Medium Access Control Protocols with Fast Collision Resolution: Supporting Real-Time and Data Services in Wireless LANs *

Younggoo Kwon, Yuguang Fang and Haniph Latchman
Department of Electrical and Computer Engineering, University of Florida
435 Engineering Building, P.O.Box 116130, Gainesville, FL 32611-6130
ykwon@ufl.edu, fang@ece.ufl.edu, latchman@list.ufl.edu

Abstract

Development of ef cient medium access control (MAC) protocols which provide both high throughput performance for data traf c and good quality of service (QoS) support for real-time traf c is the current major focus in distributed contention-based MAC research. In this paper, we propose an ef cient contention algorithm for wireless local area networks, namely the Fast Collision Resolution (FCR) algorithm. The MAC protocol with this new algorithm attempts to provide signi cantly high throughput performance for data services. To support QoS for real-time services, we incorporate the priority algorithm based on service differentiations with the FCR algorithm, and show that this prioritized FCR algorithm can simultaneously achieve high throughput and good QoS support for real-time and data services.

1. Introduction

A good medium access control (MAC) protocol for wireless local area networks (LANs) should provide an ef cient mechanism to share limited spectrum resources, together with simplicity of operation and high throughput. Medium access control algorithms in wireless LANs can be classi ed 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 wireless local area networks where no centralized scheduler exists. In this paper, we focus on the performance issues in such environments. 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 request to send (RTS) and clear to send (CTS) handshaking mechanism for collision avoidance (CA) ([2, 9, 12] and references therein). The most popular contention-based wireless MAC protocol, the carrier sense multiple access/collision avoidance (CSMA/CA), becomes the basis of the MAC protocol for the IEEE 802.11 standard[17]. However, it is observed that if the number of active users increases, the throughput performance of the IEEE 802.11 MAC protocol degrades signi cantly 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, 4, 5] and references therein). To increase the throughput performance of a distributed contention-based MAC protocol, an ef cient 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[2, 3, 5, 11, 12]. 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 ef cient distributed contention-based MAC algorithm, namely, the *fast collision resolution* (FCR) algorithm. We observe that the main de ciency 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

*This work was supported in part by the Of®ce of Naval Research Young Investigator Award under grant N000140210464.

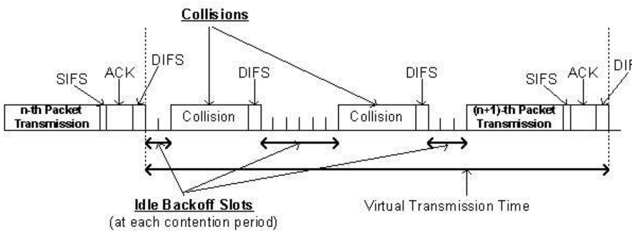


Figure 1. Basic Packet Transmission Structure of CSMA/CA

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 in the large contention window range 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 real wireless local area networks. We extend the FCR algorithm by incorporating the priority algorithm based on service differentiations[1, 10] to support QoS for real-time and data services. The prioritized FCR algorithm can achieve high throughput for best-effort data traf c transmissions while at the same time supporting QoS for real-time applications.

This paper is organized as follows. In the next section, we present the newly proposed the *fast collision resolution* (FCR) algorithm. In Section 3, the prioritized FCR algorithm for real-time and data services is explained. In the nal section, we present the conclusions.

2. Fast Collision Resolution : the 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 Figure 1.

Under high traf c load (i.e., all M stations always have packets to transmit) and under some ergodicity assumption, we can obtain the following expression for the throughput (for example, based on Figure 1, we can examine one transmission cycle)[3, 5]:

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

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

From this result, we can see that the best scenario in Figure 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

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

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, $p_{trans}(i)$, at each contention period as follows:

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

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]:

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

where B_i is the backoff timer of station i.

This means that if station i has the backoff timer 0 (i.e., $B_i = 0$), then its backoff time is 0 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, (3) can be converted to (5):

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

Thus, we conclude that if we could develop a contention-based MAC algorithm, which assigns a backoff timer 0 to the station in transmission while assigns all other stations' backoff timers as ∞ for each contention period, then we could achieve the perfect scheduling, leading to the maximum throughput. Unfortunately, such a contention-based MAC algorithm does not exist in practice. However, this does provide us the basic idea how to improve the throughput performance in the MAC protocol design. We can use the operational characteristics of the perfect scheduling to design more ef cient contention-based MAC algorithm. One way to do so is to design a MAC protocol to approximate the behavior of perfect scheduling.

From (3) and (5), 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 period*: This will decrease the average number of idle slots for each contention period, $E[B_c]$ in (1).

2. *Large random backoff timer for stations that are deferred their packet transmissions at current contention period*: The deferred station means a station which has been suspended its packet transmission with a non-zero backoff timer. Large random backoff timers for deferred stations will decrease the collision probability signi cantly (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 is deferred, its random backoff timer should be set large to avoid the future collisions. The net effect is that all deferred stations will give the successful station more time to nish the back-logged packet transmissions. When a station detects the medium is idle for a xed number of slots during backoff procedure, it would conclude that no other stations are transmitting, and hence it will reduce the backoff timer exponentially to reduce the average idle slots.

2.1 Fast Collision Resolution Algorithm

As we pointed out, the major de ciency 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 deferred 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 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 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 pre xed number of consecutive idle slots are detected.

5. Assign the maximum successive packet transmission limit ($T_{PkTrans}$) to keep fairness in serving users.

Item 1 and 4 attempt to reduce the average number of idle backoff slots for each contention period ($E[B_c]$) in (1). 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 which have 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 that a station dominates packet transmissions for a long period. If a station has performed successive packet transmissions of the maximum successive packet transmission limit ($T_{PkTrans}$), it changes its contention window size to the maximum value (maxCW) to give opportunities for medium access to other stations.

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 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., $BT_{new} = BT_{old}/2$ 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 backoff time is guaranteed to be less than or equal to $18 \times aSlotTime$ for the 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., $CW = \min[maxCW, ((CW + 1) \times 2 - 1)]$, $BT = uniform(0, CW) \times aSlotTime$, 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 which reaches the maximum successive transmission limit (or larger), then it will perform the following actions to give opportunities for the medium access to other stations: $CW = maxCW$,

Table 1. 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

$BT = uniform(0, CW) \times aSlotTime.$

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: $CW = \min[maxCW, ((CW + 1) \times 2 - 1)]$, $BT = uniform(0, CW) \times aSlotTime$.

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.

2.2 Performance Evaluation of the MAC Protocol with Fast Collision Resolution

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 1, which are based on the IEEE 802.11 network configurations[17].

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[5]:

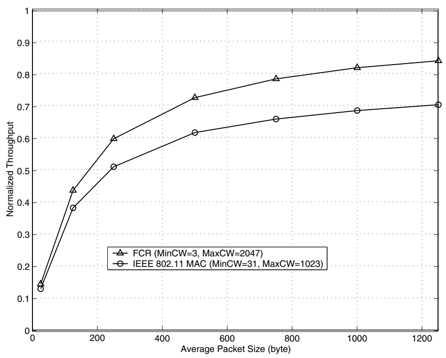
$$P[PacketLength = i\ slots] = q^{i-1}(1-q), \quad i \geq 1.$$

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

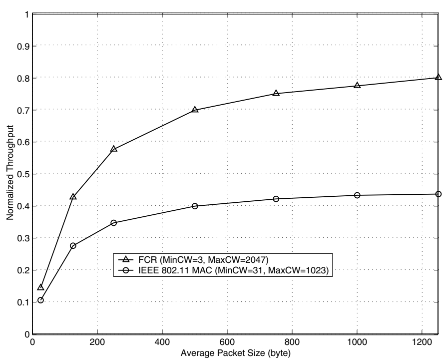
$$\bar{m} = t_s/(1-q) \quad (\mu sec)$$

where t_s is the slot time, i.e., $t_s = aSlot\,Time$.

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



(a) Throughput for 10 BE data stations wireless LAN



(b) Throughput for 100 BE data stations wireless LAN

Figure 2. Throughput Results of FCR Algorithm

Figures 2(a) and 2(b) 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 the FCR algorithm, all stations can quickly obtain the proper contention window size to prevent future collisions, consequently the probability of collisions will be decreased to quite small values. This will reduce the wasted medium idle time to a much smaller value when compared to the IEEE 802.11 MAC algorithm. In Figures 2(a) and 2(b), we can see that the

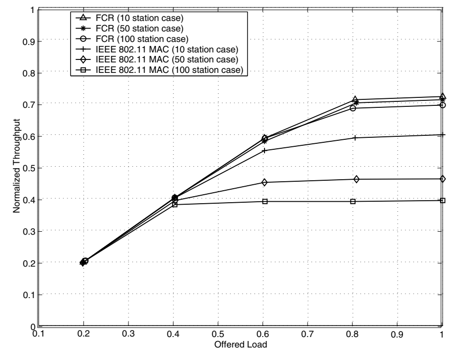
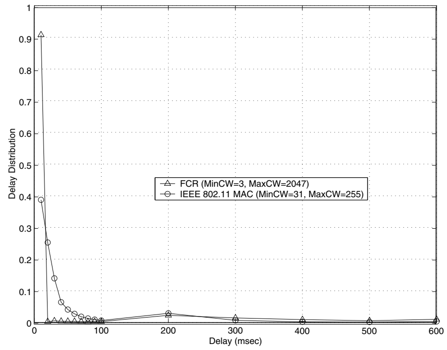


Figure 3. Throughput vs. offered load

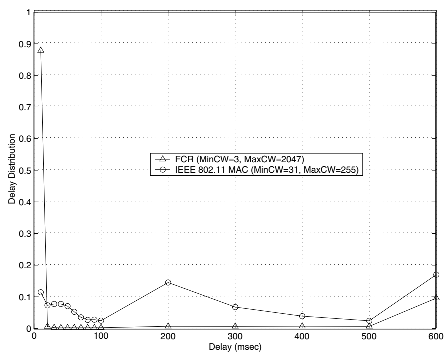
FCR algorithm signi cantly improve the throughput performance over the IEEE 802.11 MAC algorithm. Moreover, the throughput performance of the FCR algorithm are not severely degraded as the number of stations increases because of the highly ef cient collision resolution strategy.

Figure 3 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 2000μsec (500 bytes). We use a traf c generator with Poisson distribution to provide each offered load in this simulation. From Figure 3, we can see that the FCR algorithm also performs very ef ciently 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 2000μsec (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. Figures 4(a) and 4(b) 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 Figure 4(a), 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 Figure 4(b), 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



(a) Delay distribution for 10 stations wireless LAN



(b) Delay distribution for 100 stations wireless LAN

Figure 4. Throughput Results of FCR Algorithm

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.

3 Quality of Service (QoS) Support with Prioritized Fast Collision Resolution (FCR) Algorithm

In order to cope with the QoS requirements of real-time applications, many algorithms have been proposed in contention-based MAC protocols for wireless LANs. The most popular approach is to use a priority scheme for each traf c type, i.e., real-time traf c has higher priority for medium access than best-effort data traf c. With higher priority for medium access, real-time traf c will be served earlier than best-effort data traf c, which results in relative performance improvements for real-time traf c over data traf c.

We give priorities for accessing a medium by assigning different backoff ranges based on each of three main traf c types: voice, video, and best-effort data traf c. Intuitively, the smaller the backoff range is, the higher the priority for accessing a medium. The basic medium access scheme with three different traf c types is shown in Figure 5. We can see that the proposed medium access algorithm effectively provides "soft" reservation to a station for the medium access according to the traf c type which is shown in Figure 5. In this scheme, voice traf c has the highest priority (i.e., the smallest average backoff value), and video traf c has higher priority over best-effort data traf c because of different backoff regions according to the traf c type. The access guaranteed initial backoff range [0, 7] is given to voice traf c, 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. Video traf c uses a much smaller maximum contention window size than best-effort data traf c in order to give higher priority over best-effort data traf c for the medium access, i.e., video traf c will have a smaller average backoff value than data traf c which is shown in Figure 5.

In addition to assigning different backoff ranges, the prioritized FCR algorithm uses different contention algorithms with considering each traf c type. The basic procedures for the priority scheme of the prioritized FCR algorithm are shown in Figure 6 and explained as follows:

1. *Voice Packet*: 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 traf c. It has the access guaranteed initial backoff range [0, 7], which gives the highest priority to voice traf c for accessing the medium. Voice traf c needs repeated packet transmissions in constant time intervals (e.g., only one packet transmission is needed every 30 ms). The FCR algorithm works with high ef ciency for best-effort data traf c transmission, where each active station has more than one packets to transmit. However, in voice traf c transmissions where only one packet transmission is needed every 30 ms, the IEEE 802.11 MAC is more suitable because it does not increase the contention window sizes of the deferred stations. That is, after one station succeeds in transmitting a packet, and leaves the contention session, the remaining stations still keep the same contention window sizes and contend again (in the FCR algorithm, these remaining stations increase the contention window sizes). This results in small wasting idle slots in voice traf c transmissions.

Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02)
0742-1303/02 $17.00 © 2002 **IEEE**

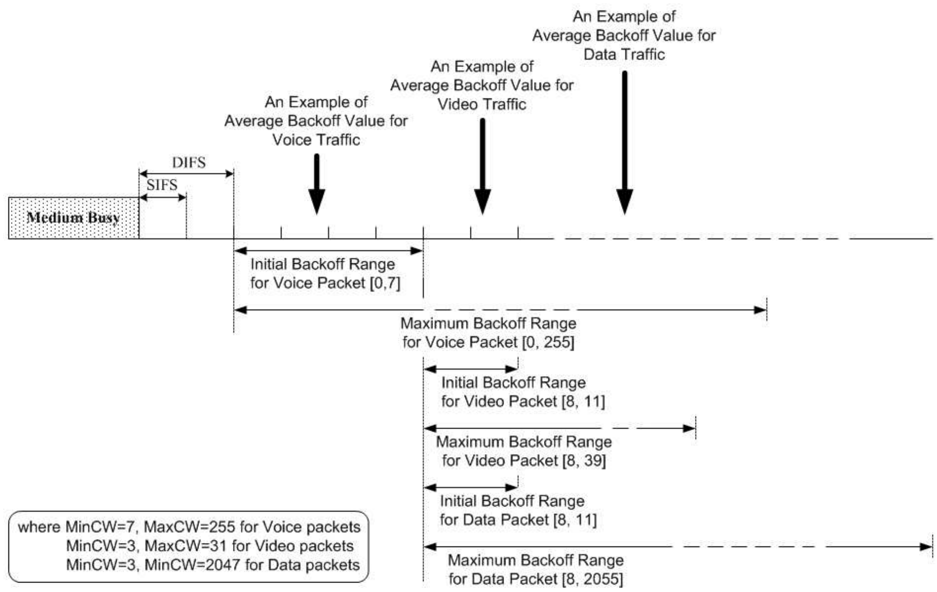


Figure 5. Medium Access Scheme for Real-Time and Data Services

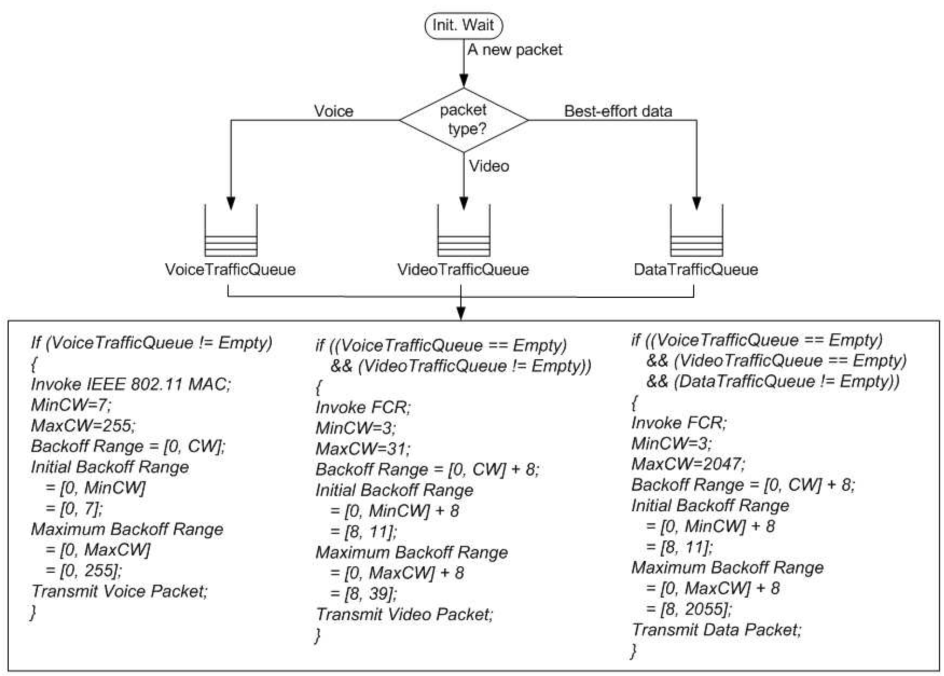


Figure 6. Priority Scheme with FCR Algorithm

2. *Video Packet* : Fast collision resolution (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 backoff range of voice traffic. The smaller maximum contention window size of video traffic (MaxCW=31) than that of best-effort data traffic (MaxCW=2047) gives video traffic higher priority for the medium access over best-effort data traffic.

3. *Best-Effort Data Packet*: Fast collision resolution (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 backoff range of voice traffic. FCR scheme with the large maximum contention window size achieves the high throughput for best-effort data traffic in addition to providing the opportunity for the medium access to voice or video traffic.

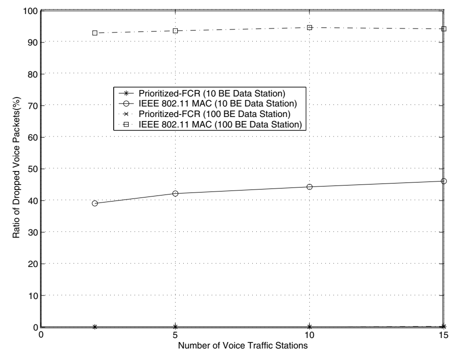
3.1 Performance Evaluations for Prioritized FCR Algorithm

We consider three different types of traffic: constant bit rate (CBR) voice traffic, variable bit rate (VBR) video traffic, and best-effort data traffic. The detailed source models used in our simulations are described as follows:

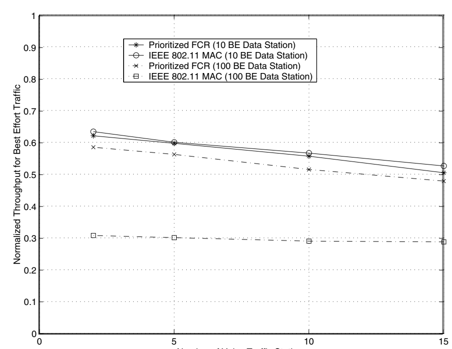
1. *Voice Model*[6, 16]: A voice source has two states, talkspurts and silent gaps identified by a speech activity detector. The probability that a principal talkspurt, with mean duration t_1 second, ends in a time slot of duration τ seconds is $\gamma = 1 - exp(-\tau/t_1)$. The probability that a silent gap, of mean duration t_2 seconds, ends during τ seconds 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 byte payload voice packet every 30 msec during talkspurts period, and we assign the deadline for voice packet delay as 30 msec (i.e., the maximum voice packet delay is 30 msec).

2. *Video Model*[6, 22]: We use the H.263 video traffic with 40 msec 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 bytes and the mean rate of video traffic is 48 kbps and the maximum rate is 480 kbps. That is, there are 2 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 msec.

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.



(a) Ratio of Dropped Voice Packets



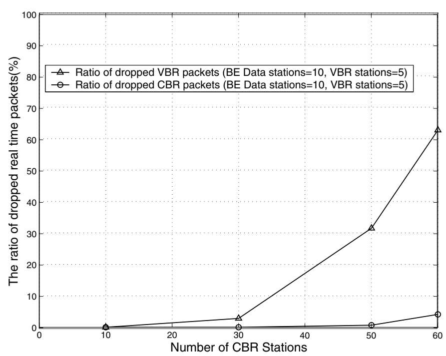
(b) Throughput of Best-Effort Data Traf®c Transmission

Figure 7. Performance Results of Prioritized FCR Algorithm for Voice and Data Traffic Transmissions

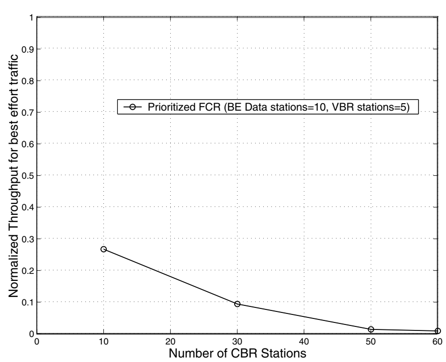
We present the simulation results of the prioritized FCR algorithm for 10 and 100 best-effort data traffic stations with varying the number of CBR voice traffic stations up to 15. We compare the results of the prioritized FCR algorithm with those of the IEEE 802.11 MAC algorithm. The ratio of the dropped voice packets to the total generated voice packets is shown in Figure 7(a), and the throughput for the

Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02)
0742-1303/02 $17.00 © 2002 **IEEE**

best-effort data traf c transmissions is shown in Figure 7(b). In Figure 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 traf c the same as the best-effort data traf c. The ratios of dropped voice packets for the prioritized FCR algorithm are close to zero for both cases. The prioritized FCR algorithm shows very low voice packet dropping ratio while still preserving high throughput performance for best-effort data traf c, which is obvious in Figures 7(a) and 7(b).



(a) Ratio of Dropped Real-Time Packets vs. Number of CBR Stations



(b) Throughput of Best-Effort Data Traf®c vs. Number of CBR Stations

Figure 8. Performance Results of Prioritized FCR Algorithm for Mixed Real-Time Traffic Transmissions

We carry out the performance evaluation of the prioritized FCR algorithm for the integration of three different traf cs: voice, video, and best-effort data. Figure 8(a) and 8(b) show the performance results of the prioritized FCR algorithm for the integration of three different traf cs. The number of best-effort data stations is 10 for all simulations. Figure 8(a) shows that the ratio of the dropped real-time packets to the generated real-time packets vs. various numbers of CBR voice stations with 10 best-effort data stations and 5 VBR video stations. The throughput of the best-effort data traf c for this case is shown in Figure 8(b). In Figure 8(a) and 8(b), we can see that the prioritized FCR algorithm can support the desired QoS for real-time applications upto 30 CBR stations with 10 best-effort data stations and 5 VBR stations. Figure 8(a) shows that voice traf c has much higher priority for channel access over video and best-effort data traf cs, so the ratio of dropped packet for voice traf c is close to zero for most cases. The ratio of dropped packet for video traf c is affected by best-effort data traf c as the number of CBR stations increases. From the simulation results, we can conclude that the QoS for voice traf c is highly satis ed and the QoS for video traf c is satisfactory in the prioritized FCR algorithm. While providing QoS for real-time traf c, the prioritized FCR algorithm achieves high throughput for best-effort data traf c when the channel is available for best-effort data traf c transmissions between real-time traf c transmissions, which is shown in Figure 8(a) and 8(b).

4 Conclusions

In this paper, we propose a new contention-based medium access control algorithm, 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 effectively resolve collisions. 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 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 fast collision resolution (FCR) algorithm to provide the QoS for real-time and data services while preserving the high throughput performance of the FCR algorithm. The priority scheme based on service differentiations is modi ed and incorporated with 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 signi cant performance improvement over the IEEE802.11 MAC algorithm. Simulation results for the prioritized FCR algorithm achieves low ratio of dropped real-time packets while providing high throughput performance for best-effort data traf c.

References

[1] I. Aad and C. Castelluccia, ªDifferentiation mechanisms for IEEE 802.11,º *IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[2] V. Bharghavan , ªMACAW: A Media Access Protocol for Wireless LAN's,º *SIGCOMM'94*, pp.212-225, London, England, Aug. 1994.

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

[4] G. Bianchi, ªPerformance Analysis of the IEEE802.11 Distributed Coordination Function,º *IEEE Journal on Selected Areas in Communications*, Vol.18, No.3, PP.535-547 Mar. 2000.

[5] F. Cali, M. Conti, and E. Gregori, ªDynamin Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit,º *IEEE/ACM Trans. on Networking*, vol. 8, NO. 6, pp.785-799, Dec. 2000.

[6] J. Chen, K. M. Sivalingam, P. Agrawal, and R.Acharya, ªScheduling Multimedia Services in a Low-Power MAC for Wireless and Mobile ATM Networks,º *IEEE Trans. on Multimedia*, Vol.1, NO.2, pp.187-201, June 1999.

[7] A. Banchs, X. Perez, M. Radimirsch, and H. J. Stuttgen, ªService differentiation extensions for elastic and real-time traf®c in 802.11 wireless LAN,º *IEEE Workshop on High Performance Switching and Routing*, pp.245-249, 2001.

[8] A. Chandra, V. Gummalla, and J. O. Limb, ªWireless Medium Access Control Protocols,º *IEEE Communications Surveys*, Second Quarter 2000.

[9] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, ªIEEE 802.11 Wireless Local Area Networks,º *IEEE Communications Magazine* Vol.35, pp.116-126, Sep. 1997.

[10] J. Deng and R. S. Chang, ªA Priority Scheme for IEEE 802.11 DCF Access Method,º *IEICE Trans. Commun.*, Vol.E82-B, NO.1, Jan. 1999.

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

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

[13] S. J. Golestani, ªA Self-Clocked Fair Queueing for Broadband Applications,º *IEEE INFOCOM '94*, pp.636-646, April 1994.

[14] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, ªPacket Reservation Multiple Access for Local Wireless Communications,º *IEEE Transactions on Communications*, vol.37, no.8, pp.885-890, Aug. 1989.

[15] P. Goyal, H. M. Vin, and H. Cheng, ªStart-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks,º *IEEE/ACM Trans. on Networing*, Vol.5, NO.5, pp.690-704, Oct. 1997.

[16] D. J. Goodman and S. X. Wei, ªEf®ciency of Packet Reservation Multiple Access,º *IEEE Trans. on Vehicular Technology*, Vol.40, NO.1, pp.170-176, Feb. 1991.

[17] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) speci®cations, *IEEE*, 1997.

[18] IEEE 802.11 Status of Project IEEE 802.11e, http://grouper.ieee.org/groups/802/11/Reports/tge_update.htm

[19] R. Jain, A. Durresi, and G. Babic, ªThroughput Fairness Index: An Explanation,º *ATM Forum/99-0045*, Feb. 1999.

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

[21] C. E. Koksal, H. Kassab, and H. Balakrishnan, ªAn Analysis of Short-Term Fairness in Wireless Media Access Protocols,º *ACM SIGMETRICS 2000*, Santa Clara, CA, Jun., 2000.

[22] Y. Kwok and V. K. N. Lau, ªA Quantitative Comparison of Multiple Access Control Protocols for Wireless ATM,º *IEEE Trans. on Vehicular Technology*, Vol.50, NO.3, pp.796-815, May, 2001.

[23] A. Muir and J. J. Garcia-Luna-Aceves, ªGroup allocation multiple access in single-channel wireless LANs,º *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference*, Phoenix, AZ, 1997.

[24] A. K. Parekh and R. G. Gallager, ªA generalized processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case,º *IEEE/ACM Trans. on Networking*, Vol. 1. No. 3, pp.344-357, June 1993.

[25] S. Xu and T. Saadawi, ªDoes the IEEE 802.11 MAC Protocol Work Well in Multihop Ad Hoc Networks?º *IEEE Communications Magazine*, June 2001.

[26] J. L. Sobrinho and A. S. Krishnakumar, ªQuality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks,º *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp.1353-1368, Aug. 1999.

[27] N. H. Vaidya, P. Bahl, and S. Gupta, ªDistributed Fair Scheduling in a Wireless LAN,º *Proc. Mobicom'2000*, Boston, MA, USA, Aug. 2000.

[28] M. Veeraraghavan, N. Cocker, and T. Moors, ªSupport of voice services in IEEE 802.11 wireless LANs,º *Proc. of IEEE INFOCOM'2001*, pp.488-497, Vol.1, 2001.