A backward-compatible multiple-round collision avoidance scheme for contention based medium access control

Jeongkyun Yun *, Jin-Ghoo Choi, Saewoong Bahk

INMC, School of Electrical Engineering, Seoul National University, Shillim-9-dong, 302-511-3, Kwanak-gu, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 28 April 2008
Received in revised form 8 March 2009
Accepted 9 March 2009
Available online 19 March 2009

Keywords:
Contention
MAC
CSMA/CA
802.11

ABSTRACT

Random-access mechanisms play an important role in wireless networks, and have been extensively studied in recent years. Although many previous studies have proposed enhanced algorithms, each one has only considered either throughput or fairness. In this paper, we propose an efficient random-access mechanism called Multi-round Collision Avoidance (MrCA) that considers throughput and fairness together. The key idea in MrCA is to avoid collisions through multiple contentions, each with a smaller sized contention window. With this simple modification, we can significantly reduce the collision probability as well as the access delay, in addition to increasing fairness index. We find the collision probability and throughput analytically. Through simulation, we validate our analytical model and find appropriate parameters for achieving good performance. We also demonstrate that, compared to the IEEE 802.11 DCF, MrCA makes the collision probability extremely low, so that it increases throughput by 25% as well as short-term fairness by 50% with 50 contending nodes. When MrCA and 802.11 DCF schemes are combined with the auto rate fallback scheme, the performance gain of MrCA over 802.11 DCF increases because MrCA lowers the collision probability significantly, which makes channel error estimation more accurate. We also discuss the issues of implementation and backward compatibility.

1. Introduction

All wireless networks have their own random-access mechanisms that define network access and communication procedures. In a centralized network, each wireless node should register when communicating through a coordinator by using a random-access method. For example, a node registers with an access point (AP) in the basic service set (BSS) of a wireless LAN by using the carrier sense multiple access with collision avoidance (CSMA/CA), or to a base station (BS) in a cellular network by using slotted ALOHA. Since using random-access is required in wireless networks, many studies have proposed efficient random-access schemes for centralized networks, such as mobile WiMAX or 3GPP cellular series [9,10].

In distributed access networks like wireless LANs, efficiency is more important than in centralized networks. Most recent studies on distributed access networks have assumed that such networks use the IEEE 802.11 distributed coordination function (DCF) [1], which uses CSMA/CA as its channel access algorithm. Since the wireless LAN uses an unlicensed band, it uses the carrier sense function to avoid interference from other devices. In addition, since the wireless stations have no collision detection (CD) ability, they use a collision avoidance (CA) scheme to reduce the collision probability. This CSMA/CA approach, however, has high contention overhead. If we reduce the contention window to reduce the contention overhead, this increases collision overhead instead.

In this paper, we develop an efficient random-access method called Multi-round Collision Avoidance (MrCA),
which is based on the CSMA/CA algorithm. MrCA adopts Ahn’s basic idea of repetitive contention [4], in MrCA, only the users who win consecutive contention rounds have the right to transmit. The MrCA can lower the collision probability significantly by using multiple smaller sized contention windows. A multiple contention scheme has the same effect as using a very large single contention window, while maintaining lower contention overhead than single contention. Because of the low collision probability, MrCA improves channel utilization and data throughput compared to IEEE 802.11 DCF. In addition, MrCA also reduces delay jitter (defined as the standard deviation of delays). We define delay as the interval between two consecutive successful data frame transmissions at each node with the assumption of continuously backlogged stations. By reducing delay jitter, MrCA improves short-term fairness significantly, which is an important metric in supporting real-time traffic. We use the fairness index metric introduced by Jain et al. [26] to show the improvement in fairness. We investigate performance through simulations and analysis.

The paper is organized as follows. In Section 2, we briefly overview the IEEE 802.11 standard and some previous work, focusing on the effectiveness of contention mechanisms in random-access. Section 3 describes our MrCA scheme, which is also given in [5], and address implementation and backward compatibility issues. Section 4 models the MrCA scheme and analyzes the collision probability and the throughput of MrCA. In Section 5, we provide simulation results obtained from an event-driven simulator, and we conclude our paper in Section 6.

2. Background

Many studies have dealt with the efficacy of wireless MAC algorithms. Among these, IEEE 802.11 DCF has been most actively researched because of its simplicity and broad deployment in the field. In this section, we briefly survey IEEE 802.11 DCF and the popular rate adaptation scheme called auto rate fallback (ARF). The ARF is not a good link adaptation algorithm, but rather a simple heuristic one that can be well matched with MrCA scheme to achieve good performance. Finally, we discuss related work on contention based random-access schemes.

2.1. IEEE 802.11 DCF

DCF is a contention based distributed access scheme defined in IEEE 802.11[1]. The DCF uses carrier sense multiple access with collision avoidance (CSMA/CA). To access the shared wireless medium, each node that has a frame to transmit picks a randomly chosen backoff counter, \( b \), in the range \([0, CW]\), where \( CW \) is the size of the contention window. \( CW \) is expressed as \( 2^x - 1 \), where \( x \) is a positive integer. \( CW \) varies as follows:

\[
\begin{align*}
CW & \leftarrow \min(2CW + 1, CW_{\text{max}}) & \text{for collision} \\
CW & \leftarrow CW_{\text{min}} & \text{for success}
\end{align*}
\]

Therefore, many consecutive collisions make \( CW \) very large, which incurs much contention overhead. The backoff counter \( b \) is decremented for each slot time when the medium is idle. If the medium becomes busy before the node finishes its backoff countdown, the node freezes the counter and waits until the medium becomes idle for a DCF Inter Frame Space (DIFS) time period again, and runs the timer continuously. When the backoff counter reaches zero, the node can access the channel and transmit a data frame or RTS (Request-to-Send) frame. At the end of data frame reception, the receiver node replies with an ACK frame after a Short Inter Frame Space (SIFS). By receiving the ACK frame, the sender releases the channel, and sets the \( CW \) value to \( CW_{\min} \) according to Eq. (1). The node now prepares for the next transmission. If a collision occurs, then the receiver will not generate an ACK frame. In this case, the sender doubles its contention window \( CW \) as shown in Eq. (1), and runs the backoff procedure again. If the number of consecutive transmission failures reaches a retransmission limit, then the node drops the frame.

To reduce collision overhead, DCF defines the RTS/CTS access method. When the payload is greater than RTS Threshold, the station transmits RTS frame prior to a data frame. The destination of the RTS frame replies with a CTS (Clear-to-Send) frame to confirm that the channel is clear. The source then transmits the data frame and will receive an ACK if the transmission is successful. The other nodes that overhear RTS or CTS frames should wait until the transmission is complete. Because the RTS frame length is short, the RTS/CTS mechanism reduces the time wasted by collisions.

2.2. ARF: Rate adaptation mechanism in IEEE 802.11

The MrCA improves the performance of the auto rate fallback (ARF) [25] mechanism.

The ARF mechanism is a heuristic open-loop rate adaptation algorithm for wireless LAN devices developed by Lucent Technologies. It adjusts the data rate by one level according to the number of consecutive ACKs received or timer expiration. If a node fails to transmit two consecutive frames, the node lowers the data rate to the next lower rate and resets the timer. If the timer expires or the source receives ten consecutive ACKs, then the algorithm raises the data rate by one level and resets the timer. However, if the very next data transmission with the increased data rate by the timer expiration fails, then algorithm lowers the data rate and restarts the timer.

Holland et al. determined experimentally that 60 ms was an appropriate value for the ARF timeout [18]. The ARF regards all consecutive ACK losses as channel error. This is, however, an overly simplistic approach in a dense network, since most ACK losses are due to collisions, not to channel error.

2.3. Related work

Bianchi analyzed the performance of IEEE 802.11 DCF using a discrete time Markov chain model [3]. In the analysis, Bianchi assumed that all the wireless nodes always have frames to transmit in their queues. Bianchi showed that the performance of the IEEE 802.11 DCF basic access method strongly depends on system parameters, such as
the minimum contention window size and the number of contending nodes. Also, if there are many nodes in the network, the network can be used for only 50% or less of the time for data transmission because of collision overhead. This observation motivates us to enhance the CSMA/CA algorithm to lower collision probability while maintaining the basic functionality of the CSMA/CA algorithm.

To reduce the overhead of the CSMA/CA algorithm, some studies have modified the binary exponential backoff (BEB) mechanism given in Eq. (1). Song et al. [6], as well as Vukovic and Smavatkul [7], proposed the Exponential Increase Exponential Decrease (EIED) backoff algorithm and analyzed its performance with a novel analytical model. This model confirmed the improved efficiency of the BEB algorithm, but does not guarantee fairness. Haas et al. [8] proposed a sensing backoff algorithm (SBA) that uses optimal analysis results without using environment-dependent parameters. It shows high fairness. The simulation results, however, show that the SBA consumes as much as 50% of channel throughput to maintain fairness even when there are very few nodes. The throughput drop is mainly caused by the idle time for waiting.

Compared to ALOHA, CSMA/CA is less fair, but achieves better throughput. Koksal et al. [12] present the short-term unfairness problem of the CSMA/CA algorithm. Since the CSMA/CA extends the backoff windows of colliding stations to lower the collision probability, the CSMA/CA is less fair in the short term. The authors also noted the trade-off relation between throughput and fairness. To enhance the shorter-term fairness, every node has to have a lower contention time under the same channel access probability. This increases the collision probability and lowers the throughput performance.

Various authors [19–21,18,22] have proposed schemes to improve random-access efficiency for IEEE 802.11 networks, multi-hop ad hoc networks, or other random-access networks. Some authors [13–17] have proposed schemes focusing on the fairness improvement of legacy DCF protocols. The fairness studies considered various issues such as up/down link fairness, temporal fairness, and throughput fairness. However, there is little work that attempts to improve both channel throughput and fairness.

HIPERLAN/1 [11] is a well-known multiple-round contention scheme. It first selects the highest priority stations, during the prioritization phase. Then, the selected stations transmit signal bursts for the duration of a random number of slots. If the channel is idle after signal bursting, they enter the yield phase, which has a normal backoff procedure in transmitting a data frame. However, it consumes too much energy in transmitting signals during the first round, and uses just two rounds of contention, which is not optimized for many possible cases.

In our previous work [5], we proposed the fundamental idea of MrCA with a simple analysis of the collision probability. Though the previous work showed good throughput and fairness performance, it lacks backward compatibility, so when MrCA stations share the same wireless access channel with legacy DCF stations, the MrCA stations starve the DCF stations. In this paper, we extend our proposed MrCA to be backward compatible to the legacy DCF scheme, and analyze the throughput performance accurately, including an analysis of the collision probability in finding the optimal parameter set.

3. Multi-round collision avoidance

In this section, we introduce our MrCA algorithm, which we originally presented in our previous work [5], in detail. MrCA improves the collision avoidance scheme of CSMA/CA in IEEE 802.11 DCF by using multiple shortened contention windows.

At the end of this section, we address the implementation issues and propose a new backward compatible version of MrCA (bMrCA). Since the original MrCA [5] utilizes the wireless channel more effectively than does the conventional DCF mechanism, MrCA stations can starve legacy DCF stations. Our newly proposed bMrCA resolves the coexistence problem with legacy DCF stations.

3.1. Multiple contention mechanism

In MrCA, each node has different contention windows and random backoff counters for $G$ contention rounds. Therefore, we express the contention windows and backoff counters as vectors, $\mathbf{CW}$ (whose elements are written as $\mathbf{CW}_i$) and $\mathbf{b}$ (whose elements are written as $b_i$), where $i \in \{1, 2, \ldots, G\}$ represents the contention round. At the beginning of each contention stage, each node picks $G$ random backoff counters in the range $[0, \mathbf{CW}_i]$, where $\mathbf{CW}_i$ varies according to a collision event.

In the first contention round, each node uses its own backoff counter $b_1$, and contends by following the same method as in CSMA/CA. Each winning node $^2$ transmits a tone signal $^3$ on the channel and enters the second contention round. The other nodes, after hearing the signal, freeze their backoff counters and wait for the next contention stage. In the same manner, only the winning nodes are allowed to proceed to the next round. Finally, the last winner $^4$ in the $G$th round transmits a data frame or an RTS frame.

Though the contention may be resolved before all $G$ rounds have finished, the winning node cannot perceive this situation. Therefore, even in this case, the node executes the contention resolution procedure for all $G$ rounds.

3.2. Freezing and dealing with a collision

If a node senses a tone signal, it freezes immediately and waits for the next contention stage. For instance, if a node hears the signal in the third round, it freezes immediately while keeping the first and second round backoff counters zero. At the next contention stage, the node immediately transmits two consecutive tone signals with

---

1. In this paper, we define 'a contention stage' as the time period from the beginning of the first round to the end of the last round.
2. The winning node is the node that finishes its backoff countdown earlier than any other nodes for that round. At the end of each round, several nodes can win together.
3. The tone signal duration is $\Delta\text{SlotTime} - \Delta\text{RxCxTurnaroundTime}$, that will be explained in Section 3.4.
4. Although the probability is very low, there can be several winners of the final round. In this case, collision occurs like in the IEEE 802.11 DCF algorithm.
zero backoff counters for the first and second contention rounds, and starts the third contention round shortly thereafter. This simple freezing mechanism gives higher priority to the delayed nodes, and improves delay jitter and fairness. We give an example in Section 3.3 to clarify the freezing mechanism.

Since the wireless channel is not clear and there may be jamming signals in the unlicensed band, it is possible for every node to freeze its counter by regarding a jamming signal as a tone signal in the middle of a multi-round contention phase. In this case, none of the backoff counters reach zero, so that no node will transmit the signal. If nodes are frozen by a jamming signal, they must have been frozen for the maximum of backoff slots, that is, \(2^{\left(\text{CW}_{\text{min}} + 1\right)}\), which cannot happen in a normal contention process. When this occurs, the nodes notice that an abnormal situation occurred, and exit from the frozen state by continuing the backoff process from the frozen round. In other words, the stations run the backoff process from the frozen round after \(2^{\left(\text{CW}_{\text{min}} + 1\right)}\) consecutive idle slots.

When a collision occurs during data transmission, the node that did not receive an ACK frame will expand its contention window. We also adopt the BEB algorithm here to reduce the collision probability. In the IEEE 802.11 DCF, the contention window is doubled for every collision as shown in Eq. (1), so that the contention window equals \(2^{\left(\text{CW}_{\text{min}} + 1\right)} - 1\), where \(j\) is the number of retransmissions. In \(\text{MrCA}\), we double the contention window for each retransmission. For the first retransmission, double the first window \(\text{CW}_1\), and for the second, double the \(\text{CW}_2\), and so on. Therefore, the \(\text{CW}_i\) is given as follows:

\[
\text{CW}_i = \begin{cases} 
2\text{CW}_{\text{min}} + 1, & \text{for } i \in \{i|0 < i < j\} \text{ when } j > 0, \\
\text{CW}_{\text{min}}, & \text{for } i \in \{i|j < i < G\}, 
\end{cases}
\]

where \(j\) is the retransmission trial number.

For example, let \(\text{CW}_{\text{min}} = 3, G = 4\), and \(\text{CW} = (\text{CW}_1, \text{CW}_2, \text{CW}_3, \text{CW}_4)\). Then, the \(\text{CW}\) is \((3,3,3,3)\) at the first contention trial \((j = 0)\). When the collision occurs, \(\text{CW}\) becomes \((7,3,3,3)\) for the second trial \((j = 1)\) and \((7,7,3,3)\) for the third, and so on. This doubling method doubles the total number of backoff counter vector candidates as the DCF doubling method does.

When the number of consecutive collisions \(j\) exceeds \(G\), the station doubles the first contention window again if the current contention stage is not the maximum contention stage. For the next collision, it doubles the second contention window again and so on. However, the maximum contention stage that is larger than \(G\) has little effect on performance because the probability of consecutive collision is very low in \(\text{MrCA}\).

### 3.3. Example operation of \(\text{MrCA}\)

In this subsection, we show a simple example of the \(\text{MrCA}\) algorithm. Fig. 1 depicts the case with three contesting stations. Here, \(G = 4\) and \(\text{CW}_{\text{min}} = 3\). Let \(b_n\) denote the backoff counter vector \((b_1, b_2, b_3, b_4)\) of node \(n\). In the figure, the time unit represents aSlotTime, and each four-square-bar represents the current random backoff counter vector at the given slot. Finally, the shaded square in the four-square-bar represents the current contending round.

At the end of the busy period plus a DIFS interval, all three stations start the first round contention by picking the backoff counter vector, \(b_n\), where \(n = 1, 2, 3\). In this example, the \(b_n\) values are randomly given as \((3,2,2,0)\), \((1,3,1,3)\), and \((1,3,2,1)\), in that order. After one slot, since the first round backoff counters of stations 2 and 3 become zero, they both transmit tone signals. Hearing the signals, station 1 freezes its backoff counter with \((2,2,2,0)\), and waits for the next contention stage.

![Fig. 1. Example of the MrCA algorithm.](image-url)
After the signal transmission, the second round contention starts immediately. In this round, only stations 2 and 3 can participate, because they have won in transmitting the signals in the previous round. The stations 2 and 3 happen to have the same second round backoff counters of 3. Therefore, after 3 slots, they transmit the signals simultaneously, and proceed to the third round together.

In the third round, station 2 finishes its backoff earlier than station 3, so station 3 should freeze its backoff counter with (0,0,1,1), and await the next contention stage. Finally, in the fourth round, \(b_2\) of station 2 reaches zero after 3 slots. Then station 2 gets the right to access the medium and transmits its data frame and receives an ACK frame if the data frame is successfully transferred.

After the transmission completion and a DIFS interval, the new contention stage starts. Station 2 resets its random counter vector to (3,1,0,2) for the next data frame in its queue and participates in the contention. Since \(b_3\) is (0,0,1,1), station 3 has higher priority than any other stations. These multiple-round contentions can lower the collision probability significantly.

### 3.4. Implementation issues and backward compatibility

Now, we discuss the implementation issues of the MrCA mechanism. To announce the winner of a round in MrCA, a station should transmit a tone signal for a duration of \(a_{\text{SlotTime}}\). The 802.11 standard [1] determines \(a_{\text{SlotTime}}\) by considering RX/TX turnaround time, process and propagation delays, and clear channel assessment (CCA) time. Therefore, after a station wins a contention round, it has enough time to switch its current RX mode to TX mode and to emit the tone signal within \(a_{\text{SlotTime}}\). More accurately, the signal duration is \(a_{\text{SlotTime}}-a_{\text{RxTxTurnaroundTime}}\).

Some stations happen to lead or lag in the current round due to missing the reception of a tone signal, joining of new traffic, or interference from hidden stations. In this case, there are two possibilities: **Case I**: A leading station transmits its payload while other stations are waiting for a tone signal, and **Case II**: A lagging station transmits a tone signal while other stations are waiting for a data transmission in the final round.

In **Case I**, the other stations consider the synchronization (SYNC) preamble of the physical layer convergence protocol (PLCP) header as consecutive tone signals. Since the PLCP SYNC field duration is much longer than a slot time, the other stations can synchronize with the transmitted frame of the leading station, although they skip a few bits of the SYNC field by considering the bits as the tone signal.

**Fig. 2** shows an example of **Case I**. In **Fig. 2**, station 1 is leading the round while the other stations are lagging the round compared to station 1. When station 1 ends its final round and transmits a data frame, the other stations are in the second round. The lagging stations consider the PLCP header of the data frame as the third and fourth rounds tone signals. At the end of the data transmission, the stations are automatically synchronized.

In **Case II**, the other stations consider the tone signal as a jamming signal and wait for Extended Inter-Frame Space (EIFS) time until they receive an error-free frame. Therefore, the lagging station can transmit its own frame after it transmits remaining tone signals. By hearing the transmitted frame, the other stations resume their contention.

When MrCA stations coexist with legacy DCF stations, they always lag while DCF stations lead, because the DCF stations are always on the final round. Since MrCA stations use a smaller contention window than that of DCF stations, **Case II** occurs more frequently than **Case I**. Therefore, the DCF stations may starve if the MrCA stations generate traffic continuously.

To solve this problem, we consider a variant of MrCA called backward compatible MrCA (bMrCA). The bMrCA
extends the first round contention window of MrCA while the contention windows of the other rounds use the same value as the original MrCA. We use $\text{CW}_\text{min}$ of the DCF scheme as the first round contention window $\text{CW}_1$ of bMrCA, so that at the beginning of every stage, bMrCA stations contend fairly with legacy DCF stations.

Still, the contention with DCF stations benefits bMrCA stations. When a station fails to transmit a data frame, DCF stations continuously double their contention windows while bMrCA stations double the first round contention window only for the first transmission failure, and the windows of the other rounds for the following failure. We address this issue with simulation results in Section 5.5.

4. Performance analysis

In this section, we analyze the MrCA scheme with simple modifications, calling this scheme the modified MrCA (mMrCA). In mMrCA, each station uses $[1, \text{CW}_1 + 1]$ as the first round contention window, while the other rounds use $[0, \text{CW}_r]$. This modification forbids stations that have just finished transmission from winning the first round sequentially, because the first round backoff counter $b_1$ cannot be set to zero for random selection. Therefore, no new stations can join the second round contention before resolving all the stations that have currently entered the second round.

For the analysis of the first round contention, we adopt Bianchi’s Markov chain model [3]. The analysis for the other rounds is included as a part of the transmission time analysis of the first round. This is made possible because we use the mMrCA here. In mMrCA, no stations can pick a ‘0’ backoff counter for the first round. Only the stations that entered the second round once can have a ‘0’ backoff counter for the first round. Therefore, there is no other station that can win the contention before the stations that had a ‘0’ backoff counter in the first round finish their contention process.

4.1. First round analysis and collision probability

Since the first round contention window doubles its window size for the first collision and maintains the doubled window size for the following collisions, we can model the Markov chain for the first round contention as shown in Fig. 3. Here, $W$ and $p$ represent the initial contention window and the collision probability, which is the ratio of collision events to the total number of transmissions in a single station, respectively. The MrCA freezes its backoff counter when the medium is busy, and continues with the frozen backoff counter after idle channel condition. This behavior is different from the one used by Bianchi [3], which starts with a backoff counter decreased by 1 after deferring. Thus we combine the state $(i, 0)$ and $(i, 1)$ as shown in Fig. 3 to reflect the ‘frozen start’ of the backoff procedure. When a station reaches the state $(i, 0)$, the station decrements its backoff counter from 1 to 0 and transmits a tone signal to enter the second round without a state transition. Since this station has no state transition, the other stations freeze the backoff counter and do not change their state, either.

Let the steady state distribution probability of state $(i, k)$ be $b_{i,k}$, where $i$ is the backoff stage and $k$ is the backoff counter. Using the same calculation procedures as Bianchi [3], the variable $\tau$, which represents the probability that a station enters the second round in a slot can be expressed as

$$\tau = b_{0,0} + b_{1,0} = \frac{2}{W(1 + p) + 1}. \quad (3)$$

To obtain $p$ and $\tau$, we need an additional equation. Let $n_A$ be the number of stations that enter the second round together with a station named ‘station $A$’. When the station $A$ transmits, we can calculate the probability of $P[n_A = k]$ as

$$P[n_A = k] = \binom{n - 1}{k} \tau^k (1 - \tau)^{n - k - 1}, \quad (4)$$

where $n$ is the total number of contending stations.

We now estimate the expectation value of the $r$th round contention window, $\mathbb{W}_r$. In the $r$th round, a station uses $2W$ with probability $p'$ which comes from $r$ consecutive retransmissions; otherwise, it uses $W$ as a contention window. Therefore we can obtain

$$\mathbb{W}_r = p'(2W) + (1 - p')W = (1 + p')W. \quad (5)$$

For simplicity, we assume the $\mathbb{W}_s$ are independent of each other. After entering the second round, the stations can have $\frac{\mathbb{C}_{\mathbb{W}_s}}{2^r}$ combinations of the backoff counters. We define the value $\mathbb{W}_s$, as $\mathbb{C}_{\mathbb{W}_s}$. To avoid collision, the other $k$ nodes have to avoid the slot used by the station $A$. The probability of the successful transmission of the station $A$ is now calculated as $\frac{1}{(\mathbb{W}_s - 1)/\mathbb{W}_s)^k}$. Therefore, we can calculate the collision probability $p$ as follows:

$$p = \sum_{k=0}^{n-1} P[n_A = k] \left(1 - \left(\frac{\mathbb{W}_s - 1}{\mathbb{W}_s}\right)^k\right). \quad (6)$$

Substituting Eqs. (3) and (4) into Eq. (6), we can obtain $p$. Since the equation is complex, it is hard to prove that it has a unique solution. Therefore, we used a computer program to find all possible solutions, and confirmed that there is only one solution.
4.2. Throughput analysis

Now we calculate the throughput using the value $p$ and the state duration of the first round in Fig. 3.

Let $n_t$ be the number of stations that enter the second round together at a certain slot after winning the first round contention. Then $P[n_t=k]$ can be expressed as

$$P[n_t=k] = \binom{n}{k} \tau^k (1-\tau)^{n-k}. \quad (7)$$

When $n_t = 0$, the channel is idle and the state duration is $\sigma$, which is the idle slot length. If $n_t > 0$, $n_t$ stations attempt to transmit data. The transmission can either succeed or fail. Let $T_k$ be the state duration when $n_t = k$. Then the average state duration is

$$\frac{\sum_{k=0}^{n} T_k P[n_t=k]}{P[n_t=k]}, \quad (8)$$

where $T_0 = \sigma$.

We now define the throughput, $S$, as the number of successfully transmitted payload bits per second under saturated traffic, which is a similar definition to that of Bianchi [3]. Before calculating the throughput $S$, we define the probability $P_t(k)$ as the probability of the transmission in one of the $W_t$ instances when $n_t = k$, and $P_s(k)$ as the probability of the successful transmission given the transmission with $P_t(k)$. Then we can obtain

$$P_t(k) = 1 - \left(\frac{W_t - 1}{W_t}\right)^k, \quad (9)$$

and

$$P_s(k) = k \left(\frac{W_t - 1}{W_t}\right)^{k-1}, \quad (10)$$

Using Eqs. (9) and (10), we can calculate $T_k$ for $k > 0$ as follows:

$$T_k = \sigma \sum_{r=2}^{\infty} \frac{W_t - 1}{k+1} + W_t P_t(k) P_s(k) T_S$$

$$+ (1 - P_s(k)) T_C \} \quad (k > 0) \quad (11)$$

where $T_S$ is the time for the successful transmission and $T_C$ is the time for the failed transmission. In Eq. (11), the idle time interval part is calculated by assuming that $k$ stations uniformly divide $W_t - 1$ contention slots for each round. For example, if $k = 1$, then the station in the second round uses $(W_t - 1)/2$ idle slots on average for contention. The definitions of $T_S$ and $T_C$ are the same as those of Bianchi [3], except that they include the times for $(G - 1)$ tone signal transmissions and one idle slot that is included in state $(i, 0)$ of the first round. Therefore we can express these as

$$T_S = \sigma G + (H + E[P]) / R + SIFS + ACK + DIFS$$

$$T_C = \sigma G + (H + E[P]) / R + DIFS, \quad (12)$$

where $H$ is the frame overhead $E[P]$, is the average payload length, $R$ is the link speed, and $ACK$ is the ACK transmission time. $SIFS$ and $DIFS$ follow the definitions in the IEEE 802.11 Direct Sequence Spread Spectrum (DSSS) PHY standard [1].

Then we obtain the throughput $S$ as

$$S = \frac{E[P] \sum_{k=1}^{n} P[n_t=k][W_t P_t(k) P_s(k)]}{\sum_{k=0}^{n} T_k P[n_t=k]}, \quad (13)$$

note that $T_0 = \sigma$.

4.3. Model validation

To validate the analytical model, we use an event-driven simulator for the wireless MAC protocol written in C++ [2]. In our simulator, we define various types of events to describe our MrCA algorithm in as much detail as possible. We provide our simulation code on our Web page [27].

For simulations, we use the same assumptions as did Bianchi [3]. That is, (1) every contending node has data frames to transmit, (2) each frame can be lost only due to collision, and (3) nodes are all located within a one-hop communication range from each other.

The simulation environment follows the IEEE 802.11 High Rate DSSS (HR/DSSS) parameters [1] that are summarized in Table 1. According to the standard, the PHY header is transmitted using the data rate of 1 Mbps that is minimal. Therefore, the transmission time for a PHY header is fixed at 192 μs. The MAC overhead includes three address fields and a frame check sequence (FCS) field.

MrCA uses the 4-round scenario with CW_{min} of 3. We set the maximum number of retransmission stages to 5. Since the collision probability is very low in MrCA, increasing the number of ‘maximum stages’ has little influence on the simulation results.

First, we plot the collision probability $p$ obtained from both analysis and simulation as shown in Fig. 4. It shows that the analysis and simulation results are very close to each other. The collision probability in mMrCA with any parameter setting is much lower than that in the IEEE 802.11 DCF.

Second, we compare the throughput results of analysis and simulation in Fig. 5. This figure shows that the analysis results are also almost the same as the simulation results. The gap in the ‘4 rounds with CW of 7’ comes from the assumption that all the random variables are independent and from the linear approximation of the window size for each round. However, the difference between the analytical and simulation results is still negligible.

4.4. Performance optimization

Using the analytical result, we can find the optimal value for the number of contention rounds with appropriate contention window size. In Fig. 6, we plot the throughput result using Eq. (13) according to the $W$ of 4, 6, and 8, which is CW_{min} + 1.

From Fig. 6, we can obtain the optimal number of contention rounds as shown in Table 2. As MrCA is not allowed
to change the value of $G$, we need to choose the appropriate number of rounds considering overall performance. Throughout the simulation study, we choose ‘4 rounds with $CW_{\text{min}} = 3$’ as a default setting, as usually, few users are contending within a network. For a slightly more dense network, such as more than five contending nodes, ‘3 rounds with $CW_{\text{min}} = 7$’ is a suitable choice.

### 5. Simulation results

We run further simulations to show the results for throughput, delay, and fairness, as well as ARF behavior (introduced in Section 2.2). We also simulate the MrCA
with the RTS/CTS option. The assumptions used in the simulations are the same as those in Section 4.3 except for the ARF simulation, which implements the path loss channel model.

5.1. Throughput performance

Fig. 7 shows the throughput performance of MrCA, mMrCA and the original IEEE 802.11 DCF using a 5.5 Mbps PHY data rate. In the simulation results, MrCA and mMrCA show similar throughput performance, and outperform the IEEE 802.11 DCF. Here, MrCA and mMrCA in Fig. 7b–d use the default setting of $G = 4$ and $CW_{\min} = 3$.

When the number of contending nodes increases to 50, MrCA shows a throughput improvement of about 25% compared to the IEEE 802.11 DCF in Fig. 7a. In Fig. 7a, we also compare two different sets of MrCA schemes. The 3-round MrCA has good overall performance while the 4-round MrCA performs better for fewer than seven contending nodes. This is because of the overhead caused by longer idle waiting times in the 3-round MrCA with $CW_{\min} = 7$.

Fig. 7b shows the simulation results for all three schemes with the RTS/CTS mechanism with 1500 bytes payload size. From Fig. 7a and b, we observe that IEEE 802.11 DCF with the basic access method performs better than the RTS/CTS mechanism when the number of contending nodes is less than ten. In contrast, RTS/CTS performs better if the number of contending nodes exceeds ten. The reason is due to the reduced collision overhead in the RTS/CTS mechanism. However, MrCA always performs better if we use the basic access mechanism. Because MrCA suffers from few collisions, it does not need RTS/CTS procedures. Though MrCA and IEEE 802.11 DCF with RTS/CTS show similar throughput performance, the results for delay jitter and fairness are quite different, which we will show in the next subsections.

In Fig. 7c, we simulated a smaller payload size, which mitigates the collision overhead. It shows that MrCA still outperforms IEEE 802.11 DCF. Similarly to the 1500-byte-payload case, Fig. 7d shows that all three schemes have similar performance when using RTS/CTS packets for smaller payloads. In some cases, DCF has slightly better

<table>
<thead>
<tr>
<th>$CW_{\min}$</th>
<th>The number of users (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2 Optimized number of rounds.

Fig. 7. Aggregated throughput.
performance than MrCA. These cases show the tone signal overhead of MrCA as we lower the collision overhead by using the RTS/CTS and shorter payload. In MrCA, the shortest possible contention time is the tone signal transmission time, while it is zero in DCF. However, the tone signal overhead is too low to affect the overall performance of MrCA.

5.2. Delay and delay jitter performance

Fig. 8 plots the simulation results for delay performance. We define ‘delay’ as the interval between two consecutive successful data transmissions: in other words, the waiting time at the head of the MAC layer queue. In Fig. 8a, because of the collision overhead, the delay performance in IEEE 802.11 DCF is 20% worse than that in MrCA. There is a small gap in the performance between MrCA and mMrCA, which comes from the slight modification of ‘a slot time’ of the first round in mMrCA.

With the RTS/CTS mechanism, all three mechanisms exhibit the same delay performance in Fig. 8b. This is similar to the result in Fig. 7b. Comparing Fig. 8a and b, the RTS/CTS mechanism has little impact on the delay performance in MrCA while it improves delay performance in IEEE 802.11 DCF.

Another advantage of our scheme is improved fairness: that is, decreasing the delay jitter, defined as the standard deviation of the delays. If the access mechanism is perfectly fair, then the delay for every frame is the same, so the standard deviation approaches zero. On the other hand, if one node waits a long time while another node transmits several frames continuously, the delay jitter increases. Therefore, the less delay jitter a network has, the more fair it is.

Fig. 9 shows the delay jitter of each case. Comparing to the average delay in Fig. 8, the delay jitter in IEEE 802.11 DCF is very high (Fig. 9a and b). In the case of MrCA and mMrCA, however, the delay jitter is much lower than that in IEEE 802.11 DCF. This proves that the MrCA is much fairer than the IEEE 802.11 DCF.

In Fig. 9, we can also see the difference between MrCA and mMrCA. With a little modification of MrCA, mMrCA reduces the delay jitter by about 30%, compared to MrCA. Again although three schemes with RTS/CTS show similar performance in throughput and delay, they show a clear difference in delay jitter performance. This proves that MrCA and mMrCA mechanisms work well in all circumstances.

5.3. Throughput fairness index

Fig. 10 shows the throughput fairness index, which follows the definition of \( \left( \frac{\sum_{i=1}^{n} x_i}{n} \right)^2 / \left( \frac{\sum_{i=1}^{n} x_i^2}{n} \right) \) in [26], where \( x_i \)
represents the throughput of node $i$. The closer the fairness index is to one, the fairer the system is. In this simulation, we use 10, 30, and 50 contending nodes and vary the simulation duration\footnote{The duration is not the program running time, but rather, the time used in the simulation.} to observe the short term and long term fairness. As the simulation duration gets longer, the fairness index indicates the long term fairness. We averaged 50 simulation runs for each point.

The simulation results show that MrCA and mMrCA improve both short term and long term fairness compared to IEEE 802.11 DCF. Especially, MrCA and mMrCA improve the short term fairness significantly compared to IEEE 802.11 DCF. Even when the number of contending nodes increases to 50, the short term fairness index stays high, while that of IEEE 802.11 DCF decreases seriously. This improvement has a critical impact on supporting real-time services like VoIP. Since using the RTS/CTS mechanism shows a similar tendency, we omitted the simulation results for the RTS/CTS mechanism.

5.4. The ARF performance

In this subsection, we show the throughput performance from applying the ARF mechanism to the MrCA and 802.11 DCF schemes. Because the ARF regards all consecutive frame losses as incurred by channel error, it unnecessarily decreases the data rate for consecutive collisions.

In Fig. 11, we simulate by using the same Bit Error Rate (BER) vs. Signal-to-Noise Ratio (SNR) curves as given in [23]. We also used the path loss model with the exponent 3.32 and background noise $-92$ dBm [24]. An AP is located at the center, and mobile nodes move away from the AP with the speed of 0.1 m/s. All the nodes always have unidirectional traffic destined for the AP. The simulation results are the moving-averaged throughput for the last 10 s. The solid lines represent the MrCA, and the dotted lines the IEEE 802.11 DCF.

MrCA adapts to the channel very precisely until the number of contending nodes reaches 10. For 25 nodes, the throughput of MrCA fluctuates around 4 Mbps and decreases gradually for the mobiles that are further away from the AP. For 50 nodes, the throughput is around 1 Mbps until the mobiles are 150 m away from the AP.

On the other hand, the IEEE 802.11 DCF uses an optimal data rate only when there is one mobile node. If there are five contending nodes, then the throughput goes down to 2 Mbps. For ten or more nodes, the throughput is less than 1 Mbps. The performance gap between MrCA and IEEE

![Fig. 10. Throughput fairness index (5.5 Mbps with 1500 bytes).](image-url)

![Fig. 11. ARF performance.](image-url)
802.11 DCF under the ARF scheme comes from the different collision probabilities. The low collision probability in MrCA makes the link estimation scheme of ARF more accurate.

5.5. Coexistence of DCF and bMrCA stations

To show the backward compatibility issue, we simulate the coexistence environment. Figs. 12 and 13 shows the per-station throughput, which is the throughput observed at a station. In the simulation, while fixing the total number of stations, we vary each number of DCF and MrCA (or bMrCA) stations. The horizontal axis represents the number of DCF stations. The leftmost plot represents MrCA/bMrCA only performance, while the rightmost plot shows DCF-only performance.

Figs. 12a and 13a show the per-station throughput for original MrCA and DCF stations. Even one MrCA station lowers the throughput of DCF stations compared to that of the DCF-only environment. This problem can be solved by using the bMrCA scheme instead of the original MrCA scheme. The throughput of DCF stations is almost the same as that of the DCF-only environment in all the cases as shown in Figs. 12b and 13b.

However, we still have a slight performance degradation with DCF stations. When 30 stations share the channel, the per-station throughput in DCF is lower than in the case of 10 stations. The reason is that the bMrCA still takes advantage of the collision event because it doubles the contention window on the second round, $CW_2$, at the second collision, while DCF uses the contention window of $4 \times CW_{\text{min}}$. Fortunately the per-station throughput in DCF is still acceptable. Therefore we can conclude that the bMrCA is backward compatible with legacy 802.11 while maintaining its excellent throughput performance.

6. Conclusion

In this paper, we presented an efficient and fair contention method called MrCA. By achieving extremely low collision probability, MrCA showed very high throughput compared to the legacy IEEE 802.11 DCF scheme. In addition, MrCA improved short term fairness as well as long term fairness. In legacy DCF, the fairness problem comes from the BEB mechanism. Some stations double their contention windows, while others keep minimum contention
windows. It makes the difference in the sizes the contention windows of different stations larger. Thus, the fairness problem arises. However, since there are few collisions in MrCA, we can solve the fairness problem while increasing throughput.

Though MrCA scheme showed its superiority to other competitive schemes in terms of throughput, the backward compatibility problem still remained. With a simple modification of MrCA, we solved this problem. In bMrCA, MrCA stations can coexist with legacy DCF stations in a reasonably fair manner.

The MrCA can be also applied to the random-access channel in centralized networks, such as UMTS, 3GPP, or WiMAX systems. It will be used to reduce the access delay greatly while lowering the collision probability. In future work, we will develop an adaptive MrCA, where MrCA stations adaptively change the number of rounds $G$ according to the network size to improve performance.

References


[11] Radio Equipment and System (RES); High Performance Radio Local Area Networks (HIPERLAN); Type 1; Functional Specification, ETSI, ETS 300 652, October 1996.


[27] The Source code of MrCA Simulator, Online Link: <http://netlab.snu.ac.kr/~jyun/files/MrCA-sim.tar.gz>.

Jeongkyun Yun received the B.S. and M.S. degrees from School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea, in 2001 and 2003, respectively, and the Ph.D. degree in Electrical Engineering and Computer Science in 2008. He is currently working in LG Electronics as a senior research engineer. His research interests include future wireless LAN protocols, next generation cellular networks, and ad hoc networks.

Jin-Ghoo Choi received the B.S. and M.S. degrees in Electrical Engineering from Seoul National University in 1998 and 2000, respectively, and the Ph.D. degree in Electrical Engineering and Computer Science in 2005. In 2005, he worked for KT, Korea, as a cell planning engineer. From 2006 to 2007, he was a Senior Engineer in Samsung Electronics. He is currently a postdoctoral researcher at The Ohio State University. His research interests include packet scheduling, channel access, and resource management in wireless networks.
Saewoong Bahk received B.S. and M.S. degrees in Electrical Engineering from Seoul National University in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania in 1991. From 1991 to 1994 he was with AT&T Bell Laboratories as a member of technical staff where he worked for AT&T network management. In 1994, he joined the school of electrical engineering at Seoul National University and currently serves as a professor. He has been serving as TPC members for various conferences including ICC, GLOBECOM, INFOCOM, PIMRC, WCNC, etc. He is on the editorial board of Journal of Communications and Networks (JCN). His areas of interests include performance analysis of communication networks and network security. He is an IEEE senior member and a member of Who’s Who Professional in Science and Engineering.