Fast Collision Resolution (FCR) MAC Algorithm for Wireless Local Area Networks

Younggoo Kwon, Yuguang Fang, and Haniph Latchman
Department of Electrical & Computer Engineering
University of Florida
Gainesville, Florida 32611-6130

Abstract—**Development of efficient medium access control (MAC) protocols is a fundamental research issue in high-speed wireless local area networks (LANs). In this paper, we focus on the throughput efficiency of medium access algorithms for high-speed wireless LANs which use Carrier Sense Multiple Access/Collision Avoidance(CSMA/CA). We propose an efficient distributed contention-based MAC protocol for wireless local area networks, namely,the *Fast Collision Resolution* (FCR) algorithm, and show that the proposed FCR algorithm provides high throughput and low latency in wireless LANs. The performance of FCR algorithm is compared with that of the IEEE 802.11 MAC algorithm via extensive simulation studies. The results show that FCR algorithm achieves a significantly higher efficiency than the IEEE 802.11 MAC algorithm and is easy to implement in wireless LANs.**

I. Introduction

Distributed contention-based MAC protocol research in wireless networks started 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 avoidance (CA) ([1], [6], [8] and references therein). The most popular contention-based wireless MAC protocol, CSMA/CA, becomes the basis of the MAC protocol for the IEEE802.11 standard[10]. However, it is observed that if the number of active users increases, the throughput performance of the IEEE802.11 MAC protocol degrades significantly because of the excessively high collision rate. Many researchers have focused on analyzing and improving the performance of the IEEE802.11 MAC (see for example [2], [3], [4] and references therein).

To increase the throughput performance of a distributed contention-based MAC protocol, an efficient collision resolution algorithm is needed to reduce the overheads (such as packet collisions and idle slots) in each contention cycle. To this end, many novel collision resolution algorithms have been proposed. For example, improved backoff algorithms are proposed to adjust the increasing and decreasing factors of the contention window size and the randomly chosen backoff values; the out-band busy-tone signaling is used to actively inform others for the busy channel status; and the contention information appended on the transmitted packets can also serve the purpose to help the collision resolution[1], [2], [7], [8]. Among these lines, Cali, Conti, and Gregori[4] proposed an interesting algorithm to improve the performance of the IEEE 802.11 MAC protocol. Their basic idea is to dynamically adjust the proper contention window size at each station based on the estimation of the number of active stations. However, in real wireless local area networks, it is not an easy task to estimate the number of active stations at run time.

This work was supported in part by the Office of Naval Research Young Investigator Award under grant N000140210464.

Although many innovative distributed contention-based MAC protocols have been proposed, it is not an easy task to satisfy all desirable properties while preserving the simplicity of implementation in real wireless LANs. In this paper, we propose a new efficient distributed contention-based MAC algorithm, namely, the *fast collision resolution* (FCR) algorithm. We observe that the main deficiency of most distributed contention-based MAC algorithms comes from the packet collisions and the wasted idle slots due to backoffs in each contention cycle. For example, in the IEEE 802.11 MAC protocol, when the number of active stations increases, there are too many stations backed off with small contention windows, hence many retransmission attempts will most likely collide again in the future, which would slow down the collision resolution. In this regard, the FCR algorithm attempts to resolve the collisions quickly by increasing the contention window sizes of both the colliding stations and the deferred stations due to prior loss in the contention procedure, i.e., we devise an algorithm so that all active stations will redistribute their backoff timers to avoid possible "future" collisions. To reduce the number of idle slots, the FCR algorithm gives a small idle backoff period for each station with successful packet transmission. Moreover, when a station detects a number of idle slots, it will start to reduce the backoff timer exponentially, comparing to the linear decrease in backoff timer in the IEEE 802.11 MAC. We attempt to keep the proposed distributed contention-based MAC easily implementable in real wireless local area networks.

This paper is organized as follows. In the next section, we describe the IEEE 802.11 MAC protocol. Then we present, in Section III, the newly proposed fast collision resolution (FCR) algorithm. Performance evaluations via simulative study for FCR algorithm is presented in Section IV. In the final section, we present the conclusions.

II. IEEE 802.11 Medium Access Control (MAC)

As we mentioned before, the most popular contention-based medium access control (MAC) protocol is the carrier sense multiple access/collision avoidance (CSMA/CA), which is widely used in the IEEE 802.11 LANs. The basic operations

0-7803-7632-3/02/$17.00 ©2002 IEEE

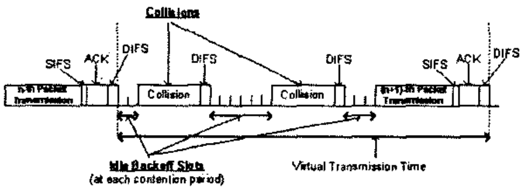

Fig. 1. Basic operations of CSMA/CA

of the CSMA/CA algorithm are shown in Fig. 1.

A packet transmission cycle is accomplished with a successful transmission of a packet by a source station with an acknowledgment (ACK) from the destination station. General operations of the IEEE 802.11 MAC protocol are as follows (we only consider the 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 distributed coordination function 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):

$$\text{Backoff Time (BT)} = \text{Random()} \times \text{aSlotTime} \quad (1)$$

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 by using the carrier sensing mechanism to determine whether there is any activity during each backoffslot. If the medium is determined to be idle during a particular backoff slot, then the backoff procedure will decrement its backoff time by a slot time ($BT_{new} = BT_{old} - aSlotTime$). If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended. After the medium is determined to be idle for DIFS period, the backoff procedure is resumed. Transmission will begin whenever the backoff timer reaches zero. After a source station transmits a packet to a destination station, if the source station receives an acknowledgment (ACK) without errors after a short inter-frame space (SIFS) idle period, the transmission is concluded to be successfully completed. If the transmission is successfully completed, the contention window (CW) for the source station will be reset to the initial (minimum) value minCW. If the transmission is not successfully completed (i.e., the source station does not receive the ACK after SIFS), the contention window (CW) size will be increased (in the IEEE 802.11 DSSS $CW = 2^{(n+5)}-1$, $retry\ counter\ n = 0, ..., 5$), beginning with the initial value $minCW$, up to the maximum value $maxCW$ (in the IEEE 802.11 DSSS, $minCW = 31$ and $maxCW = 1023$). This process is called the *binary exponential backoff (BEB)*, which intends to resolve collisions. More detailed operations can be found in [10].

III. Fast Collision Resolution : The Basic Idea

There are two major factors affecting the throughput performance in the IEEE 802.11 MAC protocol: transmission failures (due to packet collisions) and the idle slots due to backoff at each contention cycle, which are shown in Fig. 1.

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 Fig. 1, we can examine one transmission cycle)[2], [4]:

$$\rho = \frac{\bar{m}}{E[N_c](E[B_c] \cdot t_s + \bar{m} + DIFS) + (E[B_c] \cdot t_s + \bar{m} + SIFS + ACK + DIFS)} \quad (2)$$

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 $\bar{m}$ is the average packet length.

From this result, we can see that the best scenario in Fig. 1, which gives the maximum throughput, 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

$$\rho_{best} = \frac{\bar{m}}{(\bar{m} + SIFS + ACK + DIFS)} \quad (3)$$

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

$$p_{trans}(i) = \begin{cases} 1 & \text{if station } i \text{ transmits its packet at current contention period} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Suppose that under some contention-based 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:

$$p_{trans}(i) = \frac{1}{(B_i + 1)} \quad (5)$$

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 and station i will transmit a packet immediately. Therefore, this can be interpreted as 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 as that station i has the probability of packet transmission of 0 at current contention period. From this discussion, (4) can be converted to (6):

$$B_i = \begin{cases} 0 & \text{if station } i \text{ transmits its packet at current contention period} \\ \infty & \text{otherwise} \end{cases} \quad (6)$$

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 assigns 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. One way to do so is to design an MAC protocol to approximate the behavior of perfect scheduling.

From (4) and (6), we conclude that to achieve high throughput, the MAC protocol should have the following operational characteristics:

- *Small random backoff timer for the station which has successfully transmitted a packet at current contention cycle*: This will decrease the average number of idle slots for each contention period, $E[B_c]$ in (2).
- *Large random backoff timer for stations that are deferring their packet transmissions at current contention period*: The deferring station means a station which has non-zero backoff timers. Large random backoff timers for deferring stations will decrease the collision probability at subsequent contention periods (and avoid future collisions more effectively).
- *Fast change of random backoff timer according to its current state: transmitting or deferring*: When a station transmits a packet successfully, its random backoff timer should be set small. The net effect of this operation is that whenever a station seizes the channel, it will use the medium as long as possible to increase the useful transmissions. When the station is deferring, its random backoff timers should be as large as possible to avoid the future collisions. The net effect is that all deferring stations will give the successful station more time to finish the backlogged packets. When a deferring station detects the medium is idle for a fixed number of slots, it would conclude that no other stations are transmitting, and hence it will reduce the backoff timers exponentially to reduce the average idle slots.

A. Fast Collision Resolution (FCR) Algorithm

As we pointed out, the major deficiency of the IEEE 802.11 MAC protocol comes from the slow collision resolution as the number of active stations increases. An active station can be in two modes at each contention period, namely, the transmitting mode when it wins a contention and the deferring mode when it loses a contention. When a station transmits a packet, the outcome is either one of the two cases: a successful packet transmission or a collision. Therefore, a station will be in one of the following three states at each contention period: a successful packet transmission state, a collision state, and a deferring state. In most distributed contention-based MAC algorithms, there is no change in the contention window size for the deferring stations, and the backoffti mer will decrease by one slot whenever an idle slot is detected. In the proposed fast collision resolution (FCR) 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 proposed algorithm preserves the simplicity for implementation like the IEEE 802.11 MAC.

The detailed FCR 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 idle for a slot, then it will decrement its backoff time (BT) by a slot time, i.e., $BT_{new} = BT_{old} - aSlotTime$ (or the backoff timer is decreased by one unit in terms of slot). When its backoff timer reaches to zero, the station will transmit a packet. If there are $[(minCW + 1) \times 2 - 1]$ consecutive idle slots being detected, its backoff timer should be decreased much faster (say, exponentially fast), i.e., $BT_{new} = BT_{old} - BT_{old}/2 = BT_{old}/2$ ($if\ BT_{new} < aSlotTime,\ then\ BT_{new} = 0$)or the backoff timer is decreased by a half. For example, if a station has the backoff timer 2047, hence its backoff time is $BT = 2047 \times aSlotTime$,wh ich will be decreased by a slot time at each idle slot until the backoff timer reaches 2040 (we assume that $[(minCW + 1) \times 2 - 1] = 7$ or $minCW = 3$). After then, if the idle slots continue, the backoff timer will be decreased by one half, i.e., $BT_{new} = BT_{old}/2$ at each additional idle slot until either it reaches to zero or it senses a non-idle slot, whichever comes first. As an illustration, after 7 idle slots, we will have $BT = 1020 \times aSlotTime$ on the $8th$ idle slot, $BT = 510 \times aSlotTime$ on the $9th$ idle slot, $BT = 255 \times aSlotTime$ on the $10th$ idle slot, and so on until it either reaches to zero or detects a non-idle slot. Therefore, the wasted idle backoffti me is guaranteed to be less than or equal to $18 \times aSlotTime$ for above scenario. The net effect is that the unnecessary wasted idle backoff time will be reduced when a station, which has just performed a successful packet transmission, runs out of packets for transmission or reaches its maximum successive packet transmission limit.

2. *Transmission Failure (Packet Collision)*: If a station notices that its packet transmission has failed possibly due to packet collision (i.e., it fails to receive an acknowledgment 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., $CW = minCW$, $BT = uniform(0, CW - 1) \times aSlotTime$. If a station has performed successive packet transmissions which reaches the maximum successive transmission limit (or larger), then its

TABLE I
NETWORK CONFIGURATIONS

Parameter	Value
SIFS	10 μsec
DIFS	50 μsec
A slot time	20 μsec
aPreambleLength	144 bits
aPLCPHeaderLength	48 bits
Bit rate	2 Mbps

contention window size will be increased to the maximum contention window size $maxCW$ and a random backoff time (BT) value will be chosen as follows: $CW = maxCW$, $BT = uniform(0, CW-1) \times aSlotTime$.

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 FCR 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.

IV. PERFORMANCE EVALUATION

In this section, we present the simulation studies for the proposed fast collision resolution (FCR) algorithm and the IEEE 802.11 MAC protocol in a wireless LAN using direct sequence spread spectrum (DSSS). The parameters used in the simulations are shown in Table I, which are based on the IEEE 802.11 network configurations[10].

We assume that the best-effort data packets are always available at all stations. In the simulations, the packet lengths for the best-effort data packets are geometrically distributed with parameter q[4]:

$$P[PacketLength = i\ slots] = q^{i-1}(1-q), \quad i \geq 1.$$

Thus, the average transmission time for a packet (the average packet length) is given by:

$$\bar{m} = t_s/(1-q) \quad (\mu sec)$$

where t_s is the slot time, i.e., $t_s = aSlotTime$.

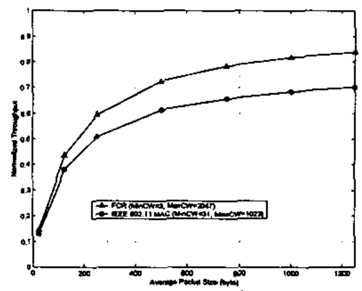

Fig. 2. Throughput for 10 BE data stations wireless LAN

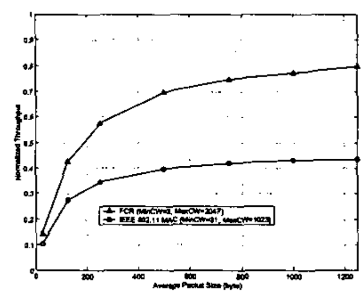

Fig. 3. Throughput for 100 BE data stations wireless LAN

We assigned the maximum successive packet transmission limit of the FCR algorithm as 10. All simulations are performed for 100 second simulation time.

Fig. 2 and 3 show the throughput results of the IEEE 802.11 MAC and FCR for 10 and 100 contending stations, where the average transmission time for a packet (i.e., the average packet length) changes from 100 μsec (25 bytes) to 5000 μsec (1250 bytes). The IEEE 802.11 MAC algorithm shows very poor throughput performance as the number of stations increases. In Fig. 2 and 3, we can see that the FCR algorithm significantly improve the throughput performance over the IEEE 802.11 MAC algorithm. Moreover, the throughput performance of the FCR algorithm are not severely degraded as the number of stations increases because of the highly efficient collision resolution strategy.

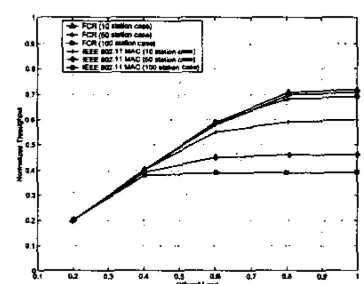

Fig. 4. Throughput vs. offered load

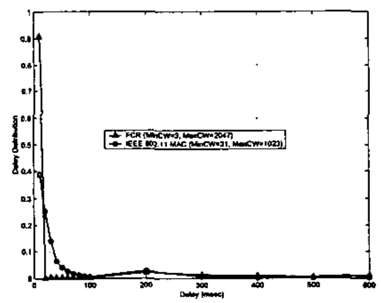

Fig. 5. Delay distribution for 10 stations wireless LAN

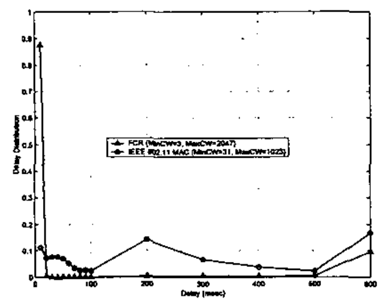

Fig. 6. Delay distribution for 100 stations wireless LAN

Fig. 4 shows the throughput vs. offered load for the IEEE 802.11 MAC and the FCR algorithm for 10, 50, 100 stations wireless LAN with the average transmission time for a packet (i.e., the average packet length) of 2000 μsec (500 bytes). We use a traffic generator with Poisson distribution to provide each offered load in this simulation. From Fig. 4, we can see that the FCR algorithm also performs very efficiently under light load conditions while providing high throughput as network load increases, and the number of stations hardly affects the performance of the FCR algorithm.

We carry out analysis for the packet delay of the IEEE 802.11 MAC and the FCR algorithm with the average transmission time for a packet (i.e., the average packet length) of 2000 μsec (500 bytes). The packet delay means the time period from the time when a packet arrives from higher layer to the MAC layer to the time it is successfully transmitted to the intended receiving station. Fig. 5 and 6 show the packet delay distributions for the IEEE 802.11 MAC and the FCR algorithm for 10 and 100 stations wireless LANs. We have not apply limitation on the number of retries in this simulation for simplicity. In Fig. 5, the FCR algorithm transmits 92% of all packets successfully within 10 msec while the remaining 8% packets spread over 10 msec to over 600 msec in delay. However, the IEEE 802.11 MAC transmits 39% packets within 10 msec, 25% packets in the range from 10 msec to 20 msec, 13% packets in the range from 20msec to 30 msec, and so on. In Fig. 6, the FCR algorithm transmits 89% of all packets successfully within 10 msec, while the IEEE 802.11 MAC transmits only 11% packets within 10 msec, 8% packets in the range from 10 msec to 20 msec, 8.5% packets in the range from 20 msec to 30 msec, and so on. In the simulation results for the packet delay, it is clear that the FCR algorithm transmits most packets successfully within pretty short time, while the IEEE 802.11 MAC transmits packets in much longer time due to collisions, which indeed shows that the FCR algorithm does resolve collision much faster than the IEEE 802.11 MAC algorithm does.

V. Conclusions

In this paper, we propose a new contention-based medium access control algorithms, namely, the fast collision resolution (FCR) algorithm. The FCR algorithm can achieve high throughput performance while preserving the implementation simplicity in wireless local area networks. In the FCR algorithm, each station changes the contention window size upon both successful packet transmissions and collisions (i.e., upon detecting a start of busy period) for all active stations in order to redistribute the backoff timers to actively avoid potential future collisions. Due to this operation, each station can quickly resolve collisions. Extensive simulation studies for throughput, delay distribution and TCP performance have demonstrated that the FCR algorithm gives significant performance improvement over the IEEE802.11 MAC algorithm.

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 IEEE802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, Vol.18, No.3, PP.535-547 Mar.2 000.

[4] F. Cali, M. Conti, and E. Gregori, 'Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. on Networking*, vol. 8, NO. 6, pp.785-799, Dec. 2000.

[5] A. Chandra, V. Gummalla, and J. O. Limb, 'Wireless medium access control protocols," *IEEE Communications Surveys*, Second Quarter 2000.

[6] B. P. Crow,I. Widjaja, J. G. Kim, and P. T. Sakai, 'IEEE 802.11 wireless local area networks," *IEEE Communications Magazine* Vol.35, pp.116-126, Sep. 1997.

[7] HIPERLAN Type 2 Standard, *ETSI*, 2000.

[8] C. Fullmer and J. Garcia-Luna-Aceves, 'Floor acquition multiple access (FAMA) for packet-ratio networks," *Proc. SIGCOMM'95*, pp.262-273, Cambridge, MA.

[9] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, 'Packet reservation multiple access for local wireless communications," *IEEE Transactions on Communications*, vol.37, no.8, pp.885-890, Aug. 1989.

[10] IEEE 802.11 Wireless LAN medium access control (MAC) and physical layer (PHY) specifi cations, *IEEE*, 1997.

[11] Y. Kwok and V. K. N. Lau, 'A quantitative comparison of multiple access control protocols for wireless ATM,"*IEEE Trans. on Vehicular Technology*, Vol.50, NO.3, pp.796-815, May,2 001.

[12] J. L. Sobrinho and A. S. Krishnakumar,"Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp.1353-1368, Aug. 1999.