A Developed Collision Resolution Algorithm in MAC Protocol for IEEE 802.11b Wireless LANs (ICEIC'04)

Kyung-Taek Chung** , Ce Pan*, Hyun Park*, Byun-Gon Kim*and Byoung-Sil Chon*

* Dept of Electronic Engineering, Chonbuk National University, Jeonju, 561-756, Korea
Tel : +82-63-274-7126 Fax : +81-63-270-2400 E-mail: pancewh@hotmail.com
**Dept of Electronic Engineering, Kunsan National University, Kunsan, 573-701, Korea
Tel : +81-63-469-4685 Fax : +81-63-469-4690 E-mail: eoe604@kunsan.ac.kr

Abstract

Design of efficient Medium Access Control (MAC) protocols with both high throughput performances is a major focus in distributed contention- based MAC protocol research. In this paper, we propose an efficient contention- based MAC protocol for wireless Local Area Networks, namely, the Developed Collision Resolution (DCR) algorithm. This algorithm is developed based on the following innovative ideas: to speed up the collision resolution, we actively redistribute the backoff timers for all active nodes; to reduce the average number of idle slots, we use smaller contention window sizes for nodes with successful packet transmissions and reduce the backoff timers exponentially fast when a fixed number of consecutive idle slots are detected. We show that the proposed DCR algorithm provides high throughput performance and low latency in wireless LANs.

Keywords: MAC, Backoff, DCR algorithm

1. INTRODUCTION

Distributed contention-based MAC protocol research in wireless networks start with ALOHA and slotted ALOHA in the 1970s. Later, MACA, MACAW, FAMA and DFWMAC were proposed by incorporating the Carrier Sense Multiple Access (CSMA) technique as well as the RTS and CTS handshaking mechanism for Collision Avoid (CA). The most popular contention-based wireless MAC protocol, CSMA/CA, has become the basis of the MAC protocol for the IEEE 802.11b standard. However, it is observed that if the number of active user increases, the throughput performance of IEEE 802.11b MAC protocol degrades significantly because of the excessively high collision rate. Many researchers have focused on analyzing and improving the performance of the IEEE 802.11b MAC.

To increase the throughput performance of a distributed contention-based MAC protocol, an efficient collision resolution algorithm is needed to reduce the overheads in each contention cycle. In this paper, we propose a new efficient distributed contention-based MAC algorithm, namely, the Developed Collision Resolution (DCR). We observed MAC algorithms comes from packet collisions and the wasted idle slots due to backoffs in each contention cycle. The DCR algorithm attempts to resolve the collisions quickly by increasing the contention window sizes of both the colliding stations and the deferring stations in the contention resolution.

This paper is organized as follows. In the next chapter, we briefly describe the IEEE 802.11b standard with MAC protocol. In chapter 3, the proposed Developed Collision Resolution (DCR) algorithm, and chapter 4 performance evaluations are presented, and in the final chapter, we present the conclusions.

2. IEEE 802.11b MEDIUM ACCESS CONTROL

The most popular contention-based medium access control (MAC) protocol is the carrier senses multiple access/collision avoidance (CSMA/CA), which is widely used in the IEEE 802.11b LANs.

A packet transmission cycle consists of a successful packet transmission by a source station followed by an acknowledgment (ACK) from the destination station. General operations of the IEEE 802.11b MAC protocol are as follows (we only consider distributed coordination function (DCF) without RTS-CTS handshake for simplicity). If a station has a packet to transmit, it will check the medium status by using the carrier sensing mechanism. If the medium is idle, the transmission may proceed. If the medium is determined to be busy, the station will defer until the medium is determined to be idle for a DCF inter-frame space (DIFS) and the backoff procedure will be invoked. The station will set its backoff timer to a random backoff time based on the current contention window size (CW):

Backoff Time (BT) = Random() x aSlotTime (1)

where Random() is an integer randomly chosen from a uniform distribution over the interval [0,CW-1].

After DIFS idle time, the station performs the backoff procedure using the carrier sensing mechanism to determine whether there is any activity during each backoff slot. If the medium is determined to be idle during a particular backoff slot, then the backoff procedure shall decrement its backoff time by a slot time. If the medium is determined to be busy at any time during a slot, then the backoff procedure is suspended. After the medium is determined to be idle for DIFS period, the backoff procedure is allowed to resume. Transmission shall begin whenever the backoff timer reaches zero. After a source station transmits a packet to a destination station, if the source station transmits a packet to a destination station, if the source station receives an acknowledgment (ACK) without errors after the short inter-frame space (SIFS) idle period, the transmission procedure is determined to be successfully completed. In this case, the contention window (CW) for this source station shall be reset to the initial (minimum) value minCW. If the transmission is not successfully completed, the contention window (CW) size shall be increased. This process is called binary exponential backoff (BEB), which resolves collisions in the contention cycle. More detailed operations can be found in ([4]).

3. DEVELOPED COLLISION RESOLUTION ALGORITHM FOR WLANs

3.1 The Basic Idea of Developed Collision Resolution

There are two major factors affecting the throughput performance in the IEEE 802.11b MAC protocol: transmission failures (we only consider failures due to packet collisions) and the idle slots due to backoff at each contention period.

Under high traffic load (i.e., all M stations always have packets to transmit) and under some ergodicity assumption, we can obtain the following expression for the throughput (for example, based on Figure 4.1, we can examine one transmission cycle):

$$\rho = \frac{\overline{m}}{E[N_c](E[B_c]\cdot t_s + \overline{m} + DIFS) + (E[B_c]\cdot t_s + \overline{m} + SIFS + ACK + DIFS} \quad (1)$$

where $E[N_c]$ is the average number of collisions in a virtual transmission time (or a virtual transmission cycle), $E[B_c]$ is the average number of idle slots resulting from backoff for each contention period, t_s is the length of a slot (i.e., aSlotTime), and $\overline{m}$ is the average packet length.

For this result, the best scenario would be the following: a successful packet transmission must be followed by another packet transmission without any overheads, in which case, $E[N_c] = 0$, $E[B_c] = 0$, the throughput would be

$$\rho_{best} = \frac{\overline{m}}{\overline{m} + SIFS + ACK + DIFS} \quad (2)$$

This can be achieved only when a perfect scheduling is provided: in such a scenario, each station shall have the probability of packet transmission, $p_{trans}(i)$, at each contention period as follows:

$$p_{trans}(i) = \begin{cases} 1 & \text{if station i transmits its packet at current contention period} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

This implies that if the current packet transmission right is assigned to station *i*, then only station *i* will transmit and all other stations will defer their packet transmissions.

Suppose that under some random backoff schemes, we could assume that the backoff timer is chosen randomly, then the probability of packet transmission for station *i* during the current contention period would depend on the backoff timer:

$$p_{trans}(i) = \frac{1}{(B_i + 1)} \quad (4)$$

where B_i is the backoff timer of station *i*. This means that if station *i* has the backoff timer 0 (i.e., $B_i = 0$), then its backoff time is 0 (i.e., $BT = B_i \times aSlotTime = 0$) and station *i* will transmit a packet immediately. Therefore, this can be interpreted to imply that station *i* has the probability of packet transmission of 1 at current contention period. If station *i* has the backoff timer ∞, then its backoff time is also ∞, which can be interpreted to mean station *i* has the probability of packet transmission of 0 at current contention period. From this discussion, (3) can be converted to (5):

$$B_i = \begin{cases} 0 & \text{if station i transmits its packet at current contention period} \\ \infty & \text{otherwise} \end{cases} \quad (5)$$

Thus, we conclude that if we could develop a contention-based MAC algorithm, which assigns a backoff timer 0 to the station in transmission while assigning all other stations' backoff timers ∞ for each contention period, then we could achieve the perfect scheduling.

leading to the maximum throughput. Unfortunately, such a contention-based MAC algorithm does not exist in practice. However, this does provide us the basic idea how to improve the throughput performance in the MAC protocol design. We can use the operational characteristics of the perfect scheduling to design more efficient contention-based MAC algorithm to approximate the behavior of perfect scheduling.

3.2 Developed Collision Resolution Algorithm

As we mentioned before, the major deficiency of IEEE 802.11b MAC protocol comes from the slow collision resolution as the number of active stations increases. In the Developed Collision Resolution (DCR) algorithm, we will change the contention window size for the deferring stations and regenerate the backoff timers for all potential transmitting stations to actively avoid "future" potential collisions. In this way, we can resolve possible packet collisions quickly. More importantly, the improved algorithm preserves the simplicity for implementation like the IEEE 802.11b MAC.

The DCR algorithm has the following characteristics:

1) Use much smaller initial (minimum) contention window size *minCW* than the IEEE 802.11b MAC;
2) Use much larger maximum contention window size *maxCW* than the IEEE 802.11b MAC;
3) Increase the contention window size of a station when it is in either collision state and deferring state;
4) Reduce the backoff timers exponentially fast when a prefixed number of consecutive idle slots are detected.

The detailed DCR algorithm is described as follows according to the state a station is in:

1) *Backoff Procedure*: All active stations will monitor the medium. If a station senses the medium for a slot, then it will decrement its backoff time (BT) by a slot time, i.e., $BT_{new} = BT_{old}$ – aSlotTime (or slot). When its backoff timer reaches to zero, the station will transmit a packet. If there are [(minCW + 1) × 2 -1] consecutive idle slots being detected, its backoff timer should be decreased much faster (say, exponentially fast), i.e., $BT_{new} = BT_{old} / 2$ (if BT_{new} < aSlotTime, then BT_{new} = 0). The net effect is that the unnecessary wasted idle backoff time will be reduced when a station runs out of packets for transmission.
2) *Transmission Failure (Packet Collision):* If a station notices that its packet transmission has failed possibly due to packet collision (i.e., if fails to receive an acknowledgement from the intended receiving station), the contention window size of the station will be increased and a random backoff time (BT) will be chosen, i.e., $CW = min(maxCW, CW\times 2)$, $BT = uniform(0, CW-1)\times aSlotTime$, where $uniform(a, b)$ indicates a number randomly drawn from the uniform distribution between a and b and CW is the current contention window size.
3) *Successful Packet Transmission:* If a station has finished a successful packet transmission, then its contention window size will be reduced to the initial (minimum) contention window size *minCW* and a random backoff time (BT) value will be chosen accordingly, i.e., $CW = minCW$, $BT = uniform(0, CW-1)\times aSlotTime$.
4) *Deferring State:* For a station which is in deferring state, whenever it detects the start of a new busy period, which indicates either a collision or a packet transmission in the medium, the station will increase its contention window size and pick a new random backoff time (BT) as follows: $CW = min(maxCW, CW\times 2)$, $BT = uniform(0, CW-1)\times aSlotTime$.

In the DCR algorithm, the station that has successfully transmitted a packet will have the minimum contention window size and smaller backoff timer, hence it will have a higher probability to gain access of the medium, while other stations have relatively larger contention window size and larger backoff timers. After a number of successful packet transmissions for one station, another station may win a contention and this new station will then have higher probability to gain access of the medium for a period of time.

4. SIMULATION AND PERFORMANCE EVALUATION

In this section, we present the simulation studies for the proposed DCR algorithms and the IEEE 802.11b MAC protocol using DSSS specification. The simulation tool is OPNET. The parameters used in the simulations are shown in Table 4.1, which are based on the IEEE 802.11b network configurations.

Table 4.1 Initial simulation parameters

	WLAN_DCR	WLAN_MAC
SIFS (μ s)	30	30
DIFS (μ s)	950	950
Slot Time (μ s)	460	460
Bit Rate (Mbps)	2	2
minCW	3	31
maxCW	2047	1023
Packet size (bytes)	1250	1250
Retry Counter	9	7
Work stations	10	10
Simulation duration (min)	2	2

From figure 4.1, we can find from the third second the average network throughput which using DCR algorithm increases much more than the throughput which non-using DCR a lgorithm. After the twenty s econds, the average throughput which using DCR algorithm becomes stably.

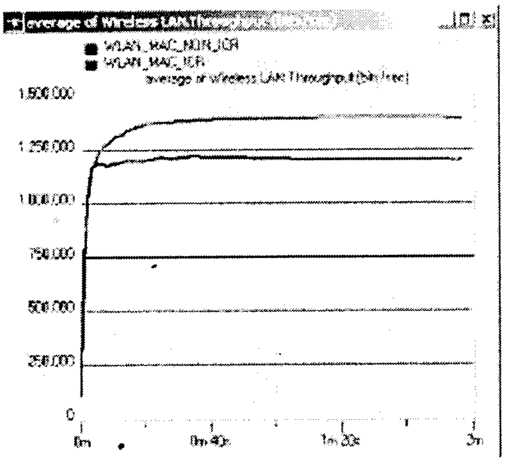

Fig 4.1. Average throughput comparison results.

Figure 4.2 is the comparison result of the average network load of these two algorithms. From this figure we can see that at the beginning of simulation, i.e. in a very shot time (0s-2s), the network goes to the highest load, following it will become stably. Comparing these two figures, we get the average network load comparison figure. Because using smaller minimum CW and larger maximum CW in DCR algorithm, the probability of collisions becomes lower than non-using DCR algorithm, the network load becomes lower than before.

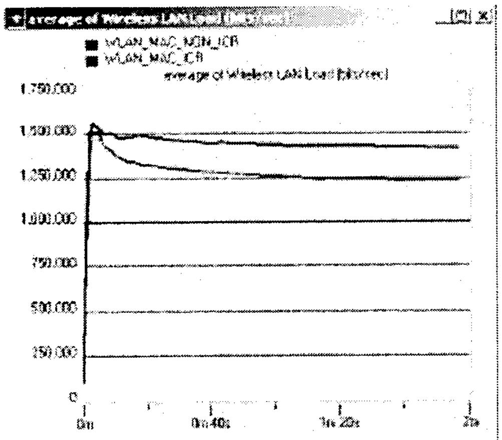

Fig 4.2. Average load comparison results.

Figure 4.3 illustrate the delay simulation results, after about 25 seconds, the delay which using DCR algorithm always shorter than 2 seconds, but the delay which non-using DCR algorithm almost longer than 2 seconds, and comparing the average delay results, I find actually, the average delay of using DCR algorithm is shorter than 1.75 seconds, much shorter than the average delay of non-using DCR algorithm after 7 seconds.

From all of the performance analyses above, we can say the IEEE 802.11b MAC algorithm without DCR shows very poor performance, and as the number of stations increases such as using 50 stations, the DCR algorithm will improve the WLAN performance more efficiently. Because in DCR algorithm, all stations except the one with successful packet transmission will increase their contention window size whenever the system has either a successful packet transmission or has a collision. This means all stations can quickly obtain the proper contention window size to prevent the future collisions, consequently the probability of collisions will be decreased to quite small values. At the same time, a station with a successful packet transmission has the minimum window size 3, which is much smaller than the minimum contention window size in IEEE 802.11B MAC algorithm (minCW = 31). This will reduce the wasted medium idle time to a much smaller value when compared to the IEEE 802.11Bb MAC without DCR algorithm.

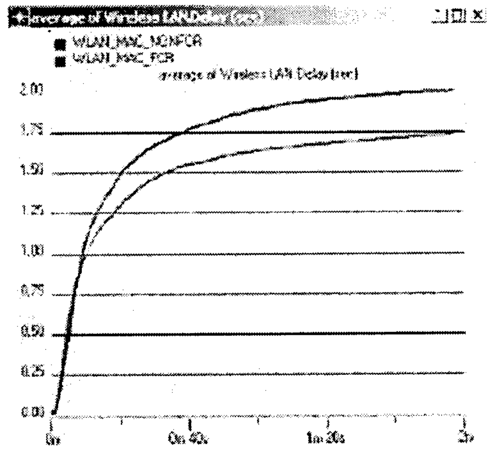

Fig 4.3. Average delay comparison results.

5. CONCLUSIONS

In this paper, we improve on a contention-based medium access control algorithm. The improved DCR algorithm can achieve high performance while preserving the implementation simplicity in wireless local area networks. In the improved DCR algorithm, each station changes the contention window size upon both successful packet transmissions and collisions for all active stations in order to redistribute the backoff timers to actively avoid potential future collisions. Due to this operation, each station can more quickly resolve collisions when there are a large number of active stations in the wireless LANs, other ideas in the DCR are to use much smaller minimum contention window size comparing to IEEE 802.11b MAC and fast decreasing backoff timers after detecting a fixed number of idle slots. These changes reduce the average number of idle slots in each contention period, which contributes to the performance improvement.

References

[1] V. Bharghavan, "MACAW: A Media Access Protocol for Wireless LAN's," SIGCOMM'94, pp.212-225, London England, Aug. 1994.

[2] V. Bharghvan, "Performance evaluation of algorithms for wireless medium access," *IEEE international Computer Performance and Dependability Symposium* IPDS'98, pp.142-149, 1998.

[3] G. Bianchi, "Performance Analysis of the IEEE 802.11B Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, Vol.18, No.3, PP.535-547 Mar. 2000.

[4] IEEE 802.11B Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, *IEEE*, 1997..

[5] K. Kim, S. Shin, and K. Kim, "A novel MAC scheme for prioritized services in IEEE 802.11Ba wireless LAN," *ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, Joint 4th IEEE International Conference*, pp.196-199, 2001