A Multi-Chain Backoff Mechanism for IEEE 802.11 WLANs

Shiang-Rung Ye and Yu-Chee Tseng
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsin-Chu, 30050, Taiwan.
Email:(shiarung,yctseng@csie.nctu.edu.tw)

Abstract

The distributed coordination function (DCF) of IEEE 802.11 standard adopts the binary exponential backoff (BEB) for collision avoidance. In DCF, the contention window of a station is reset to an initial value, \( CW_{\text{min}} \), after each successful transmission. Much research has shown that this dramatic change of window size may degrade the network performance. Backoff algorithms, such as GDCF, MILD, EIED, etc., have been proposed that try to keep the congestion level by not resetting the contention window after each successful transmission. In this paper we propose a multi-chain backoff (MCB) algorithm that enables stations to adapt to different congestion levels by using multiple backoff chains with collision events on the wireless channel as indications for stations to switch among the chains. The performance of MCB is compared to the performances of existing backoff algorithms, 802.11 DCF, GDCF, MILD, EIED, and LILD. Simulation results show that MCB can achieve higher network throughput while maintaining fair channel access than the existing algorithms.

1. Introduction

The wireless local area network (WLAN) is emerging as a promising technology that provides high-speed and low-cost wireless communications. In WLANs the medium access control (MAC) plays an important role on efficient and fair use of the wireless medium. The simplest MAC protocol, ALOHA [1], allows stations to transmit immediately upon receiving data from upper layers. With Poisson traffic arrivals, the achievable throughputs for pure ALOHA and slotted ALOHA are only 0.184 and 0.368, respectively [2]. The inefficacy of ALOHA results from its high collision probability under heavy traffic load. To decrease the collision probability, the carrier sense multiple access (CSMA) scheme [2] requires stations to sense carriers on the wireless channel before transmitting data. If the medium is busy, they should defer their transmission until the medium becomes idle. This prevents stations's frames from colliding with ongoing transmission of other stations. When a station detects that the medium is busy, it can persistently wait for the medium to become idle, and then transmit with a probability of one or \( p \). The former is called 1-persistent CSMA and the latter is \( p \)-persistent CSMA. Alternatively, a station can stop monitoring the wireless medium, and then listen to the medium again after a random time period. This is called nonpersistent CSMA.

The distributed coordination function (DCF) of IEEE 802.11 is a variant of persistent CSMA with a collision avoidance scheme. Two types of carrier sense mechanisms are defined in DCF: physical carrier sense and virtual carrier sense. The former is supported by the PHY layer via actual channel assessment, and the latter is supported by the MAC layer. The virtual carrier sense is carried out by the network allocation vector (NAV) which is declared by the Duration/ID field in overheard control frames or data frames. In DCF, only when both carrier sense mechanisms indicate that the medium is idle, can a station proceed with the rest of contention procedure.

The collision avoidance (CA) scheme of DCF further reduces the collision probability by requiring each backlogged station to perform binary exponential backoff (BEB) after the medium becomes idle. In BEB, if a station successfully transmits a frame, its contention window is reset to an initial value, \( CW_{\text{min}} \). However, if the transmission fails, the window size is doubled. The maximum window size is restricted to \( CW_{\text{max}} \). The sizes of \( CW_{\text{min}} \) and \( CW_{\text{max}} \) are defined in the PHY layer of IEEE 802.11. For frequency
hopping (FH), CWmin=15 and CWmax=1023, and for direct sequence spread spectrum (DSSS), CWmin=31 and CWmax=1023.

In the literature, it has been shown that the size of contention window has a great impact on the performance of DCF [3–10]. In this paper, we propose a multi-chain backoff (MCB) algorithm that enables stations to adapt to different congestion levels by exploiting multiple backoff chains, each of which is a sequence of backoff stages, and the collision events that occur on the wireless channel to drive transitions among these chains. The advantage of the MCB is that it doesn’t have to estimate the number of contending stations, traffic load, etc., but provides high throughput and fair channel access for WLANs with a small or large population.

The paper is organized as follows. In Section 2, we review related work on backoff algorithms. In Section 3, we present the multi-chain backoff algorithm. Section 4. shows simulation results, and Section 5. concludes this paper.

2. Related Work

In the literature, there has been much work addressing the performance issues of IEEE 802.11 DCF [3–7, 11–16]. In [11], to prevent the oscillation of the contention window in binary exponential backoff (BEB), the multiplicative increase, linear decrease (MILD) algorithm increases the contention window by 1.5 times when collision occurs, and decreases the contention window by one when transmission succeeds. To ensure fair access to the wireless medium, a transmitting station is required to attach its contention window in the next transmitted frame. Whenever any station overhears this frame, it shall adopt the window size. However, since a station with a smaller contention window has a better chance to win, this may force other colliding stations to adopt this (smaller) window size. In [3, 4], it is suggested to choose a contention window according to the estimated number of competing stations. While this may significantly improve performance, it relies on the accurate estimation of the number of competing stations.

The exponential increase, exponential decrease (EIED) algorithm [15, 17, 18] increases the contention window by a multiple when collision occurs and exponentially decreases the window size when transmission succeeds. With a relatively small decrement of the window size, compared to the increment, the EIED can outperform DCF [15]. However, our simulation results show that with such a relatively small decrement, the EIED may suffer from unfair access when the number of contending stations is small. The linear increase, linear decrease (LILD) algorithm always adjusts the contention window by a constant [8, 18], which is not suitable for networks with a large population. The GDCF backoff algorithm [9] doubles the contention window after each unsuccessful transmission and halves the window size after c consecutive successful transmissions. Although GDCF greatly improves the performance of 802.11 DCF, it may cause unfair medium access for some values of c when the number of contending stations is small. The works [5, 12] show that the performance of DCF is highly related to the number of contending stations and the minimum contention window (CWmin). The CWmin should increase as the number of contending stations increases. In the next section, we will propose a multi-chain backoff algorithm that employs multiple backoff chains, each of which has a different minimum contention window; thereby, stations can adapt to different congestion levels by switching among the chains.

3. The Multi-Chain Backoff Algorithm (MCB)

In MCB, each station maintains a transition diagram, as demonstrated in Fig. 1, to determine its current contention window. The diagram consists of c backoff chains, numbered from 0 to c − 1, each of which represents a sequence of backoff stages and is defined by the following parameters:

- \( w_i \) : the minimum contention window for chain \( i \).
- \( m_i \) : the maximum backoff stage of chain \( i \).
- \( u_i \) : the transition probability from chain \( i \) to chain \( i + 1 \). In case of \( i = c - 1 \), \( u_{c-1} = 0 \).
- \( v_i \) : the transition probability from chain \( i \) to chain \( i - 1 \). In case of \( i = 0 \), \( v_0 = 0 \).

For each backoff chain \( i \), we define \( w_0 = CW_{min} \) and \( w_{c-1} = CW_{max} \). For \( i = 1 \cdots c - 2 \), \( w_i \) could be

\[
 w_i = CW_{min} + i \cdot \left\lfloor \frac{CW_{max} - CW_{min}}{c - 1} \right\rfloor .
\]

Alternatively, we may increase \( w_i \) in an exponential manner as follows,

\[
 w_i = (CW_{min} + 1) \cdot \left[ \frac{CW_{max} + 1}{CW_{min} + 1} \right]^{i+1} - 1 .
\]
Within a backoff chain, the contention window is doubled for the next backoff stage but is limited to $CW_{\text{max}}$.

Parameters $u_i$ and $v_i$ are probabilities for a station to switch from its current chain to the next chain and the previous chain, respectively. The simplest assignment is to let all $u_i$ be the same, and all $v_i$ be the same. In this case, the optimal values for $u_i$ and $v_i$ are related to the number of competing stations, the length of a backoff slot, and the average size of data frames.

With the above defined parameters, the MCB algorithm works as follows. Initially, each station is in stage 0 of chain 0. Before transmitting data, a station randomly chooses a backoff value from the current contention window. If the medium is idle for a DIFS period, the station starts to perform backoff. For each idle slot being detected, its backoff counter is decreased by one. If the medium is busy during a backoff slot, the backoff counter is frozen and the station shall wait until the medium becomes idle. Once its backoff counter reaches zero, the station can start to transmit a frame. During the backoff period, the station shall also detect any collision event caused by other stations. A collision flag $f_{\text{col}}$ will be set if a station itself experiences a collision or it detects that the medium has been busy for a duration longer than the transmission time of the smallest frame, but this doesn’t result in a receipt of a correct frame. Assume that a station is transmitting in stage $j$ of chain $i$. In case that the transmission fails, it will move to stage $j + 1$ of chain $i$ if $j < m_i$, or stay in the same stage if $j = m_i$. In case that the transmission succeeds, if $f_{\text{col}}$ is set, it will move to stage 0 of chain $i + 1$ with probability $u_i$, but move to stage 0 of chain $i$ with probability $1 - u_i$. In case that the transmission succeeds and the $f_{\text{col}}$ is not set, it will move to stage 0 of chain $i - 1$ with probability $v_i$, but stay in stage 0 of chain $i$ with probability $1 - v_i$. After each successful transmission, the $f_{\text{col}}$ is cleared. Intuitively, when a station encounters or detects a collision event, it moves to a chain with a larger minimum contention window with probability $u_i$, and if no collision is encountered and detected, it moves to a chain with a smaller minimum contention window with probability $v_i$.

4. Performance Evaluation

This section presents our simulation results of the performance of MCB as opposed to MILD, DCF, GDCF, EIED, and LILD algorithms. The custom simulation programs are written in C++ that simulate networks with an ideal wireless channel (i.e., no hidden terminals). In addition to evaluating the saturation throughput, a fairness index ($FI$) [10] is used to examine the fairness property of a backoff algorithm,

$$FI = \frac{(\sum i S_i)^2}{n \cdot \sum i (S_i)^2},$$

where $S_i$ is the saturation throughput received by station $i$. The value of $FI$ is bounded in the interval $[1, 0]$. An algorithm is fair as its $FI$ is close to 1. Table 1 lists the MAC-layer and PHY-layer parameters, used in our simulations. Unless otherwise stated, we use the same $u$ and the same $v$ for all chains through out the simulations.

4.1. Saturation Throughput

Fig. 2 presents the saturation throughput under different $u$ and $v$, assuming a frame size of 1024 bytes. First, we vary $u$ from 0 to 1 with fixed $v = 0.5$. When $u = 0$, MCB behaves the same as IEEE 802.11. The figure shows that saturation throughput decreases as the number of competing stations $n$ increases. Given a

![Transition Diagram of MCB]

Figure 1: The transition diagram of MCB. The $j$-th backoff stage of chain $i$ is denoted by $(i, j)$ in the figure. Symbols $s$ and $f$ denote success and failure transmission events, respectively.
fixed $n$, the throughput increases as $u$ increases. Next, we fix $u = 1$ and change the value of $v$ from 0 to 1. When $v$ is small, the saturation throughput drops first and then increases as $n$ increases. The drop of throughput when $n$ is small in some cases is because MCB is quite sensitive in tuning stations’ backoff windows, thus causing longer backoff time. However, as $n$ increases, the benefit of MCB can be seen.

Fig. 3 shows the saturation throughput, assuming a frame size of 128 bytes. In Fig. 3, we first fix $v = 0.5$ but vary $u$. The throughput increases when $u$ decreases from 0 to 0.1. Further decrease of $u$ will degrade the throughput. Then, we vary $v$ from 1 to 0 with $u = 0.1$. The result indicates that a $v_i$ around 0.3 could be a good choice. Fig. 4 shows the relations between throughput and frame sizes. When the frame size increases, since less backoff overhead is incurred, the throughput also increases.

4.2. The Number of Backoff Chains

In Fig. 5, we show the throughput of MCB under different numbers of chains with $u$ and $v$ which are chosen for $n = 6$ and $n = 46$. In the case that the $u$ and $v$ are chosen for $n = 6$, the throughput of two-chain MCB drops more when the number of stations $n$ increases. In the case that $u$ and $v$ are chosen for $n = 46$, the throughput of two-chain MCB decreases more when $n$ is small. Fig. 6 shows the corresponding fairness indexes. The fairness index tends to oscillate when more backoff chains are used.

4.3. Comparison to Existing Algorithms

In the following, we compare the performance of our four-chain MCB to existing algorithms. For the four-chain MCB, the minimum contention windows are 31, 127, 511 and 1023, respectively. The $u$ and $v$ are chosen according to the simulation results in Fig. 3 and Fig. 2. The $u$ and $v$ are 0.1 and 0.3, respectively, for the frame size of 128 bytes, and are 1 and $v = 0.3$, respectively, when the frame size is 1024 bytes.
4.3.1. Comparing with GDCF

In Fig. 7, we compare the throughput of MCB to that of GDCF, assuming the frame size is 128 bytes. For GDCF, we increase the parameter $c$ from 1 to 5. With a smaller $c$, although GDCF achieves higher throughput for small $n$, the throughput drops dramatically as $n$ increases. However, with a larger $c$, it is clear that MCB outperforms GDCF.

Fig. 8 compares MCB and GDCF with a frame size of 1024 bytes. With a larger $c$, GDCF performs as well as MCB. However, as shown in Fig. 9, GDCF will cause unfair channel access when $n$ is small. Considering the case of $n = 2$, when the sizes of contention windows of the two stations are different, the occurrence of collision will double the difference of their window sizes. Since a station with a small contention window has shorter backoff time, it may reach $c$ successful transmissions to halve its window size faster than the other station.

Figure 6: Fairness Index.

Figure 7: Saturation throughput of MCB and GDCF with frame size 128 bytes.

Figure 8: Saturation throughput of MCB and GDCF with frame size 1024 bytes.

Figure 9: Fairness index of MCB and GDCF with frame size 1024 bytes.

Figure 10: Saturation throughput of MCB, IEEE 802.11, and MILD.
4.3.2. Comparing with IEEE 802.11 and MILD

Fig. 10 shows the throughputs of MCB, IEEE 802.11, and MILD. The throughput of MILD is lower than those of MCB and IEEE 802.11. In MILD, a colliding station can advertise its new contention window only if no successful transmission occurs during its backoff period. However, there is a high probability that a station with a smaller contention window successfully transmits during this period. This will force the colliding stations to adopt the small contention window before advertising their windows. The IEEE 802.11 is outperformed by MCB since MCB offers more than one chain, allowing stations to adapt to different congestion conditions.

4.3.3. Comparing with EIED

Fig. 11 compares the throughput of MCB to that of EIED. For EIED, we use EIED(x, y) to denote that if a collision occurs, $CW_{new} = \min(x \cdot (CW_{old} + 1) - 1, CW_{max})$, and if a transmission succeeds, $CW_{new} = \max((CW_{old} + 1)/y - 1, CW_{min})$. In Fig. 11, we fix $y = 2$ and increase $x$ from 2 to 32 and in Fig. 12, we fix $x = 2$ and vary $y$ from 1.01 to 2. Only the performance of EIED(2, 1.01) is comparable to MCB; however Fig. 13 shows that the wireless medium is unfairly utilized when $n < 8$. Taking EIED(2, 1.01) at $n = 2$ as an example, since the sizes of these two contention windows may not be same, when a collision occurs, the difference between the two window sizes is doubled. Since the decrement of contention windows is slow (1 for $CW < 100$), the window sizes can hardly be reduced to CWmin after a number of successful transmissions. Once a collision occurs again, the difference is doubled.
4.3.4. Comparing with LILD

Fig. 14 and Fig. 15 compare MCB to LILD. We use LILD\((x, y)\) to denote the LILD that sets \(\text{CW}_{\text{new}} = \min(\text{CW}_{\text{old}} + x, \text{CW}_{\text{max}})\) if a collision occurs, and set \(\text{CW}_{\text{new}} = \max(\text{CW}_{\text{old}} - y, \text{CW}_{\text{min}})\) if a transmission succeeds. For a smaller \(x\), the throughput will decrease as \(n\) increases, and for a larger \(x\), it increases when \(n\) increases. In Fig. 15, a large decrement will cause serious collisions and degrade throughput when \(n\) is large. If \(y\) is quite small compared to \(x\), the throughput increases as \(n\) increases, except for \(n = 2\).

5. Conclusions

In this paper, we have proposed a new MCB algorithm. MCB explores the possibility of using multiple backoff chains with different minimum contention windows and considering collision events on the wireless channel as hints to choose a proper chain. With the capability of switching to different backoff chains, MCB offers higher throughput than the existing protocols, such as GDCF, IEEE 802.11, MILD, EIED, and LILD, yet still provides fair access to the wireless channel. The work in [18] uses backoff to avoid consecutive burst errors and achieve better efficiency. How to apply our multi-chain concept to resolve these issues could be directed to future work.

References


