doubled when the packet transmission fails. For a node to obtain a backoff value, it has to first determine whether the channel is in use. If the channel is not busy, then the backoff value will decrease by 1 in every time slot and the node will transmit the data when the backoff value reaches zero. However, if the channel is busy, the backoff counter will freeze. When the channel is in an idle state, it will wait for a DCF IFS (DIFS) time period, and then, the backoff value will begin decreasing again. If the packet transmission continues to fail, the cw value will increase to cw_{max} (the maximum contention window); when the node receives an ACK packet, cw will be reset to cw_{min} . If a node receives an error packet, it has to wait for an extended IFS (EIFS) time. Then, the node determines whether the channel is in an idle state again. If it is, then after a DIFS time period, the backoff value will decrease by 1 after each idle slot.

Recent studies have revealed that many commercial network interface cards are adopted with a non-uniform distribution for generating backoff values during contention periods in order to obtain a better performance [2,3]. The p -persistent-based model in [4] shows that the backoff value is sampled by using the p value in a geometric distribution. In [4], it was proved that the analyzed result was slightly better than the simulation result (uniform distribution). In addition, Li et al. [5] have pointed out that very little research has been done on the probability distribution of slot selection. Even though

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

²Department of Mathematics Education, National Taichung University of Education, Taichung, Taiwan

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

some improvements have been proposed by Li et al. [5], however, they did not consider how to improve the performance of throughput and collision probability under the same cw mean value.

All of these perspectives have inspired us to explore whether the throughput efficiency will remain the same under different distributions but with the same cw mean value. Hence, in this article, we present a novel backoff algorithm that uses a binominal distribution rather than a uniform distribution to minimize the degrading effect of random number generators. Simulation results demonstrate that the proposed algorithm outperforms the IEEE 802.11 DCF algorithm.

The remainder of this article is organized as follows. In the following section, the background and the analytical model for the IEEE 802.11 DCF algorithm are briefly reviewed. In Section “The binominal backoff algorithm”, the proposed binominal backoff algorithm is described. In Section “Simulations”, the accuracy of the proposed binominal backoff algorithm is verified via simulation experiments. Finally, the article ends with some conclusions.

Background and related study

In recent researches, two types of contention window value adjustments have been presented to improve network performance. The first type statically adjusts the contention window size when continuous packet collision is detected [1,5,6]. The other type dynamically adjusts the contention window size on the basis of the type of network environment [4,7-9]. It is not easy to accurately estimate the contention window size in the second type of adjustment because network environments may change rapidly. Thus, this type is generally not suitable for real networks. Hence, in this study, we focus on the first type.

Bianchi and Tinnirello [10] developed an analytical model to determine whether the mean contention window size adopted by the BEB algorithm for different distributions is equal; if so, the network throughput will also be equal. Thus, the performance of each modified BEB algorithm could be examined by applying it to the problem of finding the mean contention window size.

Yun et al. [5] proposed a random walking backoff (RWBO) algorithm to enhance the performance of the IEEE 802.11 DCF. In the RWBO algorithm, the BEB algorithm decreases the backoff value by 1 with probability p_d ($0 < p_d < 1$). Similarly, the performance of the RWBO algorithm can be analyzed by applying it to different distributions such as uniform and geometric distributions to determine network throughputs.

Cali et al. [4] proposed an IEEE 802.11⁺ algorithm that uses a geometric distribution to approximate a uniform distribution. Simulation results demonstrated that the

performance of the geometric distribution was slightly better than the uniform distribution under the same mean value for the contention window. This implied that adopting a different distribution function would likely be to influence network performances. Although few studies have addressed the problem of different probability distribution functions for the BEB algorithm, the analytical model in [11] clearly explains the effects of probability distribution functions when the same mean contention window size is used.

Hu et al. [11] introduced the concept of a post-busy slot, i.e., the slot after a busy slot. After a node has transmitted data in a busy slot, it may have the opportunity to transmit data in the post-busy slot (this phenomenon is called short-term non-uniform access in [11]). Further, this phenomenon cannot be analyzed accurately using the model given in [10,12,13], because the collision probability p is neither constant nor independent, as assumed for the analytical model given in [10,12,13].

In accordance with the DCF specifications, after a DIFS time period, the backoff counter at a station will decrease the count by 1 only after another idle backoff slot has elapsed. Hence, only a successful transmitting station may access the post-busy slot after the DIFS. However, assumption 1 in ref. [12] implies that the access probability of each post-busy slot is identical. Hence, the phenomenon of short-term non-uniform access cannot be analyzed using the p -persistent-based model. Thus, Hu et al. proposed a non- p -persistent-based model to analyze this phenomenon. The main underlying concept for the non- p -persistent-based model is described as follows:

The nodes that contend for the post-busy slot can be divided into two groups: one group comprises nodes that have transmitted data in the previous busy slot, and the other group comprises nodes that have not transmitted data in the previous busy slot. In the BEB algorithm, the nodes in the former group obtain a zero backoff value from the distribution function and the nodes in the latter group obtain a non-zero backoff value from the distribution function. In other words, the different probability distributions adopted by the BEB algorithm affect the probability of a node accessing the post-busy slot. Therefore, it is necessary to analyze the non- p -persistent-based model for the effects of the different probability distributions adopted by the BEB algorithm.

The binominal backoff algorithm

In this study, we assume that (1) each node is in a saturated condition (i.e., it always has a packet to transmit) and (2) the channel is free from errors. Packet loss occurs only because of collisions during the packet

transmission process, and the hidden terminal problem is not considered.

Suppose X is the selected backoff value for the BEB algorithm. The probability that $X=x$ is 0.5 on the state space $\{0, W\}$ ($W = \min \{2^k cw_{\min} - 1, cw_{\max} - 1\}$, $k=0, 1, \dots, L$, and L is the retry limit), and hence, the probability density function is given as follows:

$$f_X(x) = \begin{cases} 0.5 & X \in \{0, W\} \\ 0 & X \notin \{0, W\} \end{cases} \quad (1)$$

Thus, the proposed algorithm is given as follows:

Backoff Time = Binominal(cm) aSlotTime.

Here,

Binominal (cw) = cw × uniform distribution either over 0 or 1 (50% each), where cw is an integer within the range of values for the PHY characteristics, aCWmin, and aCWmax, and aCWmin ≤ cw ≤ aCWmax.

aSlotTime = slot time.

As the mean contention window size for a uniform distribution and binominal distribution are equal, i.e., ((cw - 1)/2), it is possible to differentiate between the effects of these two probability distribution functions on the network performance using the non- p -persistent-based model.

However, the access probability of the binomial algorithm for each slot is different, where both the p -persistent-based model and the non- p -persistent-based model cannot be used for an analysis. Therefore, we use the following example to illustrate how the binomial algorithm performs in the backoff process.

Due to the characteristic of binomial algorithm, when the cw value is 31, the node can only choose between 0 and 31 as its backoff value, whereas the probability of it choosing other values as a backoff value is 0. At this time, we may regard each slot as 31 independent channels, where collision will only happen when the different nodes enter an identical channel. On the contrary, when

different nodes enter different channels, collision will not be possible.

As shown in Figure 1, if two nodes want to transmit data simultaneously. In the beginning, both the nodes will set their cw value to 31, if nodes 1 and 2 both choose the backoff value of 0 at the same time, then nodes 1 and 2 will collide (both nodes have to double their backoff values, which means that the cw value will be set to 63). If node 1 chooses 0 as its backoff value and node 2 chooses 63 as its backoff value, then both these nodes will be in different channels.

When the backoff value of node 1 is 0, node 2 will not be able to decrease its backoff value progressively. At this time, the backoff value of node 2 will be frozen; the node will have to wait until node 1 complete its data transmission and then compete again in the next round.

When node 1 reselects its backoff value as 31 (the backoff value of node 1 is then 31, and the backoff value of node 2 is 63), node 1 will transmit its data successfully after processing 31 empty slots; then, the backoff value of node 2 will become 63 - 31 = 32. If node 1 again chooses 31 as its backoff value, node 1 will again transmit its data successfully after processing those 31 empty slots; consequently, the backoff value of node 2 will become 32 - 31 = 1. If node 1 yet again chooses 31 as its backoff value, it will process 1 empty slot and the backoff value of node 2 will become 0. Then, node 2 will start its data transmission, and the backoff value of node 1 will become 31 - 1 = 30.

By observing the repetition of such a process, we obtain the following results: When the backoff value of node 1 is 0 and the backoff value of node 2 is 1, the data transmission of node 1 will succeed. And in the next round, when node 1 chooses 31 as its backoff value and now the backoff value of node 2 is 1. After one slot, then the backoff value of node 2 is 0 and the backoff value of node 1 is 30, the data transmission of node 2 will succeed. When the backoff value of node 2 is 31 and the backoff value of node 1 is 30, then both the nodes will transmit data in different channels without any collision

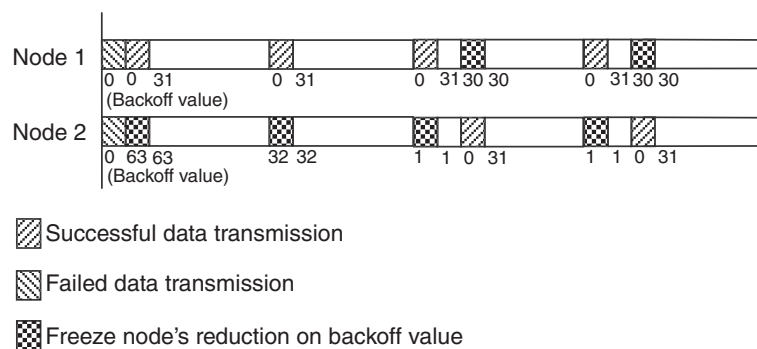


Figure 1 Examples of two nodes.

between the nodes. Therefore, after the initial collision process and different nodes choosing different backoff values in the binomial algorithm, each node would enter into different channels to reduce the probability of collision. We then use Figure 2 to illustrate the backoff phenomenon of the binomial algorithm.

When cw is set to 31, the nodes will have 100% probability of remaining in the same channel (if the backoff value is 0 or 31, then $0 \bmod 31 = 0$ and $31 \bmod 31 = 0$). When cw is set to 63, the nodes will have 50% probability of remaining in the same channel (backoff value is 0) and a 50% chance of switching to the next channel (if the backoff value is 63, then $63 \bmod 31 = 1$). When cw is set to 127, the nodes will have a 50% probability of remaining in the same channel (backoff value is 0) and a 50% chance of switching to the next three channels (if the backoff value is 127, then $127 \bmod 31 = 3$); When cw is set to 255, the nodes will have a 50% probability of remaining in the same channel (the backoff value is 0) and a 50% chance of switching to the next seven channels (if the backoff value is 255, then $255 \bmod 31 = 7$). When cw is set to 511, the nodes will have a 50% probability of remaining in the same channel (the backoff value is 0) and a 50% chance of switching to the next 15 channels (if the backoff value is 511, then $511 \bmod 31 = 15$). When cw is set to 1023, the nodes will have 100% probability of remaining in the same channel (if the backoff value is 0 or 1023, then $0 \bmod 31 = 0$ and $1023 \bmod 31 = 0$). Therefore, the transition diagram of the state can be drawn as given in Figure 2.

The state value of 0, x in Figure 2, illustrates that the node will have a 50% probability of remaining in the same channel and a 50% probability of switching to next x channel(s) (31 channels form a circulation channel, and the channel is calculated by using the modulo method).

Simulations

Environmental settings

In the simulations, we use two performance indices—throughput and conditional collision probability—with a

different node to assess the performance of each algorithm [12]. We define the throughput as follows [10-12]:

$$\text{throughput} = \frac{P_s \times \text{payload}}{P_i \times t_{\text{slot}} + P_s \times t_{\text{success}} + P_c \times t_{\text{col}}}, \quad (2)$$

where P_i is the probability that a slot is idle, P_s is the probability that a slot transmits data successfully, P_c is the probability that a slot is in a collision state, 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. Further, t_{col} is the time spent during packet collision. Notably, $t_{\text{col}} = \text{DATA}_{\text{max}} + \text{DIFS}$ when the algorithm does not utilize the RTS/CTS method. DATA_{max} is the maximum waiting time when packet collision occurs. The normalized throughput performance indicator used in this study is the same as the throughput defined by the formula shown in Equation (2).

The conditional collision probability with different nodes, P_{cc} , is defined as the collision probability that a node attempts to transmit data. A high P_{cc} value does not mean that P_c is high. For instance, if four nodes are transmitting data simultaneously in the same slot, the number of colliding nodes with P_{cc} is 4; however, the number of collisions with P_c is 1. In Figure 3, assume that there are 15 slots, with the number in a slot representing the number of nodes transmitting data in that slot. Thus, there are eight available empty slots, three slots where data are transmitted successfully and four slots with data collisions. Therefore, $P_s = 3/15$, $P_c = 4/15$, $P_i = 8/15$, and $P_{cc} = (2 + 4 + 2 + 3)/(2 + 4 + 1 + 2 + 1 + 1 + 3) = 11/14$.

Moreover, conditional collision probability with different node is used instead of the collision probability when a node is transmitting data to show the effect of packet collision.

In general, it is difficult to detect the state (idle, data transmission, collision) of a slot. This is because, in a wireless environment, the information of each node

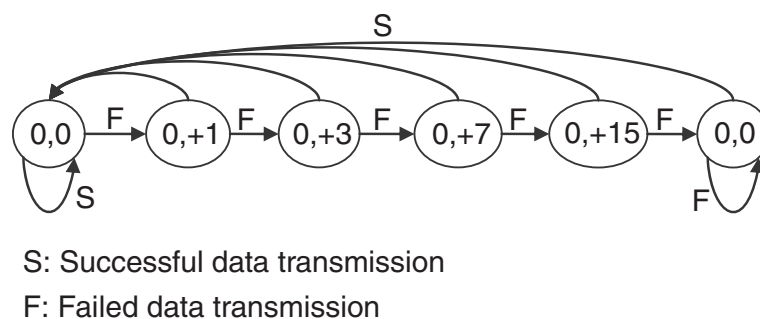


Figure 2 Transition diagram of the state in the proposed algorithm.

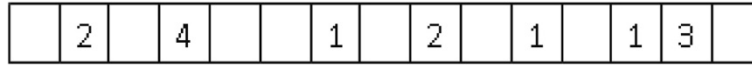


Figure 3 State of virtual slot.

needs to be collected and compiled to determine whether the slot is in the idle, collision, or data transmission state and calculate the P_i , P_s , and P_c values. However, in some cases, inconsistent information will appear for some nodes. For example, a node that receives an error packet has to enter an EIFS time, and then, a DIFS time before entering the backoff stage. If the slot is in the busy state when entering the backoff stage, the node will not be able to distinguish whether this slot has entered the EIFS time. Therefore, the slot status of this node could be different from other nodes and result in information inconsistencies between these nodes. Because this phenomenon makes it difficult to calculate the P_i , P_s , and P_c values, we can only use the P_{cc} value to represent the collision state.

We use NS2 [14] (a network simulator) as our simulation tool and use the network environment parameters presented in Table 1 as simulation parameters. Experiments are repeated 100 times using different seeds, and the results of these 100 runs are averaged to obtain the simulation results.

Experiment I: fixed contention window size

We fix the contention window size cw at 32 to compare the performance of the proposed scheme with uniform distribution, as shown in Figure 4a–d.

Figure 4b represents the standard deviation of Figure 4a, and Figure 4d represents the standard deviation of Figure 4c. Figure 4a shows that the performance of the proposed binominal algorithm is worse than the uniform distribution when the number of nodes is

less than 54. The proposed binomial algorithm in Figure 4a–d performed poorer than the uniform distribution due to the fixed cw size. This result can be expected because the proposed binominal algorithm has only one channel. Therefore, the proposed binominal algorithm will have higher collision probability and standard deviation than the uniform distribution. As for Figure 4a, the proposed binominal algorithm will only achieve higher throughput than the uniform distribution when the number of nodes are higher than 54. This result is caused by the effect of EIFS (equivalent to 18 slots ($\text{EIFS} = 10 + 50 + 112 + 192 = 364 \mu\text{s}$) [12]), which increases the probability of increasing the number of channels. Once a node enters into EIFS and there are no nodes transmitting during this time period, the node that entered into EIFS will have the opportunity to switch to different channels. Therefore, when the number of nodes increases, the probability of nodes switching to different channel also increases. Hence, the proposed binomial algorithm achieves higher throughput than the uniform distribution when the number of nodes are higher than 54.

Experiment II: comparison between different distributions

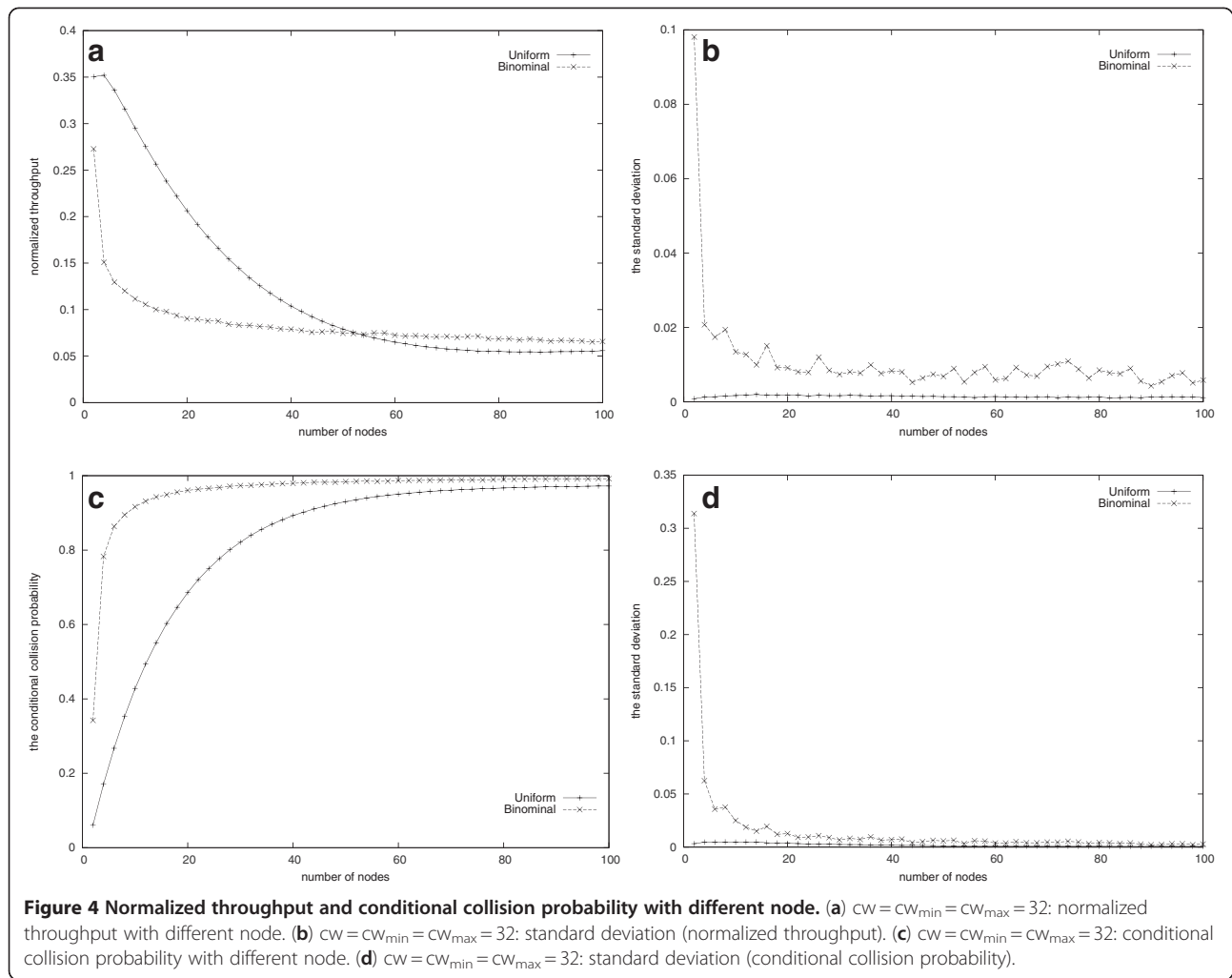
This section compares the normalized throughput and the conditional collision probability for the RWBO + BEB, uniform, geometric, and binominal distributions for the same mean cw . The mean cw for each distribution (Table 2) is computed in accordance with the RWBO + BEB(0.25) [5].

Figure 5 shows that the normalized throughput and conditional collision probability with different nodes of the binominal backoff algorithm are apparently better than other distributions. In particular, the conditional collision probability of the binominal backoff algorithm is 0–24% smaller than other distributions. Further, the throughput of the binominal backoff algorithm is 0–6% better than other distributions. This result shows the impact of different distributions on the network performance.

Figure 5b represents the standard deviation of Figure 5a, and Figure 5d represents the standard deviation of Figure 5c. As we can see from Figure 5b,d, the sampling changes in the proposed algorithm is larger than other methods. This is because the node that uses binomial backoff algorithm must search for channels first. As a result, the standard deviation in the proposed algorithm is larger than other methods. However, Figure 5b shows that the maximum standard deviation for binominal is only

Table 1 Simulation parameters

Data rate	11 Mbps
PCLP data rate	1 Mbps
Basic rate	1 Mbps
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
EIFS	$\text{SIFS} + \text{DIFS} + (\text{ACK length})/\text{basic rate}$
PHY header	192 b
MAC header	224 b
ACK length	112 b + PHY header
CW_{\min}	32
CW_{\max}	1024
Packet size	500 B + PHY header + MAC header



0.0034 and Figure 5d shows that the maximum standard deviation for binomial is only 0.0128. The reason for this is because the binomial collision probability is already small, so the changes in sampling data will be relatively larger.

Results presented in Figure 6 are derived from Figure 5a. However, we are unable to experiment with large amount of nodes in our simulations due to NS2 and hardware (i.e., CPU, memory) constraints. Therefore, the results presented in Figure 6 are produced by self-written

simulation program. Calculations used in this simulation program are based on slot. That is, we assume that a node is in success or collision state only uses one slot time. Hence, we simply investigate the percentage of slots that is in idle, success, or collision states. Each simulation is repeated 100 times with 1,000,000 slots, and average probability is calculated for each states. In addition, from the result observed from Figure 5a, we can see that the simulation results for all distributions are similar except for the binomial distribution. Therefore, simulation conducted in

Table 2 Mean of RWBO + BEB(0.25) set for various distributions

Retransmitting packet number	Mean of RWBO + BEB (0.25)	Uniform	Geometric	Binomial
0	124	Uniform (248)	Geometric (1/125)	Binomial (248)
1	252	Uniform (504)	Geometric (1/253)	Binomial (504)
2	508	Uniform (1016)	Geometric (1/509)	Binomial (1016)
3	1020	Uniform (2040)	Geometric (1/1021)	Binomial (2040)
4	2044	Uniform (4088)	Geometric (1/2045)	Binomial (4088)
5	4092	Uniform (8184)	Geometric (1/4093)	Binomial (8184)

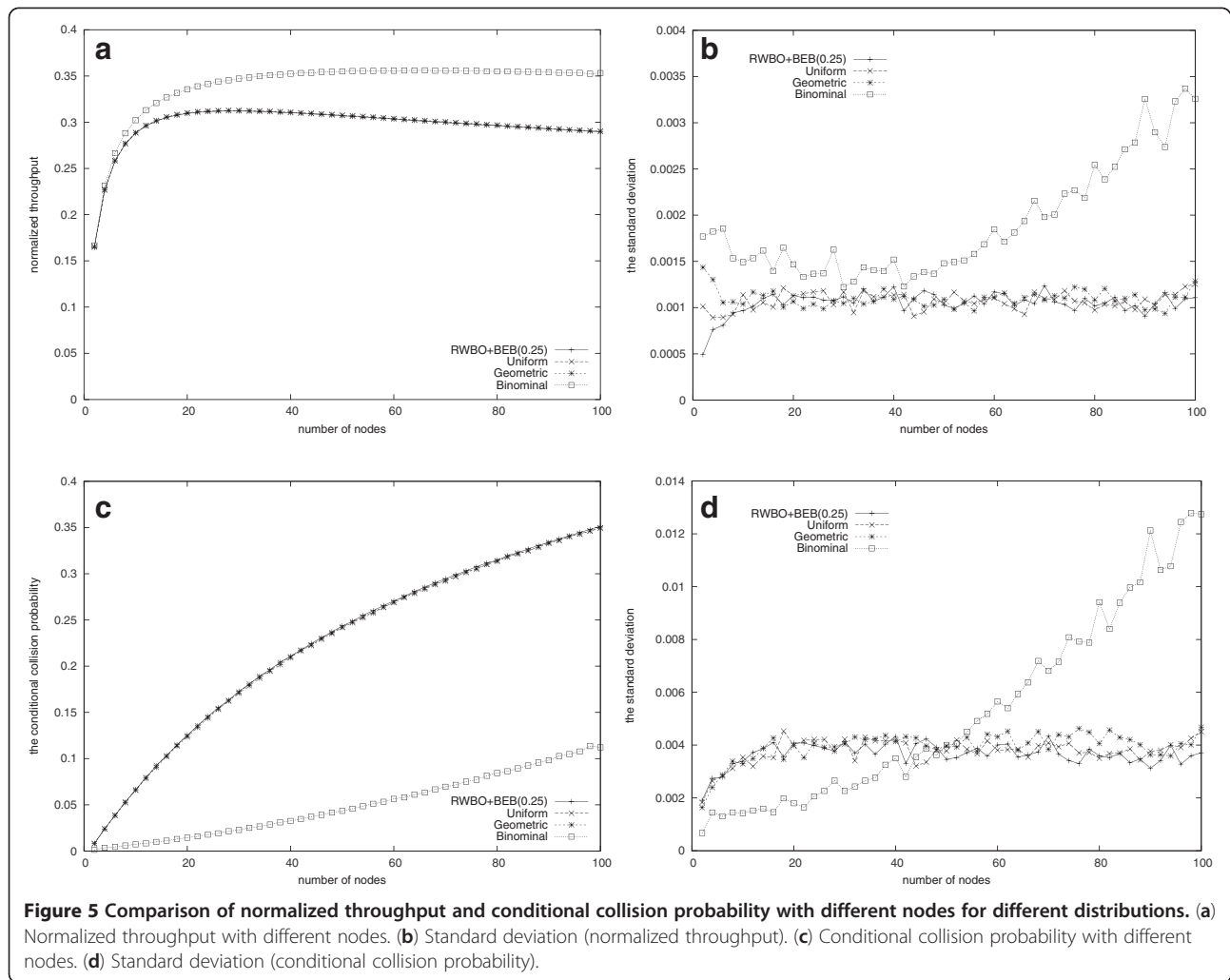


Figure 6 only compares between the binominal distribution and the uniform distribution.

Figure 6a shows that when the number of nodes reaches 10,000 in the uniform distribution, the probability of idle,

success, or collision state seems likely to tend to a fixed value of their respective. As for Figure 6b, when the number of nodes reaches 140 in the binominal distribution, same phenomenon appears to happen earlier than the

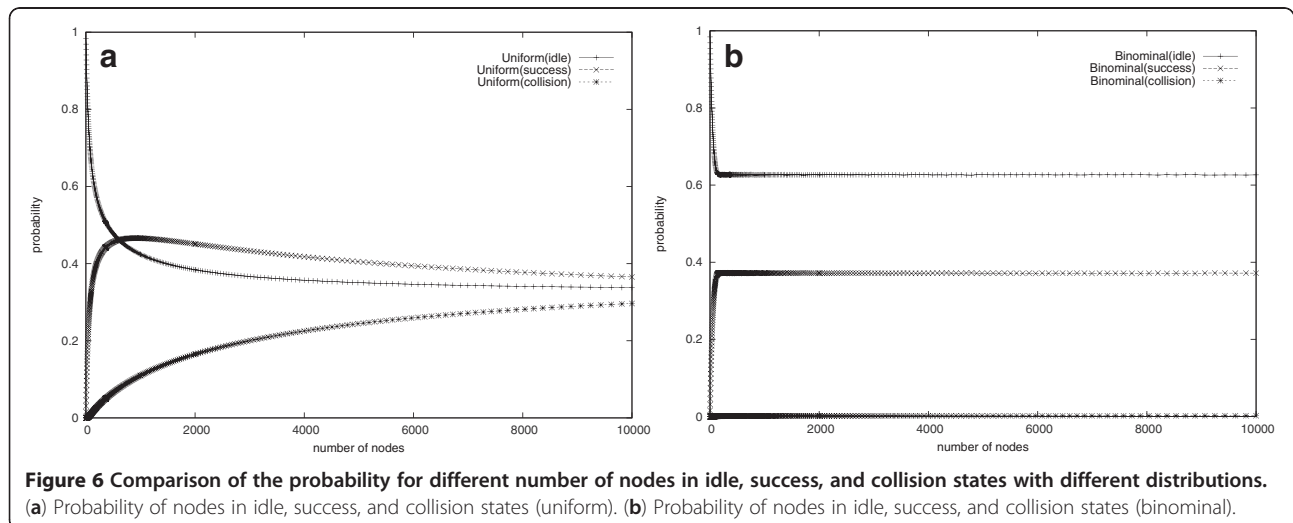


Table 3 Each node uses CBR Traffic to set its time for sending data, $x = 2, 4, 6, \dots, 16$

Node number	Start time (s)	End time (s)
$1 - x$	0	70
$x + 1 - 2x$	10	60
$2x + 1 - 3x$	20	50
$3x + 1 - 4x$	30	40

uniform distribution. Therefore, from Figure 6 and Equation (2) we can infer that when the number of nodes reaches 140 in Figure 5a, the result of the binominal distribution may also arrive at a flat horizontal line gradually. Furthermore, by comparing both Figure 6a,b, when the number of nodes reaches 144 in the uniform distribution, the probability of success state in the uniform distribution is only slightly higher than the probability of success state in the binominal distribution about 0.023. When the number of nodes reaches 6 in the uniform distribution, the probability of collision state in the uniform distribution is only slightly higher than the probability of collision state

in the binominal distribution about 0.3. Therefore, from Equation (2), we can infer that when the number of nodes increases, the normalized throughput in the binominal distribution will achieve better result than the uniform distribution.

Experiment III: changing number of nodes in simulation process with different distributions

In this experiment, the performance of the proposed algorithm is evaluated by changing the number of nodes in the simulations. We assume that the nodes transfer data to each other using CBR traffic (11 Mbps). Table 3 shows the transfer time for each node. The simulation time is 70 s.

When we use the parameters for each distribution given in Table 2 and the network parameters given in Table 3, the performance of the proposed algorithm is better than the other methods in a changing network environment (Figure 7). In particular, the conditional collision probability of the binominal backoff algorithm is 3–19% smaller than the other distributions. In addition, the throughput of the binominal backoff algorithm is slightly

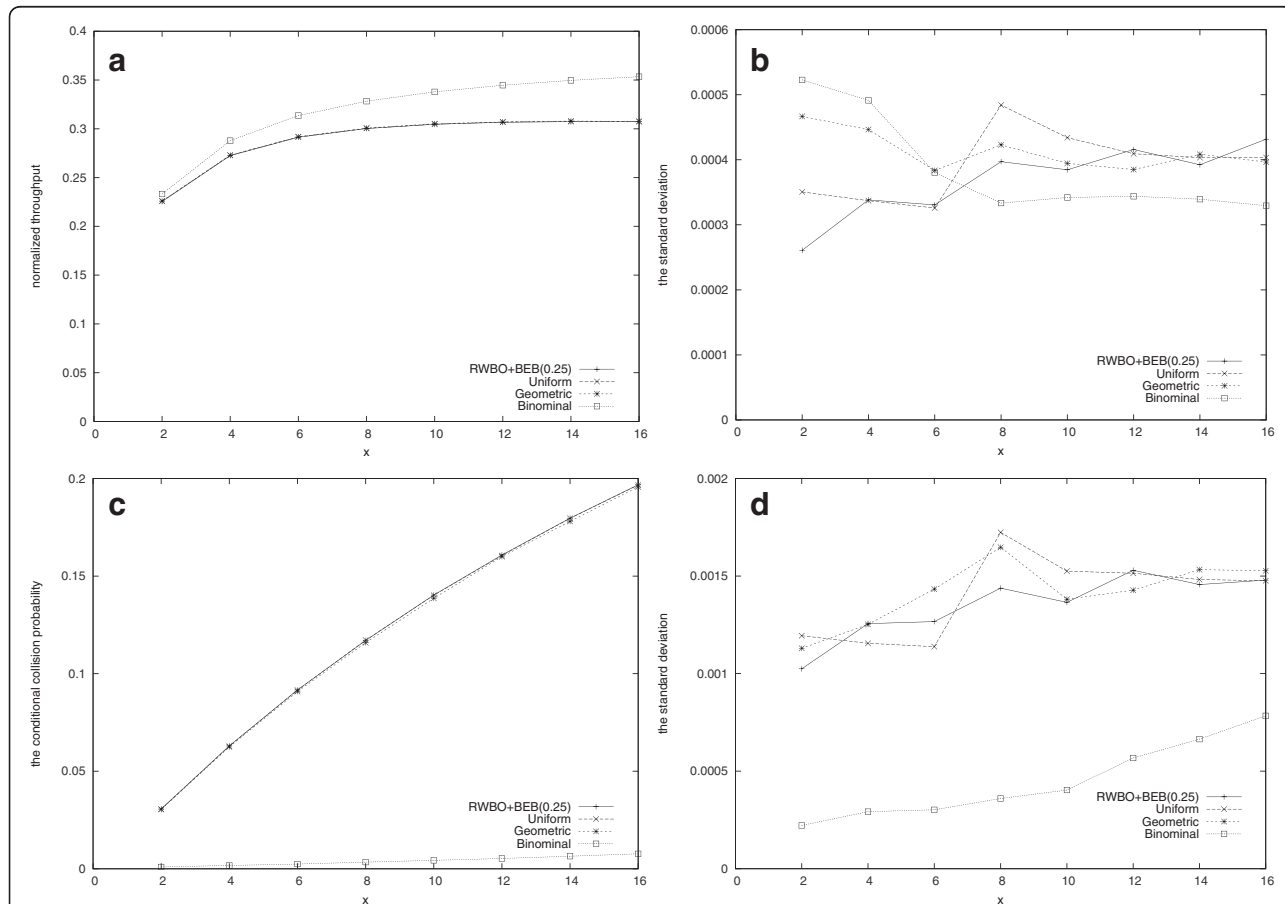


Figure 7 Comparison of normalized throughputs and conditional collision probability with different nodes for different distributions in a dynamic environment. (a) Normalized throughput with different nodes. (b) Standard deviation (normalized throughput). (c) Conditional collision probability with different node. (d) Standard deviation (conditional collision probability).

better (0–5%) than the other methods. As the mean cw is large, the performance of the binominal backoff algorithm is improved slightly. This result shows the impact of different distributions on the network performance.

Figure 7b represents the standard deviation of Figure 7a, and Figure 7d represents the standard deviation of Figure 7c. As we can see from Figure 7b,d, the sampling changes in the proposed algorithm are similar to other methods. Figure 7b shows that the maximum standard deviation for binominal is only 0.00053 and Figure 7d shows that the maximum standard deviation for binominal is only 0.00078. This is because the number of channel is 248, which is way larger than the number of node used in the experiment. As a result, the standard deviation will be relatively low.

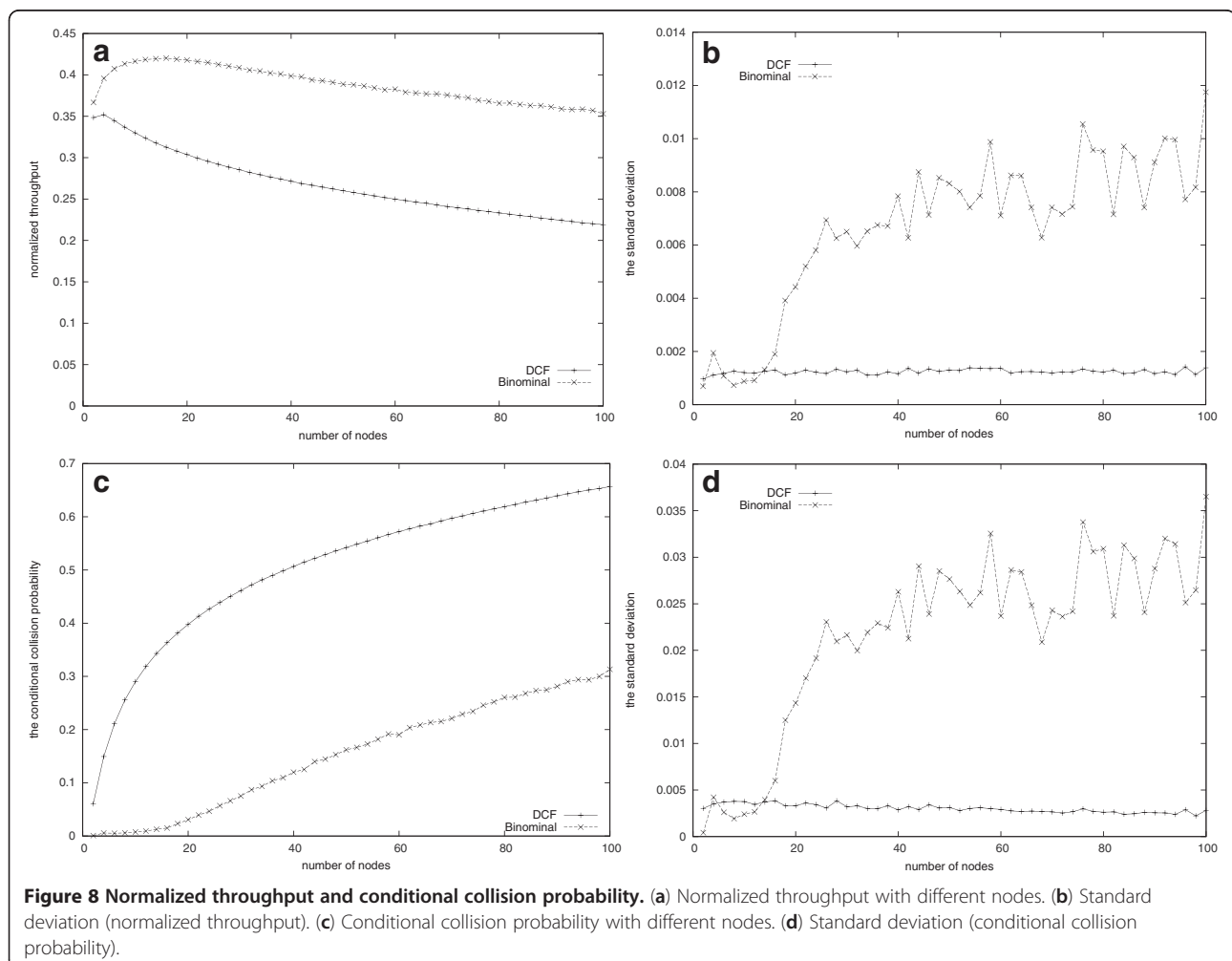
Experiment IV: comparison with DCF

We set the contention window size cw within the range $cw = [cw_{\min} = 32, cw_{\max} = 1024]$ to compare the performance of the proposed scheme with DCF, as shown in Figure 8a,c.

Figure 8c shows that the proposed binominal algorithm can decrease the conditional collision probability (6–39%), and Figure 8a shows that it has a better throughput (2–14%) than the IEEE 802.11 DCF. Thus, the binominal algorithm can decrease the conditional collision probability and gives a better throughput than the IEEE 802.11 DCF.

Figure 8b represents the standard deviation of Figure 8a, and Figure 8d represents the standard deviation of Figure 8c. As we can see from Figure 8b,d, the sampling changes in the proposed algorithm are larger than DCF method. This is because the node that uses binominal backoff algorithm must search for channels first and the number of channel is 32 where the number of node is between 2 and 100. As a result, the standard deviation in the proposed algorithm is larger than DCF method. However, Figure 8b shows that the maximum standard deviation for binominal is only 0.0118 and Figure 8d shows that the maximum standard deviation for binominal is only 0.0365.

In Figure 9, the x -axis represents the number of nodes and the y -axis represents the $E[cw]$ value measured in the



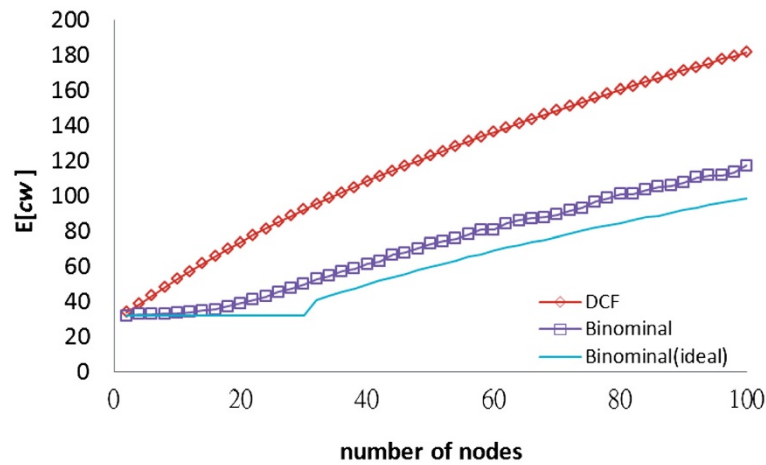


Figure 9 $E[cw]$ for DCF and binominal method.

simulation experiment related to the results shown in Figure 8. The binominal (ideal) curve represents the ideal binominal results. In Figure 9, the binominal (ideal) curve only shows the obvious value changes in $E[cw]$ at the

point where the number of nodes is 32. This is because $cw_{min} = 32$, when there are only 31 channels, and only when the number of nodes is above 32, there will be 2 or more nodes contending for one channel; then, only this

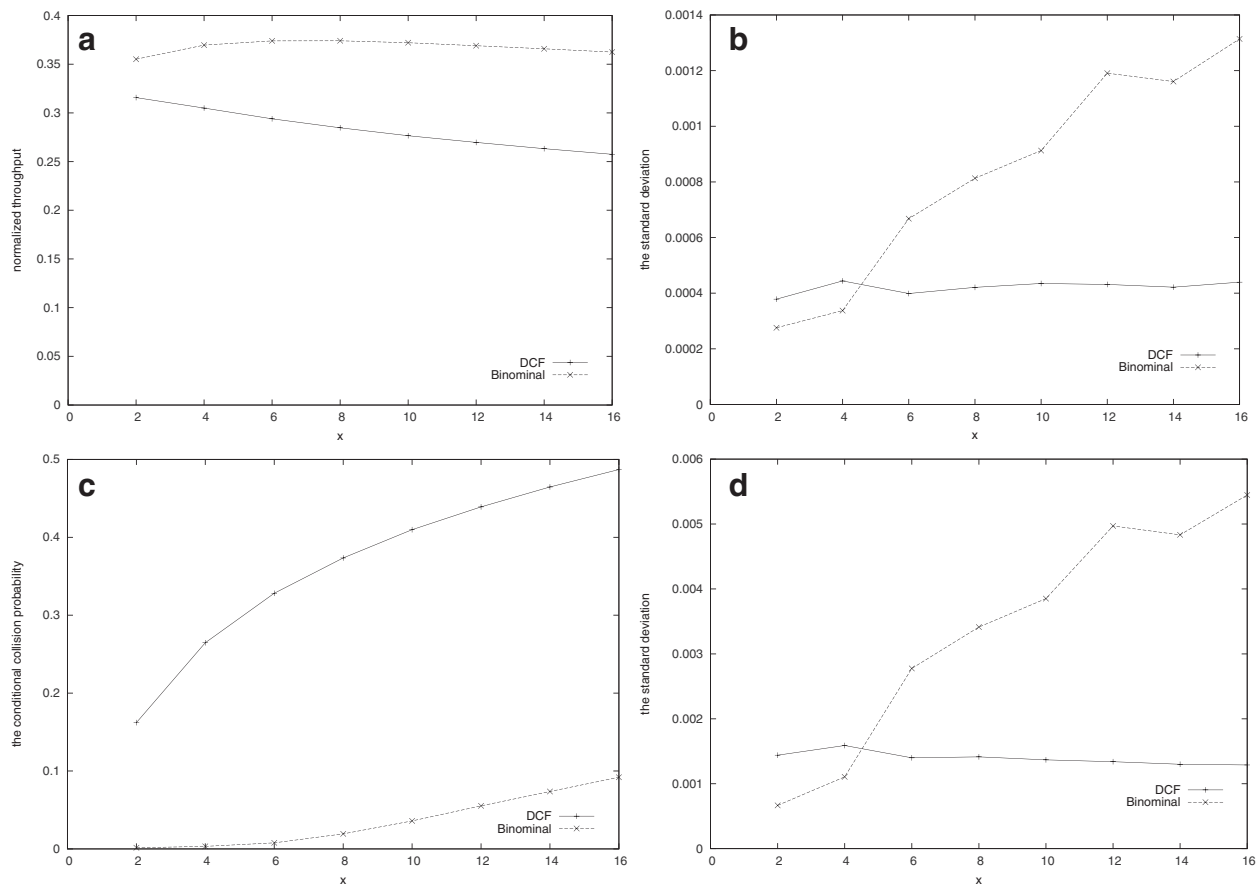


Figure 10 Comparison of normalized throughputs and conditional collision probability with different nodes in a dynamic environment. (a) Normalized throughput with different nodes. (b) Standard deviation (normalized throughput). (c) Conditional collision probability with different nodes. (d) Standard deviation (conditional collision probability).

situation will cause the value of $E[cw]$ to increase. From the comparison result between the binominal curve and the binominal (ideal) curve shown in Figure 9, due to a shorter experiment time for the binominal curve, the collision that happens at the beginning when the nodes are searching for channels has a larger influence on the incrementation of binominal curve $E[cw]$ value. Moreover, as expected, the binominal curve will become closer and closer to the binominal (ideal) curve as the simulation time increases. This phenomenon will also slightly improve the performance of the binominal method related to the results shown in Figure 8.

Experiment V: comparison with DCF for a changing number of nodes in the simulation process

In this experiment, the performance of the proposed algorithm is evaluated by changing the number of nodes in the simulations. We assume that the nodes transfer data to each other using CBR traffic (11 Mbps). Table 3 lists the transfer time for each node. The simulation time is 70 s.

When we use the network parameters given in Table 3, the performance of the proposed algorithm is found to be better than the IEEE 802.11 DCF for a changing network environment (Figure 10). In particular, the conditional collision probability of the binominal backoff algorithm is 16–40% smaller than the IEEE 802.11 DCF. In addition, the throughput of the binominal backoff algorithm is slightly better (4–11%) than the IEEE 802.11 DCF. Since the mean cw is large, the performance of the binominal backoff algorithm is slightly improved. Thus, the binominal algorithm can decrease the conditional collision probability and it gives a better throughput than the IEEE 802.11 DCF.

Figure 10b represents the standard deviation of Figure 10a, and Figure 10d represents the standard deviation of Figure 10c. As we can see from Figure 10b,d, the sampling changes in the proposed algorithm is larger than DCF method. This is because the node that uses binominal backoff algorithm must search for channels first and the number of channel is 32 where the number of node is between 8 and 64. As a result, the standard deviation in the proposed algorithm is larger than DCF method. However, Figure 10b shows that the maximum standard deviation for binominal is only 0.0013 and Figure 10d shows that the maximum standard deviation for binominal is only 0.0055. The reason for this is because the binominal collision probability is already small, so the changes in sampling data will be relatively larger.

Conclusions

In this article, we proposed a novel backoff algorithm that uses a binominal distribution, instead of a uniform distribution, to select the backoff value. After the initial collision process and after different nodes have chosen

different backoff values in the binomial algorithm, each node enters different channels to reduce the probability of collision. Simulation results showed that the proposed algorithm performs better than the IEEE 802.11 DCF algorithm for conditional collision probability and normalized throughput. We plan to study the behavior of the binomial backoff algorithm further as our future study and also plan to propose a mathematical model to analyze the performance of this type of algorithm.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan. ²Department of Mathematics Education, National Taichung University of Education, Taichung, Taiwan. ³Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan.

Received: 18 June 2011 Accepted: 8 August 2012

Published: 23 August 2012

References

1. IEEE Computer Society, *IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (IEEE Press, New York, 1999)
2. G. Bianchi, A. Di Stefano, C. Giaconia, A. Scaglione, L. Scalia, G. Terrazzino, I. Tinnirello, Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards, in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Anchorage, Alaska, USA, May 2007, pp. 1181–1189
3. F. Babich, M. Comisso, A. Dorn, A practical method for verifying the uniformity of the backoff distribution in 802.11 network cards, in *Proceedings of IEEE International Conference on Communications (ICC)*, Cape Town, South Africa, May 2010, pp. 1–5
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)
5. L. Yun, L. Ke-Ping, Z. Wei-Liang Zhao, 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)
6. H. Minooei, H. Nojumi, Performance evaluation of a new backoff method for IEEE 802.11. *Comput. Commun.* **30**(18), 3698–3704 (2007)
7. 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)
8. 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)
9. B.J. Kwak, N.-O. Song, L.E. Miller, Performance analysis of exponential backoff. *IEEE/ACM Trans. Netw.* **13**(2), 343–355 (2005)
10. G. Bianchi, I. Tinnirello, Remarks on IEEE 802.11 DCF performance analysis. *IEEE Commun. Lett.* **9**(8), 765–767 (2005)
11. C. Hu, H. Kim, J.C. Hou, Short-term non-uniform access in IEEE 802.11-compliant WLANs: a study on its impact on the saturation performance. *Comput. Netw.* **52**, 61–76 (2008)
12. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Sel. Areas Commun.* **18**(3), 535–547 (2000)
13. J. Jeong, S. Choi, C. Kim, Achieving weighted fairness between uplink and downlink in IEEE 802.11 DCF-based WLANs, in *Proceedings of 2nd ACM International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, Lake Vista, Florida, USA, Aug. 2005, pp. 10–22
14. *Network Simulator 2005* <http://www.isi.edu/nsnam/ns/>. Accessed 10 Oct 2010

doi:10.1186/1687-1499-2012-274

Cite this article as: Kuo et al.: Performance enhancement of IEEE 802.11 DCF using novel backoff algorithm. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:274.