Design of MAC Protocols With Fast Collision Resolution for Wireless Local Area Networks

Younggoo Kwon, Yuguang Fang, *Senior Member, IEEE*, and Haniph Latchman

***Abstract*—Development of efficient medium access control (MAC) protocols providing both high throughput performance for data traffic and good quality of service (QoS) support for real-time traffic is the current major focus in distributed contention-based MAC protocol research. In this paper, we propose an efficient contention resolution algorithm for wireless local area networks, namely, the fast collision resolution (FCR) algorithm. The MAC protocol with this new algorithm attempts to provide significantly higher throughput performance for data services than the IEEE 802.11 MAC algorithm and more advanced dynamic tuning backoff (DTB) algorithm. We demonstrate that this algorithm indeed resolves collisions faster and reduces the idle slots more effectively. To provide good fairness performance and to support good QoS for real-time traffic, we incorporate the self-clocked fair queueing algorithm and a priority scheme into the FCR algorithm and come up with the real-time FCR (RT-FCR) algorithm, and show that RT-FCR can simultaneously achieve high throughput and good fairness performance for nonreal-time traffic while maintaining satisfactory QoS support for real-time traffic.**

***Index Terms*—Backoff, IEEE 802.11, medium access control (MAC), quality of service (QoS), wireless local area networks (WLANs).**

I. Introduction

A GOOD medium access control (MAC) protocol for wireless local area networks (WLANs) should provide an efficient mechanism to share limited spectrum resources, together with simplicity of operations and high performance. The ideal performance would be low delay under low network load while high throughput under high network load, although in reality it is usually difficult to achieve both. Therefore, various MAC protocols have been developed to suit the various applications, where various tradeoff factors have been considered.

MAC algorithms in WLANs can be classified into two broad categories, namely, contention-based MAC algorithms and reservation-based MAC algorithms. It is challenging to address throughput, fairness and QoS issues in the distributed contention-based WLANs where no centralized scheduler exists. In this paper, we focus on the performance issues of the distributed contention-based MAC protocols.

Manuscript received October 5, 2001; revised May 2, 2002, December 8, 2002; accepted March 18, 2003. The editor coordinating the review of this paper and approving it for publication is L.-C. Wang. This work was supported in part by the U.S. National Science Foundation under Grant ANI-0093241 (NSF CAREER Award) and Grant ANI-0220287, and by the U.S. Office of Naval Research under Grant N000140210464 (ONR Young Investigator Award) and Grant N000140210554.

Y. Kwon is with the Department of Computer Engineering, Sejong University, Seoul 143-747, Korea (e-mail: ygkwon@sejong.ac.kr).

Y. Fang and H. Latchman are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6130 USA (e-mail: fang@ece.ufl.edu; latchman@list.ufl.edu).

Digital Object Identifier 10.1109/TWC.2004.827731

Distributed contention-based MAC protocol research in wireless networks started with ALOHA and slotted ALOHA in the 1970s. Later, multiple access collision avoidance (MACA), MACA wireless (MACAW), floor acquisition multiple access (FAMA), and distributed foundation wireless MAC (DFWMAC) were proposed for WLANs by incorporating the carrier sense multiple access (CSMA) technique with collision avoidance (CA) provisioning ([2], [9], [12], and references therein). The most popular contention-based wireless MAC protocol, the carrier sense CSMA/CA, becomes the basis for the MAC protocol for the IEEE 802.11 standard [17]. However, it is observed that when the number of active users increases, the throughput performance of the IEEE 802.11 MAC protocol degrades significantly because of the excessively high collision rate. Many researchers have focused on analyzing and improving the performance of the IEEE 802.11 MAC (see, for example, [3]–[5], and references therein).

To increase the throughput performance of a distributed contention-based MAC protocol, an efficient collision resolution algorithm is necessary 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 [2], [3], [11], [12]. Along these lines, Cali *et al.* [5] proposed an interesting algorithm to improve the performance of the IEEE 802.11 MAC protocol. Their basic idea is to dynamically assign the optimal contention window size at each station based on the estimation of the number of active stations. However, in real WLANs, it is not an easy task to accurately estimate the number of active stations at run time.

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 WLANs. 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

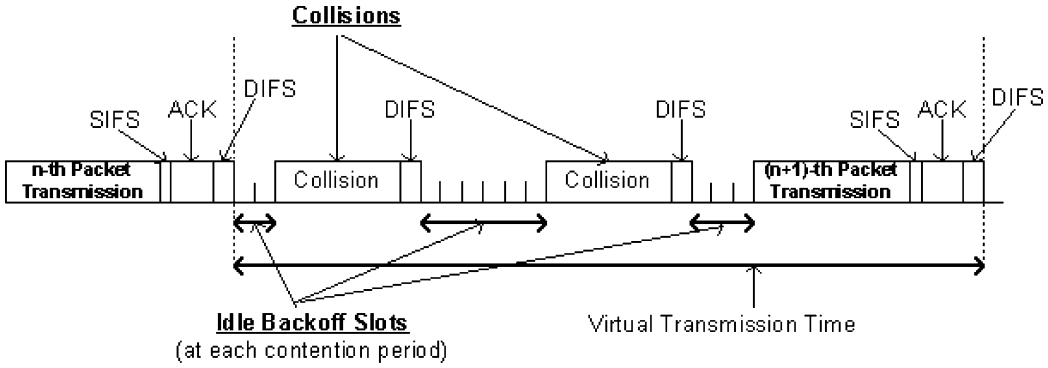

Fig. 1. Basic packet transmission structure of CSMA/CA.

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, more importantly, the deferring stations in the contention procedure, i.e., we devise an algorithm to redistribute the backoff timers in a larger contention window range for all active stations to avoid possible "future" collisions. To reduce the number of idle slots, the FCR algorithm gives a small idle backoff timer for the station with a 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 WLANs.

To address QoS, we then present a modified FCR algorithm, namely, the real time fast collision resolution (RT-FCR) algorithm, which improves the fairness and supports QoS for real-time applications. In this algorithm, we modify the distributed self-clocked fair queueing (SCFQ) [13], [29] algorithm, then combine a priority scheme based on the service differentiations [1], [10] to improve the FCR algorithm. The RT-FCR can achieve high throughput for the best-effort data traffic while at the same time provide high degree of fairness, and support QoS for real-time applications.

The rest of this paper is organized as follows. In the next section, we describe some well-known distributed contention-based MAC protocols to facilitate the comparisons with the proposed algorithms. Then, in Section III, we present the newly proposed FCR algorithm and RT-FCR algorithm. The performance analysis is carried out in Section IV. In the final section, we present the conclusions.

II. Distributed Contention-Based MAC Protocols for Wireless LANs

The IEEE 802.11 MAC [17] is the representative distributed contention-based MAC protocol widely used in current WLANs. The recently proposed dynamic tuning backoff (DTB) [5] algorithm improves the throughput performance of the IEEE 802.11 MAC by dynamically assigning the optimal contention window size to each station based on the run-time estimation of the number of active stations. In the following, we describe the basic operational procedures for these MAC algorithms to facilitate our comparative study in Section IV.

A. IEEE 802.11 MAC

As we mentioned before, the most popular contention-based MAC protocol is the CSMA/CA, which is widely used in the IEEE 802.11 LAN's. The basic operations 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 and with an acknowledgment (ACK) from the destination station. General operations of the IEEE 802.11 MAC protocol are as follows (since the RTS-CTS mechanism is optional [5], [17], we only consider the distributed coordination function (DCF) without the RTS-CTS handshaking 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)

$$\text{Backoff Time (BT)} = \text{B} \times \text{aSlotTime} \tag{1}$$

where B is the backoff timer which is a randomly chosen integer from a uniform distribution over the interval between zero and the current contention window size CW $(\text{B} = \text{uniform}[0, \text{CW}])$, and aSlotTime is the length of a unit time slot.

After a DIFS idle time, the station performs the backoff procedure with the carrier sensing mechanism by determining 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 will decrement its backoff time by a slot time $(BT_{\text{new}} = BT_{\text{old}} - aSlotTime)$. If the medium is determined to be busy at any time during a backoff slot with a nonzero backoff timer, then the backoff procedure is suspended. That is, if a station is deferring its packet transmission, then it will freeze the value of the backoff timer and the contention window size until next contention period. 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 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 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 CW size will be increased (e.g., $CW = 2^{(n+5)} - 1$, retry counter $n = 0, \ldots, 5$), beginning with the initial value $minCW$, up to the maximum value $maxCW$ (e.g., $minCW = 31$ and $maxCW = 1023$). This process is called the binary exponential backoff (BEB), which intends to resolve collisions. More detailed operations can be found in [17].

B. Dynamic Tuning Backoff (DTB)

Cali *et al.* [5] derive the average size of the contention window that maximizes the aggregate throughput under the assumption that all stations have the same average contention window size of transmitting a packet in steady state. They assume that in steady state, a station transmits a packet with the probability of $p = 1/(E[B] + 1)$, where $E[B]$ is the average value of the backoff timer. Since the average value of the backoff timer can be expressed as $E[B] = (E[CW] - 1)/2$, where $E[CW]$ is the average contention window size of sending a packet, the probability for a packet transmission is obtained by using the average contention window size as $p = 2/(E[CW] + 1)$. Based on this observation, Cali *et al.* are able to derive the following formula for the aggregate network throughput ρ (refer to [5] for detailed derivation procedures): [see (2) at bottom of page] where $\bar{m}$ is the average packet length, M is the number of active stations, τ is the maximum propagation time, q is the parameter for the geometric distribution of packet length, t_s is the length of a slot (i.e., aSlotTime), $E[coll]$ is the average collision length, and $E[S](= \bar{m} + 2\tau + SIFS + ACK + DIFS)$ is the average time to complete a successful packet transmission without any collisions.

Now, the aggregate network throughput ρ is derived as a function of the probability of a packet transmission p and the number of active stations M from (2), because all other parameters (τ, t_s, $\bar{m}$, q) are determined by the simulation configuration. This means that if the number of active stations M is fixed and given, then we can obtain the optimal p value, which maximizes the network throughput. This maximum throughput is the theoretical throughput limit or analytical upper bound based on the analysis approach from [5].

In the DTB algorithm, the throughput of the IEEE 802.11 MAC protocol, with an optimal backoff window size tuned to the optimal p value for each M, can be improved significantly. However, the p value and, hence, the optimal contention window size, depends on the number of active stations. The DTB method needs to compute the optimal contention window size at run-time based on the estimate of the number of active stations. If the estimation is not accurate, the wasted slots and packet collisions will be significant. However, to accurately estimate the number of active stations at run-time is not an easy task for practical WLANs with a distributed contention-based MAC protocol. In the next section, we will present a new MAC algorithm that achieves better performance.

III. FCR Algorithm

A. Basic Idea

There are two major factors affecting the throughput performance in the IEEE 802.11 MAC protocol: transmission failures (we only consider failures due to packet collisions) and the idle slots due to the 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) [3], [5]: [see (3) at bottom of page) 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 successful packet transmission without any overheads, in which case, $E[N_c] = 0$, $E[B_c] = 0$, the throughput would be

$$\rho_{\text{best}} = \frac{\bar{m}}{(\bar{m} + SIFS + ACK + DIFS)}. \tag{4}$$

$$\rho = \frac{\bar{m}}{\frac{(1-p)}{Mp}t_s + \frac{1-(1-p)^M - Mp(1-p)^{M-1}}{Mp(1-p)^{M-1}}[E[coll] + \tau + DIFS] + E[S]}$$
$$E[coll] = \frac{t_s}{1 - [(1-p)^M + Mp(1-p)^{M-1}]} \cdot \left[\sum_{h=1}^{\infty} \left\{h \cdot \left[(1 - pq^h)^M - (1 - pq^{h-1})^M\right]\right\} - \frac{Mp(1-p)^{M-1}}{(1-q)}\right] \tag{2}$$

$$\rho = \frac{\bar{m}}{E[N_c]\,(E[B_c] \cdot t_s + \bar{m} + DIFS) + (E[B_c] \cdot t_s + \bar{m} + SIFS + ACK + DIFS)} \tag{3}$$

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

$$p_{\text{trans}}(i) = \begin{cases} 1, & \text{if station } i \text{ transmits its packet at current contention period} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Suppose that under 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 [5]

$$p_{\text{trans}}(i) = \frac{1}{(B_i + 1)} \quad (6)$$

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, (5) can be converted to (7):

$$B_i = \begin{cases} 0, & \text{if station } i \text{ transmits its packet at current contention period} \\ \infty, & \text{otherwise} \end{cases} . \quad (7)$$

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 to ∞ for each contention cycle, 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. One way to do so is to design a MAC protocol to approximate the behavior of perfect scheduling.

From (5) and (7), we conclude that to achieve high throughput, the MAC protocol should have the following operational characteristics.

1) *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 (3).
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 been suspended its packet transmission with a nonzero backoff timer. Large random backoff timers for deferring stations will decrease the collision probability significantly (and avoid future collisions more effectively).
3) *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 for a certain period of time to increase the useful transmissions. When the station transmission is deferred, its random backoff timer should be set large to avoid the future collisions. The net effect is that all deferring stations will give the successful station more time to finish the back-logged packet transmissions. When a station detects the medium is idle for a fixed number of slots during backoff procedure, it would conclude that no other stations are transmitting, and hence it will reduce the backoff timer exponentially fast to reduce the average idle slots.

B. 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 backoff timer will decrease by one slot whenever an idle slot is detected. In the proposed 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 of implementation like the IEEE 802.11 MAC.

The FCR algorithm has the following characteristics:

1) use much smaller initial (minimum) contention window size $minCW$ than the IEEE 802.11 MAC;
2) use much larger maximum contention window size $maxCW$ than the IEEE 802.11 MAC;
3) increase the contention window size of a station when it is in both collision state and deferring state;
4) reduce the backoff timers exponentially fast when a prefixed number of consecutive idle slots are detected.
5) assign the maximum successive packet transmission limit (T_{PkTrans}) to achieve good fairness performance.

Items 1 and 4 attempt to reduce the average number of idle backoff slots for each contention period $(E[B_c])$ in (3). Items 2 and 3 are used to quickly increase the backoff timers, hence, quickly decrease the probability of collisions. In item 3, the FCR algorithm has the major difference from other contention-based MAC protocols such as the IEEE 802.11 MAC. In the IEEE 802.11 MAC, the contention window size of a station is increased only when it experiences a transmission failure (i.e., a collision). In the FCR algorithm, the contention window size of a station will increase not only when it experiences a collision but also when it is in the deferring mode and senses the start of a new busy period. Therefore, all stations having packets to transmit (including those which are deferred) will change their contention window sizes at each contention period in the FCR

algorithm. Item 5 is used to avoid the situation that a station dominates the packet transmissions for a long period of time. If a station has performed successive successful packet transmissions for the maximum successive packet transmission limit (T_{PkTrans}), it changes its contention window size to the maximum value (maxCW) in order to give opportunities for medium access to other stations (the maximum successive packet transmission limit of station i, $T_{\mathrm{PkTrans,i}}$ will be dynamically assigned by using the distributed SCFQ algorithm to improve fairness in the RT-FCR algorithm discussed later, while we use a constant value of the maximum successive packet transmission limit (T_{PkTrans}) in the FCR algorithm).

The detailed FCR algorithm is described as follows according to the state of the station:

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., $BT_{\mathrm{new}} = BT_{\mathrm{old}} - aSlotTime$ (or the backoff timer is decreased by one unit in terms of slot). When its backoff timer reaches 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., $BT_{\mathrm{new}} = BT_{\mathrm{old}} - BT_{\mathrm{old}}/2 = BT_{\mathrm{old}}/2$ (if $BT_{\mathrm{new}} < aSlotTime$, then $BT_{\mathrm{new}} = 0$) or the backoff timer is decreased by a half. For example, if a station has the backoff timer of 2047, hence its backoff time is $BT = 2047 \times aSlotTime$, which 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., $BT_{\mathrm{new}} = BT_{\mathrm{old}}/2$ at each additional idle slot until either it reaches to zero or it senses a nonidle slot, whichever comes first. As an illustration, after seven idle slots, we will have $BT = 1020 \times aSlotTime$ on the eighth idle slot, $BT = 510 \times aSlotTime$ on the ninth idle slot, $BT = 255 \times aSlotTime$ on the tenth idle slot, and so on until it either reaches zero or detects a nonidle slot. Therefore, the wasted idle backoff time is guaranteed to be less than or equal to $18 \times aSlotTime$ for the described 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. We remark here that other backoff timer exponential decreasing algorithm can be developed to optimize the overall performance.
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., $CW = \min[maxCW, ((CW + 1) \times 2 - 1)]$, $BT = uniform(0, CW) \times aSlotTime$, where $uniform(a, b)$ indicates an integer 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., $CW = minCW$, $BT = uniform(0, CW) \times aSlotTime$. If a station has performed successive packet transmissions for the maximum successive transmission limit, then it will perform the following actions to give opportunities for the medium access to other stations: $CW = maxCW$, $BT = uniform(0, CW) \times aSlotTime$.
4) *Deferring State*: For a station in the 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: $CW = \min[maxCW, ((CW + 1) \times 2 - 1)]$, $BT = uniform(0, CW) \times aSlotTime$.

In the FCR algorithm, the station that has successfully transmitted a packet will have the minimum contention window size and a small value of the 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.

To elaborate the operations of the FCR algorithm, we use some examples to illustrate the major difference between the IEEE 802.11 MAC and the FCR algorithm. Table I shows an example of the IEEE 802.11 MAC operations with the contention window size $CW = 2^{(n+3)} - 1$, retry counter $n = 0, \ldots, 7$ (i.e., $\mathrm{minCW} = 7$ and $\mathrm{maxCW} = 1023$). In this example, there are 10 active stations contending for the use of the medium based on the IEEE 802.11 MAC. When the contention begins (i.e., the medium is determined to be idle for DIFS period by the carrier sensing mechanism), each station performs the backoff procedure with its random backoff time (BT) determined from the initial contention window range [0, 7] (hence $BT = uniform[0, 7] \times aSlotTime$). When a station detects the current slot idle, it will decrement its backoff time by a slot time $BT_{\mathrm{new}} = BT_{\mathrm{old}} - aSlotTime$ (i.e., the backoff timer is decreased by one unit). After one idle slot, the backoff timers of stations 0 and 8 reach to zero, thus, in the following slot, both station 0 and station 8 will transmit their packets at the same time and a collision will occur. The backoff procedures of all deferring stations are suspended and will resume after the medium is determined to be idle for DIFS period (i.e., next contention period). After stations 0 and 8 notice that their packet transmissions failed, their contention window sizes will be increased to 15 and their backoff timers will be chosen in the range of [0, 15] randomly. When a new DIFS period is detected, stations 2 and 4 transmit packets after one idle slot and a collision occurs. Stations 1 and 6 transmit packets and a collision occurs in the following contention period. After then, when the next DIFS period is detected, station 7 has a successful packet transmission. In the whole contention cycle (the time period starting with the end of a successful packet transmission and ending with the

TABLE I
EXAMPLE OF IEEE 802.11 BINARY EXPONENTIAL BACKOFF ALGORITHM

0	1	2	3	4	5	6	7	8	9	Station Number
1(7)	3(7)	2(7)	7(7)	2(7)	6(7)	3(7)	4(7)	1(7)	6(7)	Contention Begins
0(7) 8(15)	2(7)	1(7)	6(7)	1(7)	5(7)	2(7)	3(7)	**0(7)** 14(15)	5(7)	Collision on station 0 & 8
7(15)	1(7)	**0(7)** 4(15)	5(7)	**0(7)** 9(15)	4(7)	1(7)	2(7)	13(15)	4(7)	Collision on station 2 & 4
6(15)	**0(7)** 10(15)	3(15)	4(7)	8(15)	3(7)	**0(7)** 5(15)	1(7)	12(15)	3(7)	Collision on station 1 & 6
5(15)	9(15)	2(15)	3(7)	7(15)	2(7)	4(15)	**0(7)** **3(7)**	11(15)	2(7)	Successful Packet Transmission on station 7

TABLE II
EXAMPLE OF FAST COLLISION RESOLUTION BACKOFF ALGORITHM

0	1	2	3	4	5	6	7	8	9	Station Number
1(3) 1(7)	**0(3)** 3(7)	2(3) 2(7)	1(3) 7(7)	2(3) 2(7)	2(3) 6(7)	3(3) 3(7)	3(3) 4(7)	1(3) 1(7)	**0(3)** 6(7)	Collision on station 1 & 9
0(7) 8(15)	2(7) 10(15)	1(7) 2(15)	6(7) 1(15)	1(7) 12(15)	5(7) 4(15)	2(7) 15(15)	3(7) 6(15)	**0(7)** 14(15)	5(7) 3(15)	Collision on station 0 & 8
7(15) 22(31)	9(15) 18(31)	1(15) 28(31)	**0(15)** 1(3)	11(15) 5(31)	3(15) 17(31)	14(15) 11(31)	5(15) 9(31)	13(15) 14(31)	2(15) 23(31)	Success on station 3
21(31) 40(63)	17(31) 9(63)	27(31) 38(63)	**0(3)** 3(3)	4(31) 58(63)	16(31) 24(63)	10(31) 17(63)	8(31) 20(63)	13(31) 44(63)	22(31) 1(63)	Success on station 3
39(63) 100(127)	8(63) 55(127)	37(63) 29(127)	2(3) 5(7)	57(63) 111(127)	23(63) 46(127)	16(63) 81(127)	19(63) 30(127)	43(63) 9(127)	**0(63)** 1(3)	Success on station 9
99(127) 67(255)	54(127) 29(255)	28(127) 189(255)	4(7) 11(15)	110(127) 55(255)	45(127) 210(255)	80(127) 160(255)	29(127) 240(255)	8(127) 120(255)	**0(3)** 2(3)	Success on station 9

start of the next successful packet transmission), there have been three consecutive collisions before one successful packet transmission. We observe in Table I that most contention window sizes chosen for the backoffs are not big enough to avoid future packet collisions. Since the IEEE 802.11 MAC cannot provide the proper contention window size as the number of active stations increases, collisions are not resolved quickly, which leads to poor throughput performance.

Table II shows an example for the FCR algorithm with the contention window size $CW = 2^{(n+2)} - 1$, retry counter $n = 0, \ldots, 9$ (i.e., $\mathrm{minCW} = 3$ and $\mathrm{maxCW} = 2047$). In Table II, stations 1 and 9 collide in the first contention period. Stations 1 and 9 then increase their contention window sizes to 7 and pick up their backoff timers in the range of [0, 7] randomly. All deferring stations also increase their contention window sizes to 7 and pick up the new backoff timers in the range of [0, 7] randomly. In the second contention period, stations 0 and 8 collide and will repeat the same procedure. In the third contention period, station 3 transmits a packet successfully. We observe in Table II that most contention window sizes of the deferring stations are increased quickly (which makes large backoff timers), so the FCR algorithm resolves the contentions very effectively, which results in significantly lower collision probability during each contention period in the future.

In Table I and Table II, we can clearly see the major differences in the operations between the IEEE 802.11 MAC and the FCR algorithm. To put it briefly, the high throughput of the FCR algorithm comes from: the small backoff timer for the station that transmits a packet at current contention period (this reduces the wasted idle slots), the large backoff timers for the stations that are deferring for packet transmissions (this reduces the collision probability), and faster change of backoff timers according to the current state: transmitting or deferring. This means that the FCR algorithm satisfies the required conditions for high throughput performance shown in (7).

C. RT-FCR Algorithm

In the FCR algorithm, we focus on the throughput performance for the best-effort data services. However, intensive research has geared to address the QoS in the MAC layer. In this section, we attempt to extend the FCR algorithm to improve fairness and to support QoS for real-time applications in WLANs. We first modify the distributed self-clocked fair queueing (SCFQ) [13], [29] algorithm and the priority algorithm based on service differentiations [1], [10], and then incorporate them into the FCR algorithm to address the fairness for data traffic and QoS support for real-time traffic. We call this extended FCR algorithm as the RT-FCR algorithm.

1) Fair Scheduling: Distributed Self-Clocked Fair Queueing Algorithm: Fairness is an important issue in MAC protocol design for WLANs. The IEEE 802.11 MAC has the inherent unfairness characteristics [21], [27], [29]. FCR makes thing worse because the deferring nodes will tend to defer their transmissions further by expanding their contention windows upon detecting any start of busy periods before the backoff timers expire. However, with proper provisioning in the FCR algorithm, we can address the fairness issue while maintaining high throughput performance of FCR algorithm. The idea is to modify the SCFQ algorithm [13] and incorporate it into FCR algorithm. We combine these two algorithms and dynamically assign the successive transmission period of the FCR algorithm by using the modified SCFQ algorithm. We call this new algorithm the fairly scheduled FCR (FS-FCR) algorithm. The SCFQ algorithm has been used to address the fairness issue for IEEE 802.11 WLANs by Vaidya *et al.* [29]. While the approach proposed by Vaidya *et al.* is packet-by-packet based and controlling the backoff timers, our approach is based on multiple successive packet transmissions (i.e., dynamically control the maximum successive transmission period). The basic operations of the FS-FCR algorithm are described in the following.

1) Each arriving packet to the queue in the MAC layer of a station is tagged with a service tag before it is placed in the queue.
2) When a k-th packet of station i, P_i^k, arrives at the queue of the station, a service tag F_i^k is assigned as follows (detailed explanation can be found in [13])

$$F_i^k = \max\left\{v\left(a_i^k\right), F_i^{k-1}\right\} + \frac{L_i^k}{\phi_i}$$

where $v(a_i^k)$ is the virtual time at the time instance of a_i^k, where a_i^k is the real time when packet P_i^k arrives, L_i^k is the size of packet P_i^k, and ϕ_i is the weight of the flow i.
3) The virtual time $v(t)$ is updated whenever there is a successful packet transmission. The virtual time is set to the service tag of that packet just successfully transmitted. The virtual time $v(t)$ approximately represents the normalized fair amount of packet transmissions that each station should have performed. We say it "approximately" because a packet with the smallest service tag shall not be guaranteed to be served first in distributed contention-based MAC protocols. Once a busy period is over, i.e., when all stations do not have any packets to transmit, the virtual time is reset to zero.
4) Whenever a new station i acquires the medium for a packet transmission, the maximum successive transmission limit (i.e., the successive transmission time period) of the station i, $T_{\mathrm{PkTrans,i}}$, is determined by the difference between the virtual time $v(t)$ and the service tag F_i^k at the front of the packet flow at station i. If the service tag of station i is much smaller than the current virtual time, then its maximum successive transmission limit is assigned a value large enough to reduce the discrepancy between the current virtual time and the service tag at the front of flow i. If the service tag of station i is close to or larger than the current virtual time, then its maximum successive transmission limit is assigned to the minimum or small value to avoid increasing the discrepancy between the current virtual time and the service tag at the front of the packet flow of station i.

An example for assigning the maximum successive transmission limit we study here is given as follows:

$$T_{\mathrm{PkTrans,i}} = g\left[v(t) - F_i^k\right]$$

where

$$g[x] = \begin{cases} 1, & x \le (-10 \times \bar{m}) \\ 3, & (-10 \times \bar{m}) < x \le (-5 \times \bar{m}) \\ 7, & (-5 \times \bar{m}) < x \le (5 \times \bar{m}) \\ 10, & (5 \times \bar{m}) < x \le (10 \times \bar{m}) \\ 20, & (10 \times \bar{m}) < x \le (20 \times \bar{m}) \\ 30, & (20 \times \bar{m}) < x \le (30 \times \bar{m}) \\ 40, & (30 \times \bar{m}) < x \le (40 \times \bar{m}) \\ 50, & x > (40 \times \bar{m}) \end{cases}$$

where $\bar{m}$ is the average packet length. For example, if the service tag of station i is leading the virtual time by ten times or more of the average packet length when it acquires the medium access (i.e., station i may transmit too many packets compared to other stations), it assigns its maximum successive transmission limit as the minimum value of 1 ($T_{\mathrm{PkTrans,i}} = g[x] = 1$). If the service tag of station i is lagging the virtual time by more than 40 times of the average packet length (i.e., station i may transmit too few packets compared to other stations), it assigns its maximum successive transmission limit as the maximum value of 50 ($T_{\mathrm{PkTrans,i}} = g[x] = 50$).

We notice that the RT-FCR algorithm attempts to combine advantages of the FCR algorithm and the SCFQ scheme. In this way, we can achieve high throughput and good fairness performance simultaneously.

2) QoS Support With Priority-Based MAC: In order to deal with the QoS requirements for real-time applications, many algorithms have been proposed in the contention-based MAC protocols for WLANs. The most popular approach is to use a priority scheme for each traffic type, i.e., the real-time traffic has higher priority for medium access than the best-effort data traffic. With higher priority for medium access, real-time traffic will be served earlier than the best-effort data traffic, which results in relative performance improvements for real-time traffic over data traffic.

In the RT-FCR algorithm, we give priorities by assigning different backoff ranges based on each of three main traffic types: voice, video, and best-effort data traffic. Intuitively, the smaller the backoff range is, the higher the priority. The basic medium access scheme with three different traffic types is shown in Fig. 2, and the backoff ranges for the medium access are assigned according to each traffic type shown in Table III.

In Fig. 2 and Table III, we can see that the proposed medium access algorithm effectively provides "soft" reservation to a station according to the traffic type. In this scheme, voice traffic has the highest priority (i.e., the smallest average backoff value), and video traffic has higher priority over best-effort data traffic because of different backoff ranges according to the traffic type. The access guaranteed initial backoff range [0, 7] is given to voice traffic, i.e., only voice packets can be transmitted on this backoff range and other packets (video or data) will be transmitted beyond this backoff range which is shown in the backoff ranges for video and data traffic in Table III (for these backoff ranges, the constant 8 is added to move the backoff ranges for video and data traffic beyond the initial backoff range of voice traffic). Video traffic uses a much smaller maximum contention window size than best-effort data traffic in order to give higher priority over best-effort data traffic for the medium access, i.e., video traffic will have a smaller average backoff value than data traffic shown in Fig. 2.

In addition to assigning different backoff ranges, the RT-FCR algorithm uses different contention algorithms according to traffic types. The basic procedures for the priority scheme of the RT-FCR algorithm are shown in Fig. 3 and explained in the following.

1) *Voice Packet*: The IEEE 802.11 MAC algorithm with the minimum contention window size of 7 and the maximum contention window size of 255 is used for a station with voice traffic. It has the access guaranteed initial backoff range [0, 7], which gives the highest priority to voice traffic for accessing the medium. Voice traffic needs repeated packet transmissions in constant time intervals (e.g., only one packet transmission is needed every 30

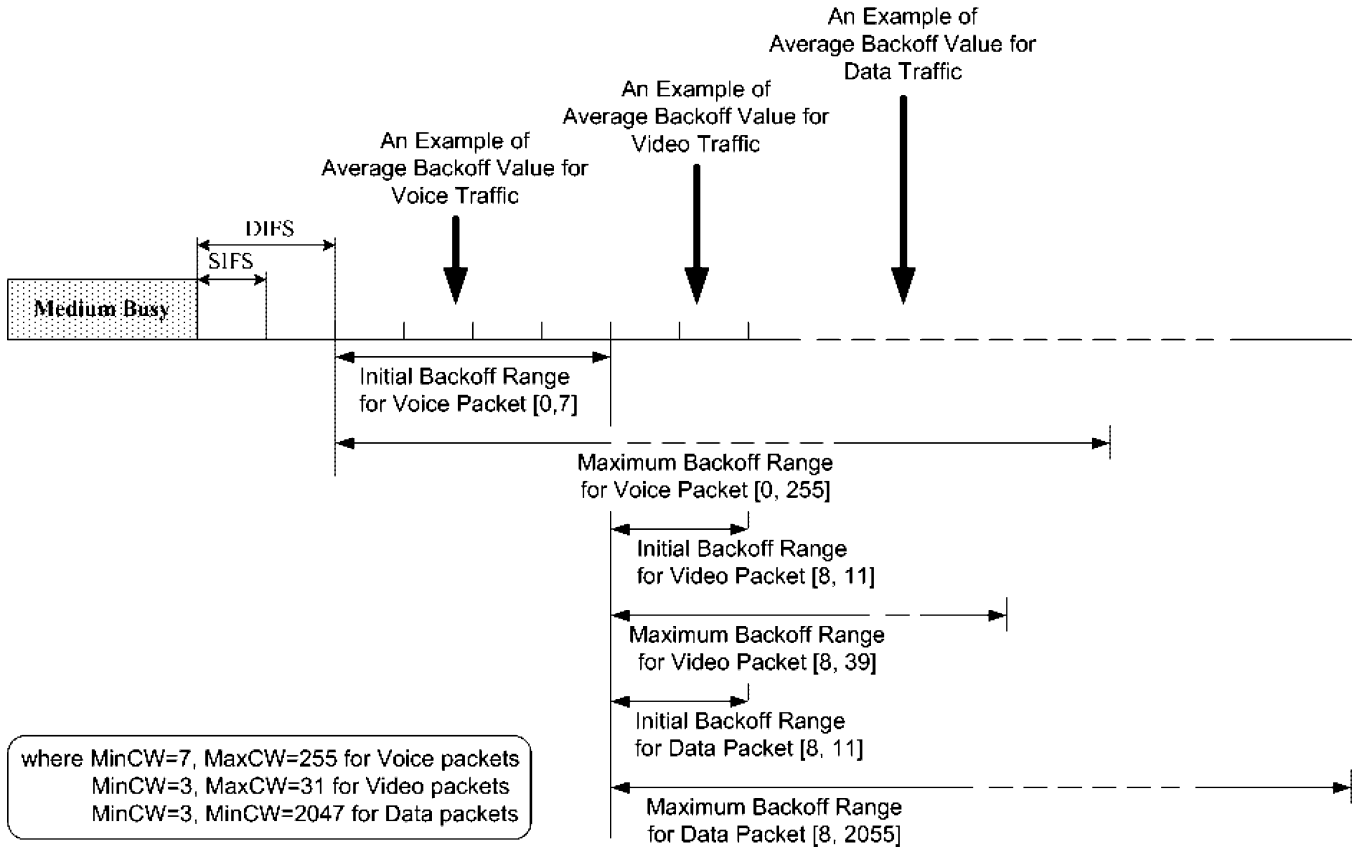

Fig. 2. RT-FCR medium access scheme.

TABLE III
ASSIGNING BACKOFF RANGE

Backoff range for voice traffic	{[0,7]:[0,15]:[0,31]:[0,63]:[0,127]:[0,255]}
Backoff range for video traffic	{[0,3]:[0,7]:[0,15]:[0,31]}+8
Backoff range for data traffic	{[0,3]:[0,7]:[0,15]:[0,31]:[0,63]:[0,127]:[0,255]:[0,511]:[0,1023]:[0,2047]}+8

ms). The FCR algorithm works well for best-effort data traffic with high load. However, for voice traffic, the traffic load is low, the IEEE 802.11 MAC is more suitable because it does not increase the contention window sizes of the deferring stations. This results in smaller wasting idle slots for voice traffic.

In Table IV, the voice packet dropping ratio is shown for the FCR and IEEE 802.11 MAC with 15 voice stations and ten best-effort data stations. From this simple example, we can see that the FCR algorithm does not support CBR traffic well compared to the IEEE 802.11 MAC, which is why we choose IEEE 802.11 MAC for CBR traffic.

2) *Video Packet*: FCR algorithm with the minimum contention window size of 3 and the maximum contention window size of 31 is used for video packet transmissions. It starts the contention for video packet transmissions after the initial access guaranteed backoff range for voice traffic. The smaller maximum contention window size of video traffic ($\mathrm{MaxCW} = 31$) than that of best-effort data traffic ($\mathrm{MaxCW} = 2047$) gives video traffic higher priority over best-effort data traffic.

3) *Best-Effort Data Packet*: FCR algorithm with the minimum contention window size of 3 and the maximum contention window size of 2047 is used for best-effort data traffic. It starts the contention for best-effort data packet transmissions after the initial access guaranteed backoff range for voice traffic. FCR scheme with the large maximum contention window size achieves the high throughput for best-effort data traffic in addition to giving the opportunity to voice or video traffic for access.

We point out that the QoS support here is in the statistical sense, which, in the authors' opinion, is the best we can guarantee in the MAC layer in the contention-based MAC protocol.

IV. PERFORMANCE EVALUATION

In the following, we will carry out the performance analysis for both FCR and RT-FCR.

A. Performance Evaluation of FCR

In this subsection, we present the performance analysis for the proposed FCR algorithm using simulations for the frequency

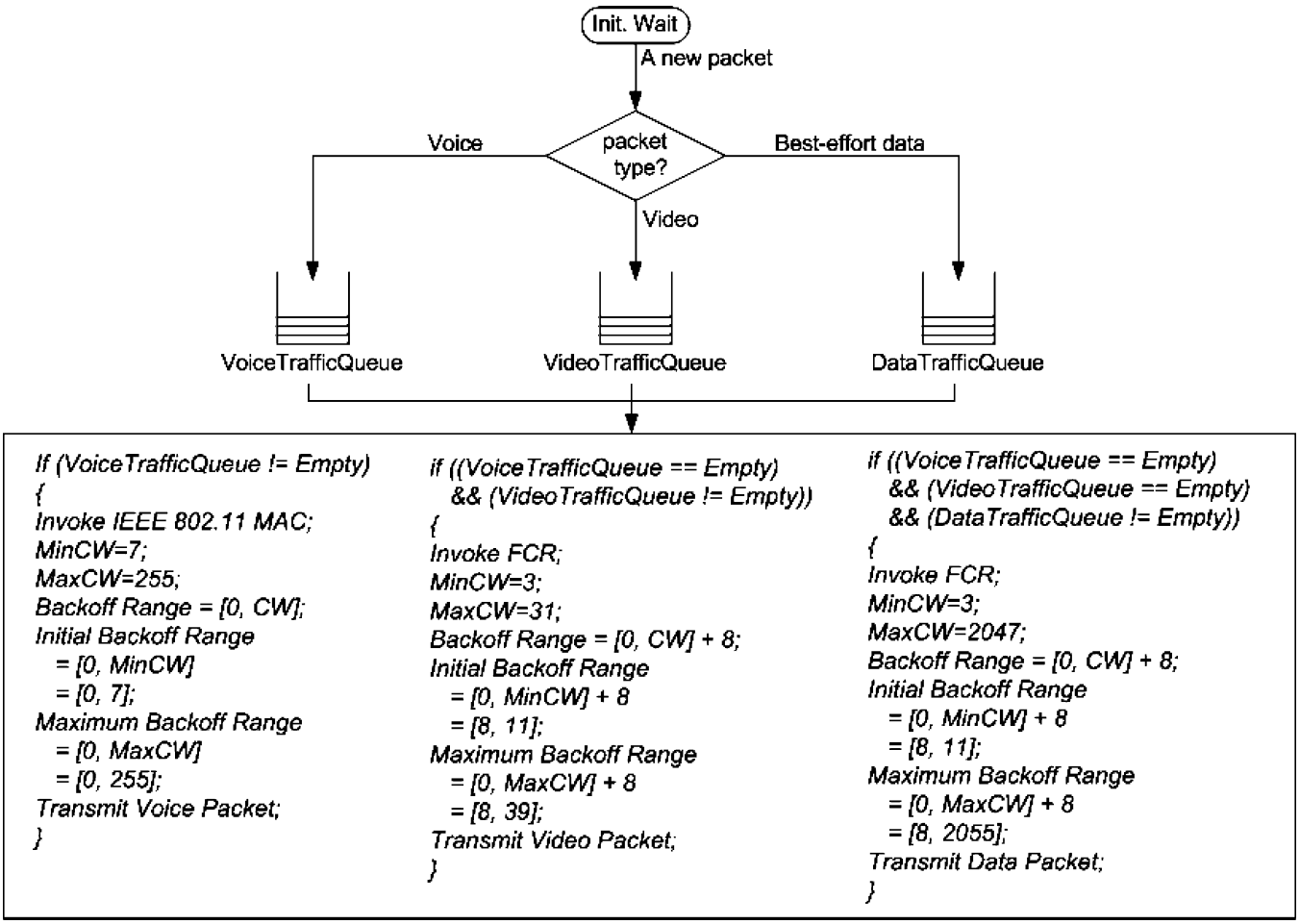

Fig. 3. Priority scheme of RT-FCR algorithm.

TABLE IV
VOICE PACKET DROPPING RATIO (%) FOR 15 VOICE STATIONS AND 10 BEST-EFFORT DATA STATIONS

(MinCW,MaxCW)	(3,255)	(3,511)	(3,1023)	(7,255)	(15,255)
FCR	9.8	11	11.2	15.6	31.3
IEEE 802.11	0	0	0	0	0

TABLE V
NETWORK CONFIGURATIONS

Parameter	Value
SIFS	28 μsec
DIFS	128 μsec
aSlotTime	50 μsec
Bit rate	2 Mbps
Propagation delay	1 μsec

hopping spread spectrum (FHSS) WLANs [17]. The parameters used in the simulations are shown in Table V, which is consistent with those used in [5]. 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 (we use the same simulation environment used in [5] for the DTB and the IEEE 802.11 MAC algorithm for the performance comparison):

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

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

$$\bar{m} = \frac{t_s}{(1-q)}\ (\mu s)$$

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

We assign the maximum successive packet transmission limit in the FCR algorithm as 10 (i.e., $T_{\mathrm{PkTrans}} = 10$) for illustrative purpose, and more careful choice of this parameter will be investigated in the future. All simulations are performed for 100-s simulation time.

In Table VI and VII, the throughput results of the FCR and IEEE 802.11 MAC algorithms with the average packet length of 40 slots are shown for various (MinCW, MaxCW) combinations. From Table VI, we can see that if we use large minimum contention window size (MinCW) of 7 and 15 in the FCR algorithm, the throughput is decreased due to the wasting idle slots. If we use small maximum contention window size (MaxCW) of 1023 and 511, then the throughput is decreased because of the high collision probability under large number of users. Too large value of the MaxCW such as 4095 also decreases the throughput of the FCR algorithm for small number of stations. The proper values for MinCW and MaxCW are critical for the throughput performance and optimization should be carried out. Based on our scenarios in our simulation study, we find that the choice of $\mathrm{MinCW} = 3$ and $\mathrm{MaxCW} = 2047$ makes a good

TABLE VI
THROUGHPUT RESULTS OF FCR ALGORITHM ON VARIOUS (MinCW, MaxCW) COMBINATIONS

(MinCW,MaxCW)	(3,511)	(3,1023)	(3,2047)	(3,4095)	(7,1023)	(7,2047)	(15,2047)
10 Data Stations	0.7833	0.7872	0.7852	0.7795	0.7569	0.7577	0.7033
100 Data Stations	0.6507	0.7221	0.7656	0.7792	0.7128	0.7454	0.6662

TABLE VII
THROUGHPUT RESULTS OF IEEE 802.11 MAC ALGORITHM ON TWO DIFFERENT (MinCW, MaxCW) COMBINATIONS

(MinCW,MaxCW)	(15,1023)	(31,255)
10 Data Stations	0.6075	0.6564
100 Data Stations	0.3775	0.3197

throughput performance, hence we will choose these basic parameters for our simulation study for the FCR algorithm hereafter. The throughput results for the IEEE 802.11 MAC algorithm with two different (MinCW, MaxCW) combinations are shown in Table VII. Current IEEE 802.11 FHSS standard provides the minimum contention window size and the maximum contention window size as (15, 1023) ([17]), while (31, 255) is used in ([5]). If we use (31, 255), the throughput is better for 10 data station case than when we use (15, 1023). For 100 data station case, the throughput when using (15, 1023) shows better result. In this paper, we use $\mathrm{MinCW} = 31$ and $\mathrm{MaxCW} = 255$ as the basic parameters for the simulations for the IEEE 802.11 MAC algorithm to preserve the same simulation environments in [5]. All other parameters for the simulations are the same as in [5].

Fig. 4(a)–(c) show the throughput results of the IEEE 802.11 MAC, DTB, and FCR algorithms for 10, 50, and 100 contending stations, respectively, where the average packet length changes from 10 slots $(q = 0.9)$ to 100 slots $(q = 0.99)$. The IEEE 802.11 MAC algorithm shows very poor throughput performance as the number of active stations increases. The main reason is that the probability of collisions becomes higher as the number of stations becomes larger. In the FCR algorithm, all stations, except the one with successful packet transmission, will increase their contention window sizes whenever the system has either a successful packet transmission or has a collision. This implies that all stations can quickly repick large contention window sizes to avoid future possible collisions, consequently, the probability of collisions will be decreased to small values. At the same time, a station with a successful packet transmission has the minimum contention window size of 3, which is much smaller than the minimum contention window size used in the IEEE 802.11 MAC algorithm $(\mathrm{minCW} = 31)$. This will reduce the wasted medium idle time to a much smaller value when compared to the IEEE 802.11 MAC and the Dynamic Tuning Backoff algorithm. In Fig. 4(a)–(c), we can see that the FCR algorithm significantly improve the throughput performance over the IEEE 802.11 MAC algorithm. The FCR algorithm shows higher throughput performance than the theoretical throughput limit (the analytical upper bound) of the DTB algorithm, and has much smaller wasting idle slots for each contention period than the DTB algorithm while both algorithms have similar values of the probability of collisions. 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(d) shows the throughput versus the offered load for the FCR algorithm for 10, 50, 100 stations WLANs with the average packet length of 40 slots. We use a traffic generator with Poisson distribution to provide each offered load in this simulation. From Fig. 4(d), we can see that the FCR algorithm also performs well under light load conditions and provides high throughput as network load increases, and the number of stations hardly affects the performance of the FCR algorithm due to the adaptive nature of the FCR algorithm.

We also carry out the analysis for the packet delay of the IEEE 802.11 MAC and the FCR algorithm with the average packet length of 40 slots. 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. Figs. 5(a) and 5(b) show the packet delay distributions for the IEEE 802.11 MAC and the FCR algorithm for 10 and 100 stations WLANs. We have not applied the limitation on the number of retries in this simulation for simplicity. In Fig. 5(a), the FCR algorithm transmits 91% of all packets successfully within 10 ms while the remaining 9% packets spread over 10 to over 600 ms in delay distribution. However, the IEEE 802.11 MAC transmits 39% packets within 10 ms, 25% packets in the range from 10 to 20 ms, 13% packets in the range from 20 to 30 ms, and so on. In Fig. 5(b), the FCR algorithm transmits 88% of all packets successfully within 10 ms, while the IEEE 802.11 MAC transmits only 11% packets within 10 ms, 8% packets in the range from 10 to 20 ms, 8.5% packets in the range from 20 to 30 ms, and so on. In the simulation results for the packet delay, it is clear that the FCR algorithm transmits most packets successfully within comparatively shorter 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 more effectively than the IEEE 802.11 MAC algorithm does.

B. Performance Evaluation of RT-FCR

In this subsection, we present the studies on the RT-FCR algorithm in the WLANs utilizing FHSS [17].

1) Source Models: We consider three different types of traffic: constant bit rate (CBR) voice traffic, variable bit rate (VBR) video traffic, and best-effort data traffic. Voice sources have two phase process with talkspurts and silent gaps. During a talkspurt, voice sources generate CBR traffic. H.263 video sources generate VBR traffic with 40-ms interframe period. We assume that the best-effort data sources always have packets to transmit. The detailed source models used in our simulations are described in the following.

1) *Voice Model* [6], [16]: A voice source has two states, talkspurts and silent gaps identified by a speech activity detector. The probability that a principal talkspurt, with mean duration t_1 second, ends in a time slot of duration τ s is $\gamma = 1 - exp(-\tau/t_1)$. The probability that a silent

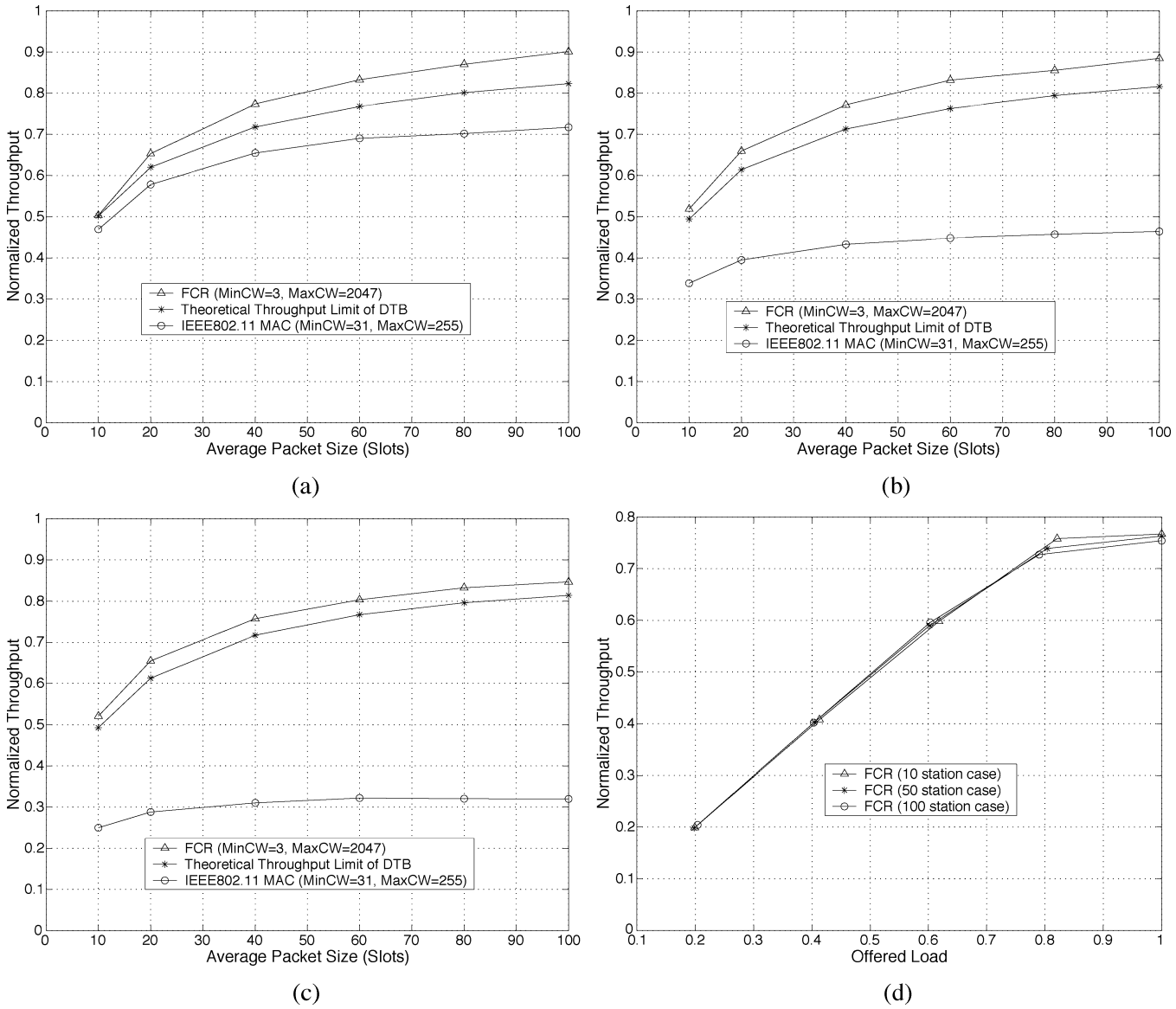

Fig. 4. Throughput performance for FCR algorithm. (a) The 10 BE data station case. (b) The 50 BE data station case. (c) The 100 BE data station case. (d) Throughput versus offered load.

gap, of mean duration t_2 s , ends during τ s time slot is $\sigma = 1 - exp(-\tau/t_2)$. Measured mean values for t_1 of principal talkspurts and t_2 of principal silent gaps are 1.00 and 1.35 seconds. We use 32-kbps voice traffic sources which generate one 120-B payload voice packet every 30 ms during talkspurts period, and we assign the deadline for voice packet delay as 30 ms (i.e., the maximum voice packet delay is 30 ms).

2) *Video Model* [6], [22]: We use the H.263 video traffic with 40-ms interframe period, i.e., 25 frames per second. During an interframe period, each video source generates a frame consisting of a variable number of packets. As soon as packets become available from the coder, they could be transmitted at the maximum rate the channel allows. The video packet size is 120 B and the mean rate of video traffic is 48 kbps and the maximum rate is 480 kbps. That is, there are two packets per frame for the mean rate and the maximum number of packets per frame is 20. We use the deadline for video packet delay as 120 ms.
3) *Best-effort Data Model* [5]: It is assumed that best-effort data sources always have packets to transmit. We use the parameter $q = 0.975$ from the geometric distribution for best-effort data packet length, which implies that the average packet length of best-effort data traffic is 40 slots.

In the RT-FCR algorithm, the maximum successive transmission limit of station $i(T_{\mathrm{PkTrans,i}})$ is controlled by the distributed SCFQ algorithm to provide a high degree of fairness. We use the fairness index defined by Jain [19] to evaluate the degree of fairness for each algorithm. This fairness index is defined as

$$\text{FairnessIndex} = \frac{\left(\frac{\sum_i T_i}{\phi_i}\right)^2}{n \cdot \sum_i \left(\frac{T_i}{\phi_i}\right)^2} \tag{8}$$

where n is the number of flows, T_i is the throughput of flow i, ϕ_i is the weight of the flow i (we assume all stations have the same weight in the simulations). From Cauchy-Schwartz inequality,

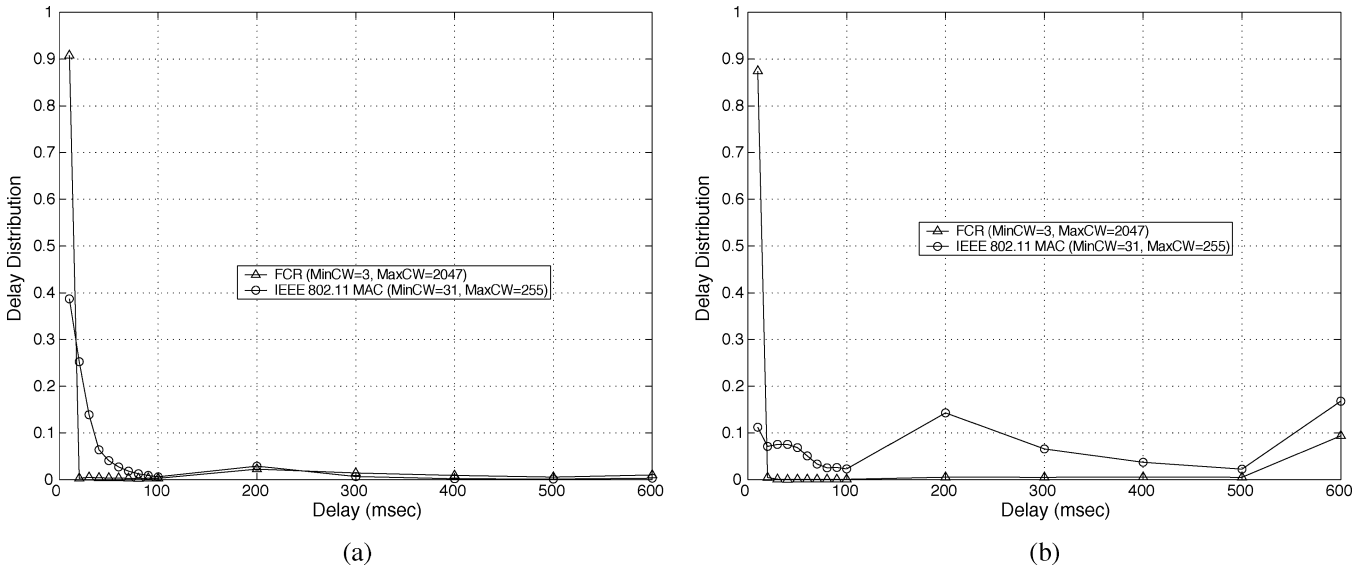

Fig. 5. Delay distribution for FCR algorithm. (a) The 10 BE data station case. (b) The 100 BE data station case.

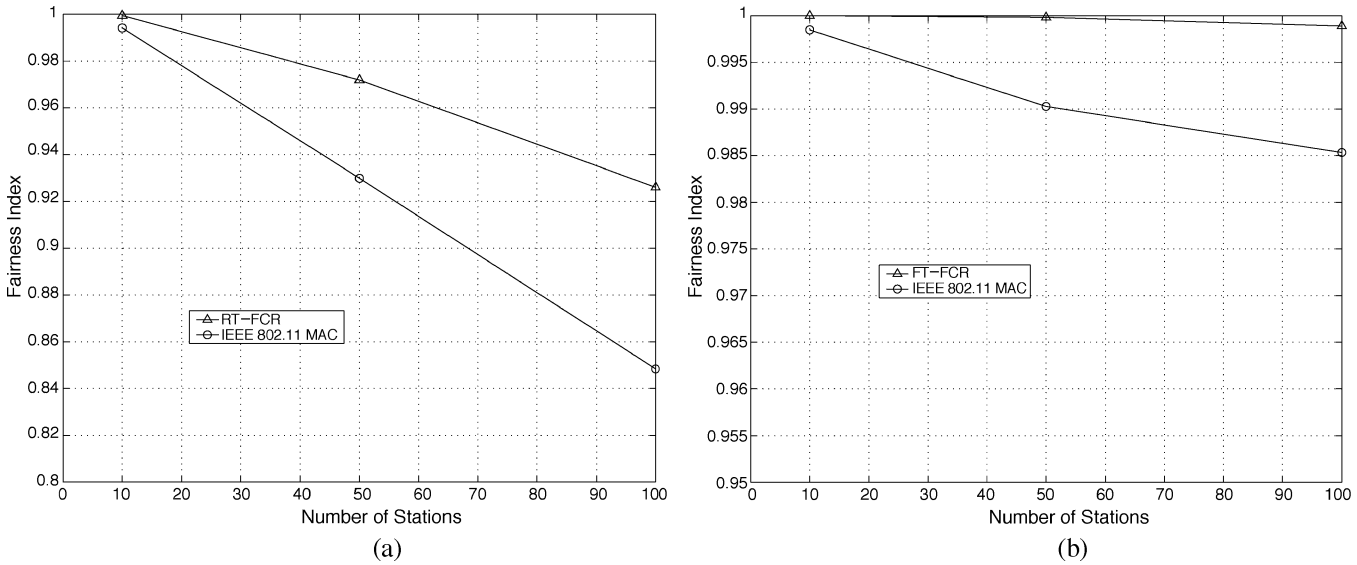

Fig. 6. Fairness index. (a) 10 sec simulation; (b) 100 sec simulation.

we obtain FairnessIndex ≤ 1, the equality holds if and only if all $T_i/\phi_i (i = 1, 2, \ldots, n)$ are equal. Thus, the intuition behind this index is that the higher the fairness index (i.e., closer to 1), the better in terms of fairness. Fig. 6 shows the result of the fairness index of the RT-FCR algorithm and the IEEE 802.11 MAC algorithm for best-effort data traffic transmissions for 10 and 100 seconds simulation time. In Fig. 6, we observe that the RT-FCR algorithm improves the fairness performance in both 10 and 100 simulation time compared to the results for the IEEE 802.11 MAC algorithm.

We present the simulation results of the RT-FCR algorithm for 10 and 100 best-effort data traffic stations by varying the number of CBR voice traffic stations up to 15. We compare the results of the RT-FCR algorithm with those of the IEEE 802.11 MAC algorithm. The ratio of the dropped voice packets to the total generated voice packets is shown in Fig. 7(a), and the throughput for the best-effort data traffic transmissions is shown in Fig. 7(b). In Fig. 7(a), the IEEE 802.11 MAC algorithm loses over 40% of voice packets with 10 best-effort data stations and over 90% with 100 best-effort data stations. This is expected because the IEEE 802.11 DCF mode treats real-time traffic the same as the best-effort data traffic. The ratios of dropped voice packets for the RT-FCR algorithm are close to zero for both cases. The RT-FCR algorithm shows very low voice packet dropping ratio while still preserving high throughput performance for best-effort data traffic, which is obvious in Fig. 7(a) and (b).

In Table VIII and IX, the fairness indexes of best-effort data stations in Fig. 8(b) and (d) are shown. We can see that the fairness for best-effort data stations is highly satisfied while the priority algorithm supports the desired QoS for real-time services.

We also carry out the performance evaluation of the RT-FCR algorithm for the integration of three different traffics: voice, video, and best-effort data. Fig. 8(a)–(d) show the performance results of the RT-FCR algorithm for the integration of three different traffics. The number of best-effort data stations is 10

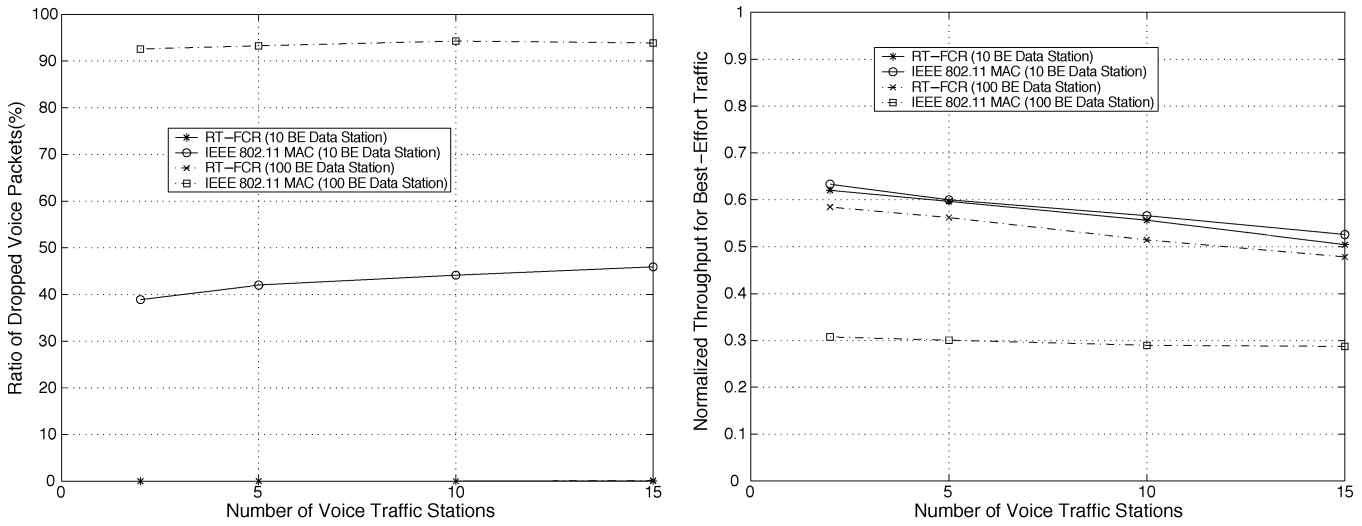

Fig. 7. Performance results of RT-FCR algorithm for voice and data traffic. (a) Ratio of dropped voice packets. (b) Throughput for best-effort data traffic.

TABLE VIII
FAIRNESS INDEX OF BEST EFFORT DATA STATIONS FOR DIFFERENT NUMBER OF CBR STATIONS

Number of CBR Stations	10	30	50	60
Fairness Index	0.9998	0.9908	0.9644	0.9021

TABLE IX
FAIRNESS INDEX OF BEST EFFORT DATA STATIONS FOR DIFFERENT NUMBER OF VBR STATIONS

Number of VBR Stations	2	5	10	15	20
Fairness Index	1.0	0.9998	0.98	0.9453	0.9113

for all simulations. Fig. 8(a) shows that the ratio of the dropped real-time packets to the generated real-time packets versus various numbers of CBR voice stations with 10 best-effort data stations and 5 VBR video stations. The throughput of the best-effort data traffic for this case is shown in Fig. 8(b). In Fig. 8(a) and (b), we can see that the RT-FCR algorithm can support the desired QoS for real-time applications up to 30 CBR stations with ten best-effort data stations and five VBR stations. Fig. 8(c) shows that the result of dropped real-time packets to the generated real-time packets versus various numbers of VBR video stations with ten data stations and five voice stations. The throughput of best-effort data traffic for this case is shown in Fig. 8(d). In Fig. 8(c) and (d), we can see that the RT-FCR algorithm can support the desired QoS for real-time applications up to ten VBR stations with ten best-effort data stations and five voice CBR stations. Fig. 8(a) and (c) show that voice traffic has much higher priority for medium access over video and best-effort data traffics, so the ratio of dropped packet for voice traffic is close to zero for most cases. The ratio of dropped packets for video traffic is affected by best-effort data traffic as the number of CBR stations or VBR stations increases. From the simulation results, we can conclude that the QoS for voice traffic is highly satisfied and the QoS for video traffic is satisfactory in the RT-FCR algorithm. While providing QoS for real-time traffic, the RT-FCR algorithm achieves high throughput for best-effort data traffic when the channel is available for best-effort data traffic transmissions between real-time traffic transmissions, which is shown in Fig. 8(b) and (d).

V. CONCLUSION

In this paper, we propose a new contention-based MAC algorithm, namely, the FCR algorithm. The FCR algorithm can achieve high throughput performance while preserving the implementation simplicity in WLANs. 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 effectively resolve collisions faster. Other ideas we incorporate in the FCR are using smaller minimum contention window size comparing to the IEEE 802.11 MAC and faster decrease of backoff timers after detecting a number of consecutive idle slots. These changes could reduce the average number of idle slots in each contention period, which contributes to the throughput improvement. We extend the FCR algorithm, namely RT-FCR algorithm, to improve the fairness and to provide the QoS for real-time applications. For the fairly scheduling scheme of the RT-FCR algorithm, we use the distributed self-clocked fair queueing algorithm, while the priority scheme based on service differentiations is modified and incorporated into the FCR algorithm to support the QoS for real-time applications. Extensive simulation studies for throughput and delay distribution have demonstrated that the FCR algorithm gives significant performance improvement over the IEEE802.11 MAC algorithm. Simulation results for the RT-FCR algorithm have shown high degree of fairness and low ratio of dropped real-time packets while providing high throughput performance for best-effort data traffic.

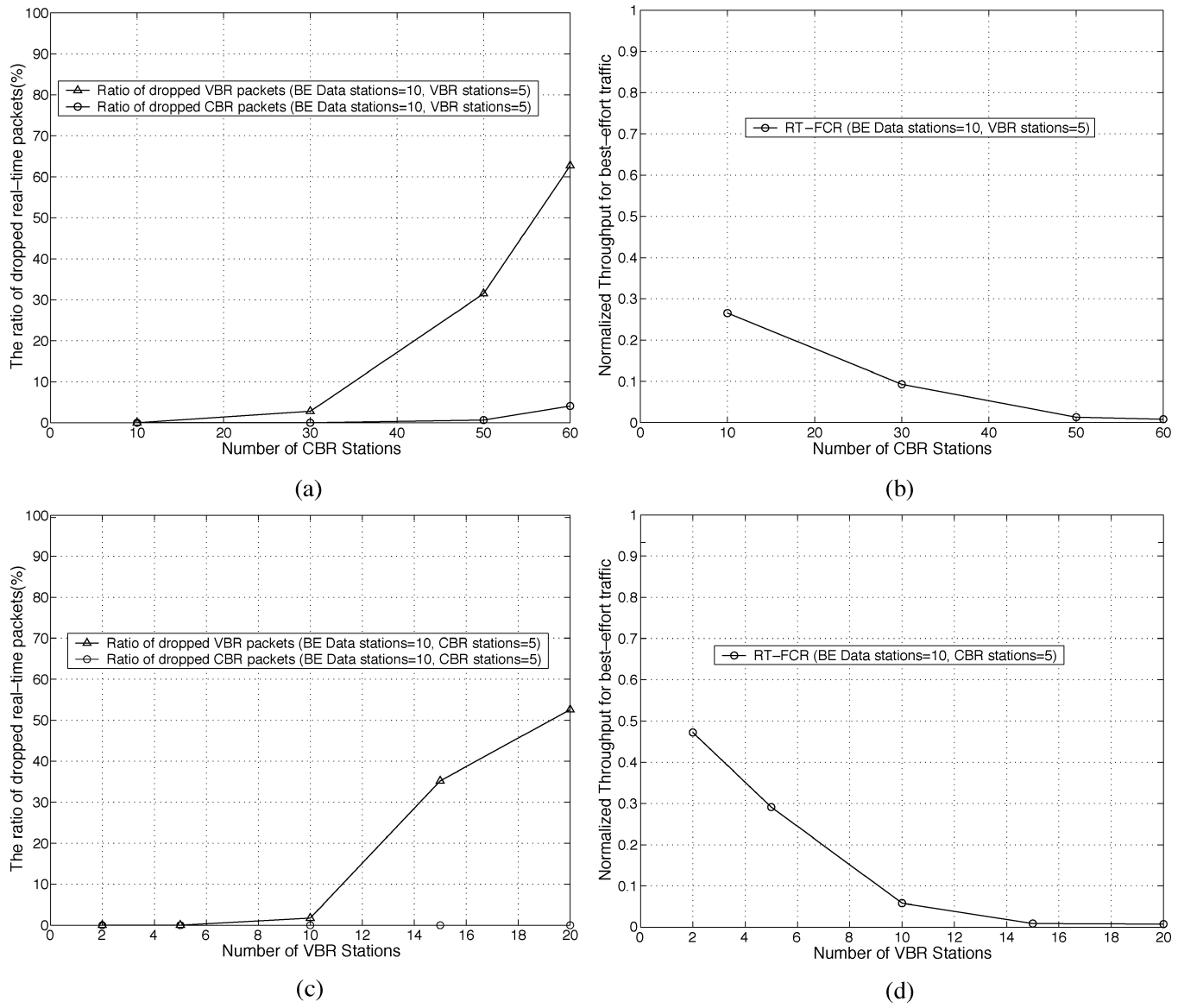

Fig. 8. Performance results of RT-FCR algorithm for mixed real-time traffic transmissions. (a) Ratio of dropped real-time packets versus the number of CBR stations. (b) Throughput for best-effort data traffic versus the number of CBR stations. (c) Ratio of dropped real-time packets versus the number of VBR sttions. (d) Throughput for best-effort data traffic versus the number of VBR stations.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the reviewers and the editor for their constructive comments, which significantly improve the quality of this paper.

REFERENCES

[1] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 209–218.

[2] V. Bharghavan, "MACAW: A media access protocol for wireless LAN's," in *Proc. SIGCOMM'94*, London, U.K., Aug. 1994, pp. 212–225.

[3] V. Bharghvan, "Performance evaluation of algorithms for wireless medium access," in *Proc. IEEE Int. Computer Performance and Dependability Symp.*, 1998, pp. 142–149.

[4] G. Bianchi, "Performance analysis of the IEEE802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 535–547, Mar. 2000.

[5] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Network.*, vol. 8, pp. 785–799, Dec. 2000.

[6] J. Chen, K. M. Sivalingam, P. Agrawal, and R. Acharya, "Scheduling multimedia services in a low-power MAC for wireless and mobile ATM networks," *IEEE Trans. Multimedia*, vol. 1, pp. 187–201, June 1999.

[7] A. Banchs, X. Perez, M. Radimirsch, and H. J. Stuttgen, "Service differentiation extensions for elastic and real-time traffic in 802.11 wireless LAN," in *Proc. IEEE Workshop on High Performance Switching and Routing*, 2001, pp. 245–249.

[8] A. Chandra, V. Gummalla, and J. O. Limb, "Wireless medium access control protocols," *IEEE Commun. Surveys*, pp. 2–15, Second Quarter 2000.

[9] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol. 35, pp. 116–126, Sept. 1997.

[10] J. Deng and R. S. Chang, "A priority scheme for IEEE 802.11 DCF access method," *IEICE Trans. Commun.*, vol. E82-B, no. 1, Jan. 1999.

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

[12] C. Fullmer and J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-ratio networks," in *Proc. SIGCOMM*, Cambridge, MA, 1995, pp. 262–273.

[13] S. J. Golestani, "A self-clocked fair queueing for broadband applications," in *Proc. IEEE INFOCOM* , Apr. 1994, pp. 636–646.
[14] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.*, vol. 37, pp. 885–890, Aug. 1989.
[15] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," *IEEE/ACM Trans. Network.*, vol. 5, pp. 690–704, Oct. 1997.
[16] D. J. Goodman and S. X. Wei, "Efficiency of packet reservation multiple access," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 170–176, Feb. 1991.
[17] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.
[18] IEEE 802.11 Status of Project IEEE 802.11e*http://grouper.ieee.org/groups/802/11/Reports/tge_update.htm* [Online]
[19] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: an explanation," in *Proc. ATM Forum/99-0045*, Feb. 1999.
[20] K. Kim, S. Shin, and K. Kim, "A novel MAC scheme for prioritized services in ieee 802.11a wireless LAN," in *Proc. ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, Joint 4th IEEE Int. Conf.*, 2001, pp. 196–199.
[21] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," in *Proc. ACM SIGMETRICS* , Santa Clara, CA, June 2000.
[22] Y. Kwok and V. K. N. Lau, "A quantitative comparison of multiple access control protocols for wireless ATM," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 796–815, May 2001.
[23] Y. Kwon, Y. Fang, and H. Latchman, "A novel medium access control protocol with fast collision resolution for wireless LANs," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM'03)*, San Francisco, CA, Mar./Apr. 2003.
[24] ——, "Fast collision resolution (FCR) MAC algorithm for wireless local area networks," in *Proc. IEEE Globecom*, Taipei, Taiwan, Nov. 17–21, 2002.
[25] A. Muir and J. J. Garcia-Luna-Aceves, "Group allocation multiple access in single-channel wireless LANs," in *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conf.*, Phoenix, AZ, 1997.
[26] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Network.*, vol. 1, pp. 344–357, June 1993.
[27] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop ad hoc networks?," *IEEE Commun. Mag.*, pp. 130–137, June 2001.
[28] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1353–1368, Aug. 1999.
[29] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proc. Mobicom*, Boston, MA, Aug. 2000.
[30] M. Veeraraghavan, N. Cocker, and T. Moors, "Support of voice services in IEEE 802.11 wireless LANs," in *Proc. IEEE INFOCOM*, vol. 1, 2001, pp. 488–497.

Younggoo Kwon received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, Korea, in 1993 and 1996, respectively, and the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Florida, Gainesville, in 2002.

From 2002 to 2003, he was a Senior Member of the Research Staff at Samsung Electro-Mechanics Central R&D Center. Since 2004, he has been an Assistant Professor in the Department of Computer Engineering, Sejong University, Seoul, Korea. He has authored/coauthored more than ten technical papers and book chapters in the areas of wireless/mobile networks and powerline communications. Currently, he is also an active participant of standardization working group in WLANs and PLC. His current research interests include the area of wireless/mobile and powerline networks with emphasis on the QoS guarantee and adaptation in MAC layer for multimedia home networking based on wireless LAN, UWB, and PLC solutions.

Yuguang Fang (S'92–M'94–SM'99) received the B.S. and M.S. degrees in mathematics from Qufu Normal University, Qufu, Shandong, China, in 1984 and 1987, respectively, the Ph.D degree in systems and control engineering from the Department of Systems, Control and Industrial Engineering, Case Western Reserve University, Cleveland, OH, in 1994, and the Ph.D degree in electrical engineering from the Department of Electrical and Computer Engineering, Boston University, Boston, MA, in 1997.

From 1987 to 1988, he held research and teaching positions in both the Department of Mathematics and the Institute of Automation at Qufu Normal University. From September 1989 to December 1993, he was a Teaching/Research Assistant in the Department of Systems, Control and Industrial Engineering, Case Western Reserve University, where he held a Research Associate position from January 1994 to May 1994. From June 1994 to August 1995, he held a Postdoctoral position in the Department of Electrical and Computer Engineering, Boston University. From September 1995 to May 1997, he was a Research Assistant in the Department of Electrical and Computer Engineering, Boston University. From June 1997 to July 1998, he was a Visiting Assistant Professor in the Department of Electrical Engineering, University of Texas, Dallas. From July 1998 to May 2000, he was an Assistant Professor in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark. Since May 2000, he has been an Assistant Professor in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. His research interests span many areas including wireless networks, mobile computing, mobile communications, automatic control, and neural networks. He has published over 80 papers in refereed professional journals and conferences. He is an Editor for ACM Wireless Networks, an Area Editor for ACM Mobile Computing and Communications Review, an Associate Editor for Wiley International Journal on Wireless Communications and Mobile Computing. He is also actively involved with many professional conferences such as ACM MobiCom'01, IEEE INFOCOM'98, INFOCOM'00, IEEE WCNC'02, WCNC'00 (Technical Program Vice-Chair), WCNC'99, and the International Conference on Computer Communications and Networking (IC3N'98) (Technical Program Vice-Chair).

Dr. Fang is an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, an Editor for IEEE JOURNAL ON WIRELESS COMMUNICATIONS. He is Feature Editor for Scanning the Literature in IEEE PERSONAL COMMUNICATIONS. He has received the prestigious National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He is listed in Marquis *Who's Who in Science and Engineering*, *Who's Who in America*, and *Who's Who in World*. He is a Member of ACM.

Haniph Latchman received the B.S. degree (first class honors) from the University of The West Indies, Trinidad and Tobago, in 1981 and the D.Phil. degree from Oxford University, Oxford, U.K., in 1986.

He joined the University of Florida, Gainesville, in 1987, where he teaches graduate and undergraduate courses and conducts research in the areas of control systems, communications and computer networks. He is also the Director of the Laboratory for Information Systems and Telecommunications and the Codirector of the Research Laboratory for Control System and Avionics. He has published more than 80 technical journal articles and conference proceedings and has given conference presentations in the areas of his research in multivariable and computer control systems and communications and internetworking. He is the author of the books *Computer Communication Networks and the Internet* (New York: McGraw Hill, 1997) and *Linear Control Systems—A First Course* (New York: Wiley, 1999).

Dr. Latchman is an Associate Editor for the IEEE TRANSACTIONS ON EDUCATION. He has received numerous teaching and research awards, including several best-paper awards, the University of Florida Teacher of the Year Award, two University-wide Teaching Improvement Program Awards, College of Engineering Teacher of the Year Awards, the IEEE 2000 Undergraduate Teaching Award, and a 2001 Fulbright Fellowship. He was also the 1983 Jamaica Rhodes Scholar.