Improving Transport Layer Performance by Using A Novel Medium Access Control Protocol with Fast Collision Resolution 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 efficient medium access control (MAC) protocols is a fundamental research issue in high-speed wireless local area networks (LANs). In this paper, we focus on the performance improvement both of MAC layer and transport layer by using a novel medium access control protocol in high-speed wireless LANs which use carrier sense multiple access/collision avoidance(CSMA/CA). We propose an efficient distributed contention-based MAC protocol, namely, the Fast Collision Resolution (FCR) algorithm, and show that the proposed FCR algorithm provides high throughput and low latency while improving the fairness performance for serving users in wireless LANs. The performance of the FCR algorithm is compared with that of the IEEE 802.11 MAC algorithm via extensive simulation studies both in MAC layer and transport layer. The results show that the FCR algorithm achieves a significantly higher efficiency than the IEEE 802.11 MAC algorithm and is well suited for transport layer protocols such as transmission control protocol (TCP) and user datagram protocol (UDP).

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Network**]: Local and Wide-Area Networks—*Access schemes*; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Algorithms

Keywords

Medium Access Control (MAC), Wireless LANs (WLANs), IEEE 802.11, Backoff, TCP, UDP

1. INTRODUCTION

A good medium access control (MAC) protocol for wireless local area networks (LANs) should provide an efficient mechanism to share limited spectrum resources, together with simplicity of operation and high throughput. MAC protocol research (we focus on distributed contention-based algorithms) in wireless networks started with ALOHA and slotted ALOHA in the 1970s. Later, MACA, MACAW, FAMA and DFWMAC were proposed by incorporating the carrier sense multiple access (CSMA) technique as well as the RTS and CTS handshaking mechanism for collision avoidance (CA) ([2, 9, 12] and references therein). The most popular contention-based wireless MAC protocol, CSMA/CA, becomes the basis of the MAC protocol for the IEEE 802.11 standard([16, 15]). However, it is observed that if 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, 4, 5] and references therein).

To increase the throughput performance of a distributed contention-based MAC protocol, an efficient collision resolution algorithm is needed to reduce the overheads (such as packet collisions and idle slots) in each contention cycle. To this end, many novel collision resolution algorithms have been proposed. For example, improved backoff algorithms are proposed to adjust the increasing and decreasing factors of the contention window size and the randomly chosen backoff values, or dynamically adjust the proper contention window size at each station based on the estimation of the number of active stations; 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]).

Transmission control protocol (TCP) and user datagram protocol (UDP) are the prevalent transport layer protocols which are used with the internet protocol (IP) of the network layer. They support transparent data transfer and perform flow and congestion control, ordering of received data, acknowledgment of correctly received data, etc. TCP and UDP run above the network and MAC layers, therefore, MAC layer protocols for wireless LANs should support TCP and UDP well. However, the low bandwidth and high error rate (even moderate packet loss rate) of the wireless channel can cause severe effects on the performance of the transport layer([25, 24]). The overheads of MAC layer may cause many retransmission segments which are not acknowledged within the retransmission time out (RTO) interval in the TCP operation, and result in performance degradation. Therefore, the evaluation of the proposed MAC algorithms for wireless LANs should be performed over transport layer as well as MAC layer.

Although many innovative distributed contention-based MAC pro-

MSWiM'02, September 28, 2002, Atlanta, Georgia, USA.

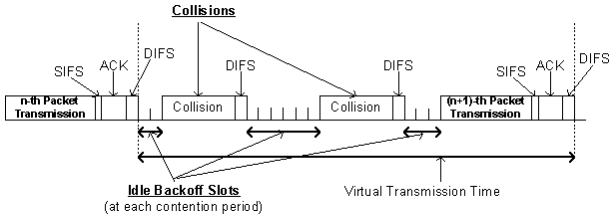


Figure 1: Basic operations of CSMA/CA

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

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

2. THEORETICAL BACKGROUNDS

2.1 IEEE 802.11 Medium Access Control (MAC)

As we mentioned before, the most popular contention-based medium access control (MAC) protocol is the carrier sense multiple access/collision avoidance (CSMA/CA), which is widely used in the IEEE 802.11 LANs. The basic operations of the CSMA/CA algorithm are shown in Figure 1.

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

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

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

After DIFS idle time, the station performs the backoff procedure by using the carrier sensing mechanism to determine whether there is any activity during each 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 ($BT_{new} = BT_{old} - aSlotTime$). If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended. After the medium is determined to be idle for DIFS period, the backoff procedure is resumed. Transmission will begin whenever the backoff timer reaches zero. After a source station transmits a packet to a destination station, if the source station receives an acknowledgment (ACK) without errors after a short inter-frame space (SIFS) idle period, the transmission is concluded to be successfully completed. If the transmission is successfully completed, the contention window (CW) for the source station will be reset to the initial (minimum) value minCW. If the transmission is not successfully completed (i.e., the source station does not receive the ACK after SIFS), the contention window (CW) size will be increased (in the IEEE 802.11 DSSS $CW = 2^{(n+5)} - 1,\ retry\ counter\ n = 0, ..., 5$), beginning with the initial value $minCW$, up to the maximum value $maxCW$ (in the IEEE 802.11 DSSS, $minCW = 31$ and $maxCW = 1023$). This process is called the *binary exponential backoff (BEB)*, which intends to resolve collisions. More detailed operations can be found in ([16]).

2.2 Transport Layer Overview

The transport layer provides end-to-end communication services between different hosts. It makes available transparent data transfer using the services of the network layer below. Therefore, it generally supports various methods of flow control, error recovery and ordering of received data, acknowledgement of correctly received data, and multiplexing and demultiplexing sessions together. Applications and end users of the TCP/IP suite employ one of two protocols from transport layer: the transmission control protocol (TCP) or the user datagram protocol (UDP)([21]). We briefly explain the basic functions for these two protocols.

2.2.1 Transmission Control Protocol (TCP)

TCP is a pervasive transport protocol which gives a dependable data transfer service. It provides reliability for each end host by performing a connection oriented data transfer with supporting diverse flow and congestion control as well as error recovery. If the data segments and acknowledgments are lost, that is, the sender can not receive an acknowledgment for a data segment within predetermined timeout interval, it retransmits the data segment. Therefore, the design strategy for timeout and retransmission has been the main issue to improve the TCP performance([21]).

When delivering large amount of data, a sender should decide the transfer speed considering the receiver's buffer status to avoid network congestions and resulting data loss. Slow start is the procedure that can control the amount of data in-transit between sender and receiver. It works by monitoring the rate that new packets are transferred into the network and the rate that acknowledgments from the receiver are returned. The slow start mechanism counts on the sliding window and congestion window operations. The sliding

window mechanism allows the sender to transmit multiple packets before it stops and waits for an acknowledgment. If a connection is established, the sending host transmits data to the receiving host. The receiver acknowledged with advertising its receiving window size which allows the amount of data the sender can transmit. After the acknowledgement is received, the sender can send additional segments which is limited by the advertised window size of the receiver. Besides, a TCP sender manages its data transfer rate by using the congestion window(cwnd). When a new TCP connection is established, cwnd is set to one segment. Each time an ACK is received, the congestion winodw is increased by one segment. This phase is known as slow start. Therefore, the sender can transmit up to the minimum of the congestion window and the advertised window from the receiver.

If packets get lost because of packet damages in transit or network congestions, TCP operates flow control or congestion control algorithms. Congestion avoidance algorithm is a way to take care of lost packets. Congestion avoidance algorithm operates with slow start by maintaining the congestion window size and the slow start threshold size. If a segment is not acknowledged within some retransmission time out (RTO) interval, TCP performs retransmission to assure a reliable data delivery. If congestion occurs and RTO is expired, TCP assumes that a segment has been lost and retransmits it with setting the congestion window size as one segment and a slow start threshold as one-half of current window (but at least two segments). When new data is acknowledged by the receiver, either slow start or congestion avoidance is performed. If the congestion window is less than or equal to the slow start threshold, the slow start is triggered and increase the congestion window exponentially. Otherwise, congestion avoidance is triggered and the congestion window (cwnd) is increased by 1/cwnd. Slow start continues until the congestion window arrives at the slow start threshold size. This is known as the congestion avoidance phase. If the same segment is lost consecutively, a backoff procedure is invoked, and the RTO is doubled after each retransmission.

If packet loss is detected, TCP slow start and congestion avoidance are performed and degrade the data throughput severely. To overcome this performance degradation, fast retransmit and fast recovery have been designed to speed up the recovery of the connection. Fast retransmit and fast recovery detect a segment loss by monitoring duplicate acknowledgements. When a segment is lost, TCP at the receiver will keep sending ACK segments indicating the next expected sequence number which corresponds to the lost segment. The reception of three or more duplicate ACKs is a strong indication that a segment has been lost. Then, TCP fast retransmit mechanism carries out retransmission of the missing segment even before the retransmission timer expires. If only one or two packets are lost and there is still normal data flows between the two hosts, it is not necessary to reduce the transmission rate rapidly by using slow start. Therefore, fast recovery mechanism performs congestion avoidance instead of slow start, after a fast retransmit of the missing segment.

2.2.2 User Datagram Protocol (UDP)

User datagram protocol (UDP) is defined as a datagram mode of packet-switched computer communication and is a simple, datagram-oriented, connectionless, transport layer protocol([21]). UDP protocol supposes that the internet protocol (IP) is used in the network layer protocol, and performs a procedure for application programs to send messages with a minimum overhead of the protocol mechanism. UDP is transaction oriented, and delivery and duplicate protection are not assured. That is, it sends out the datagrams, but there is no guarantee that they ever reach the receiver. However, a lot of applications are better supported by using UDP because of no connection establishment, small packet overhead, and unfettered transmission rate. UDP encapsulates raw IP datagrams and sends them without having to establishing a connection. Many client-server applications that have one request and one response are much better suited for UDP rather than TCP which establishes and later releases a connection. Under UDP environments, the application is communicating almost directly with IP. UDP takes data packets from application process, attaches source and destination port number fields for the multiplexing/demultiplexing service, and passes the resultant segment to the network layer. The network layer encapsulates the segment into an IP datagram and then transfers the segment to the receiving host. If the segment comes to the receiving host, UDP deliver the data in the segment to the corresponding application process([21]).

3. 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 backoff at each contention cycle, which are shown in Figure 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 Figure 1, we can examine one transmission cycle)([3, 5]):

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

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

From this result, we can see that the best scenario in Figure 1, which gives the maximum throughput, would be the following: a successful packet transmission must be followed by another packet transmission without any overheads, in which case, $E[N_c] = 0, E[B_c] = 0$, the throughput would be

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

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

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

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

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

where B_i is the backoff timer of station i.

This means that if station i has the backoff timer 0 (i.e., $B_i = 0$), then its backoff time is 0 and station i will transmit a packet immediately. Therefore, this can be interpreted as that station i has the probability of packet transmission of 1 at current contention period. If station i has the backoff timer ∞, then its backoff time is also ∞, which can be interpreted as that station i has the probability

of packet transmission of 0 at current contention period. From this discussion, (4) can be converted to (6):

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

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

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

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 (2).

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 non-zero backoff timers. Large random backoff timers for deferred stations will decrease the collision probability at subsequent contention periods (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 as long as possible to increase the useful transmissions. When the station is deferred, its random backoff timers should be as large as possible to avoid the future collisions. The net effect is that all deferred stations will give the successful station more time to finish the back-logged packets. When a deferred station detects the medium is idle for a fixed number of slots, it would conclude that no other stations are transmitting, and hence it will reduce the backoff timers exponentially to reduce the average idle slots.

3.1 Fast Collision Resolution (FCR) Algorithm

As we pointed out, the major deficiency of the IEEE 802.11 MAC protocol comes from the slow collision resolution as the number of active stations increases. An active station can be in two modes at each contention period, namely, the transmitting mode when it wins a contention and the deferring mode when it loses a contention. When a station transmits a packet, the outcome is either one of the two cases: a successful packet transmission or a collision. Therefore, a station will be in one of the following three states at each contention period: a successful packet transmission state, a collision state, and a 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 prefixed number of consecutive idle slots are detected.

5. Assign the maximum successive packet transmission limit 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 (2). 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 due to backoff) 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, 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 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 8*th* idle slot, $BT = 510 \times aSlotTime$ on the 9*th* 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 above scenario. The net effect is that the unnecessary wasted idle backoff time will be reduced when a station, which has just performed a successful packet transmission, runs out of packets for transmission or reaches its maximum successive packet transmission limit.

2. *Transmission Failure (Packet Collision)*: If a station notices that its packet transmission has failed possibly due to packet collision (i.e., it fails to receive an acknowledgment from the intended receiving station), the contention window size of the station will be increased and a random backoff time (BT) will be chosen, i.e., $CW = \min(maxCW, CW \times 2)$, $BT = uniform(0, CW-1) \times aSlotTime$, where $uniform(a, b)$ indicates a number randomly drawn from the uniform distribution between a and b and CW is the current contention window size.

3. *Successful Packet Transmission*: If a station has finished a successful packet transmission, then its contention window size will be reduced to the initial (minimum) contention window size $minCW$ and a random backoff time (BT) value will be chosen accordingly, i.e., $CW = minCW$, $BT = uniform(0, CW-1) \times aSlotTime$. If a station has performed successive packet transmissions which reaches the maximum successive transmission limit (or larger), then its contention window size will be increased to the maximum contention window size $maxCW$ and a random backoff time (BT) value will be chosen as follows: $CW = maxCW$, $BT = uniform(0, CW-1) \times aSlotTime$.

4. *Deferring State*: For a station which is in deferring state, whenever it detects the start of a new busy period, which indicates either a collision or a packet transmission in the medium, the station will increase its contention window size and pick a new random backoff time (BT) as follows: $CW = \min(maxCW, CW \times 2)$, $BT = uniform(0, CW-1) \times aSlotTime$.

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

4. PERFORMANCE EVALUATION

In this section, we present the simulation studies for the proposed fast collision resolution (FCR) algorithm and the IEEE 802.11 MAC protocol in a wireless LAN using direct sequence spread spectrum (DSSS). The parameters used in the simulations are shown in Table 1, which are based on the IEEE 802.11b network configurations([15]). The transmission rates for data and ACK frame are 11 Mbps and 2 Mbps each.

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]):

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

Parameter	Value
SIFS	10 μsec
DIFS	50 μsec
A slot time	20 μsec
aPreambleLength	144 bits
aPLCPHeaderLength	48 bits
Bit rate	2, 11 Mbps

Table 1: Network Configurations

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

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

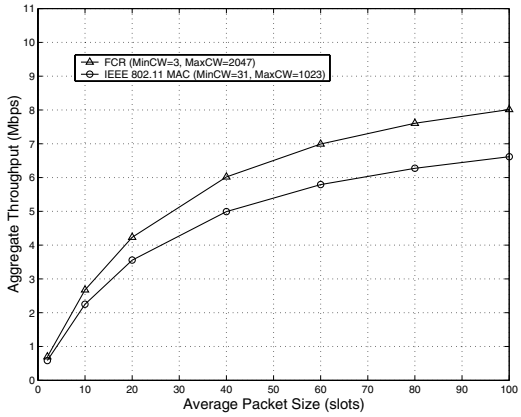
where t_s is the slot time, i.e., $t_s = aSlotTime$.

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

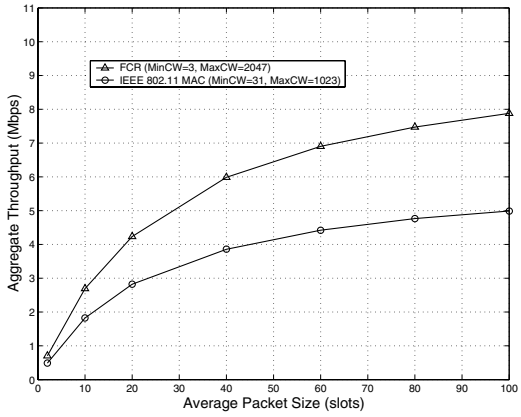
Figures 2(a), 2(b) and 2(c) show the throughput results of the IEEE 802.11 MAC and FCR algorithms for 10, 50, and 100 contending stations, where the average transmission time for a packet (i.e., the average packet length) changes from 10 slots ($q = 0.9$) to 100 slots ($q = 0.99$). The IEEE 802.11 MAC algorithm shows very poor throughput performance as the number of 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 size whenever the system has either a successful packet transmission or has a collision. This means all stations can quickly obtain the proper contention window size to prevent future collisions, consequently the probability of collisions will be decreased to quite small values. At the same time, a station with a successful packet transmission has the minimum contention window size of 3, which is much smaller than the minimum contention window size in the IEEE 802.11 MAC algorithm (minCW=31). 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), 2(b) and 2(c), we can see that the FCR algorithm significantly improve the throughput performance over the IEEE 802.11 MAC algorithm. Moreover, the throughput performance of the FCR algorithm are not degraded much as the number of stations increases because of the highly efficient 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 40 slots ($q = 0.975$). We use a traffic generator with Poisson distribution to provide each offered load in this simulation. The normalized aggregate throughput is shown as the offer load is increased. From Figure 3, we can see that the FCR algorithm also performs very efficiently under light load conditions while providing high throughput as network load increases, and the number of stations hardly affects the performance of the FCR algorithm.

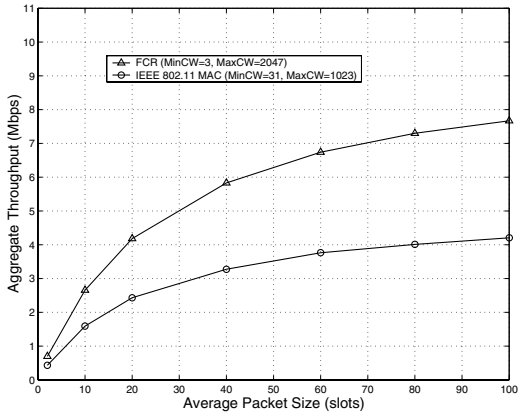
We carry out analysis for the packet delay of the IEEE 802.11 MAC and the FCR algorithm with the average transmission time for a packet (i.e., the average packet length) of 40 slots ($q = 0.975$). The packet delay means the time period from the time when a packet arrives from higher layer to the MAC layer to the time it is successfully transmitted to the intended receiving station. Figures 4(a) shows the average delay of the IEEE 802.11 MAC and



(a) 10 BE data stations wireless LAN



(b) 50 BE data stations wireless LAN



(c) 100 BE data stations wireless LAN

Figure 2: Throughput for Various Number of Stations

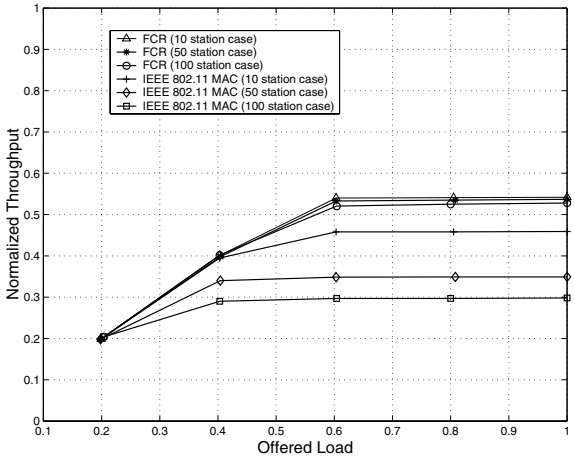
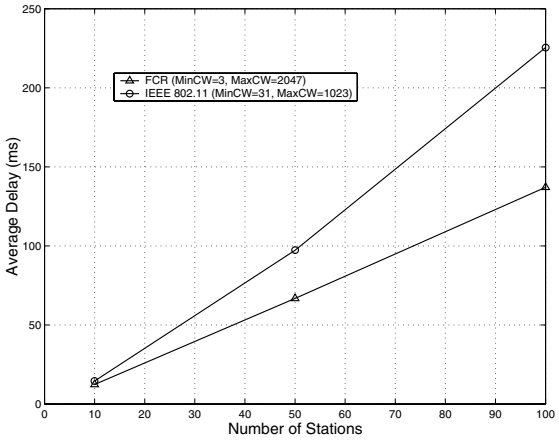


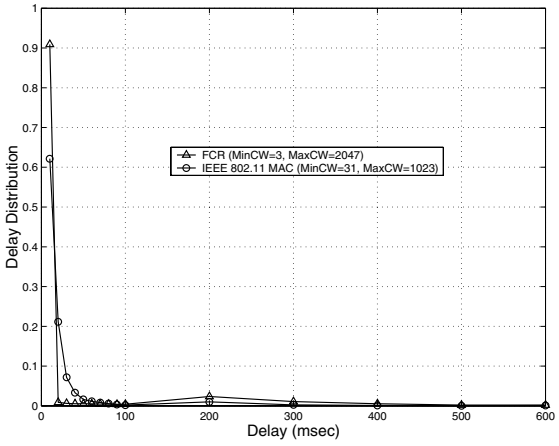
Figure 3: Throughput vs. offered load

the FCR algorithm for 10, 50, and 100 stations wireless LANs. Figures 4(b) and 4(c) show the packet delay distributions for 10 and 100 stations. We have not apply limitation on the number of retries in this simulation for simplicity. In Figure 4(b), the FCR algorithm transmits 91% of all packets successfully within 10 msec while the remaining 9% packets spread over 10 msec to over 600 msec in delay. However, the IEEE 802.11 MAC transmits 62% packets within 10 msec, 21% packets in the range from 10 msec to 20 msec, 7% packets in the range from 20msec to 30 msec, and so on. In Figure 4(c), the FCR algorithm transmits 88% of all packets successfully within 10 msec, while the IEEE 802.11 MAC transmits only 18% packets within 10 msec, 16% packets in the range from 10 msec to 20 msec, 12% packets in the range from 20 msec to 30 msec, and so on. In the simulation results for the packet delay, it is clear that the FCR algorithm transmits most packets successfully within pretty short time, while the IEEE 802.11 MAC transmits packets in much longer time due to collisions, which indeed shows that the FCR algorithm does resolve collision much faster than the IEEE 802.11 MAC algorithm does.

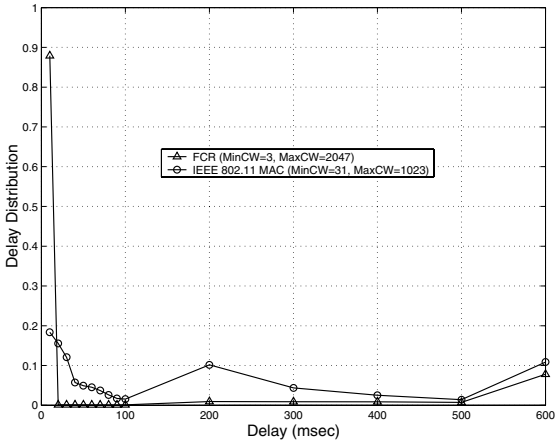
We also run simulations to verify the efficiency of co-operations for the FCR algorithm with the transport layer protocols such as TCP and UDP by using the GloMoSim network simulator([1]). We checked the performance results on the transport layer by using different MAC layer protocols: IEEE 802.11 and FCR. In Figure 5 and 6, the throughput for FTP connections and the throughput and packet delivery ratio for constant bit rate (CBR) traffic are shown. In Figure 5(a), the throughput result is shown for various number of FTP connections(10, 50, and 100). All FTP connections continuously send, from source stations to destination stations, data packets with 1460 bytes, and the simulation time is 100 sec. Figure 5(b) shows the throughput result for CBR stations with the UDP operation. CBR stations generate 1460 byte packets at every 1 ms. In Figure 6(a) and 6(b), the throughput and the packet delivery ratio are shown for voice traffic stations (32 $kbps$ CBR traffic). In Figure 5 and 6, we observe that the FCR algorithm improves the performance (aggregate throughput, packet delivery ratio) of the transport layer compared to the IEEE 802.11 MAC. This means the proposed FCR algorithm supports well the transport layer protocols such as TCP and UDP. From this simulation result, we can see that the efficient collision resolution scheme of the FCR algorithm can also significantly improves the performance at higher layers.



(a) Average Delay

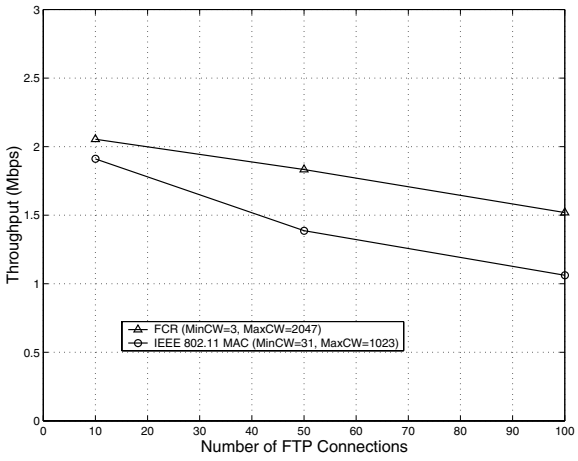


(b) Delay distribution for 10 stations wireless LAN

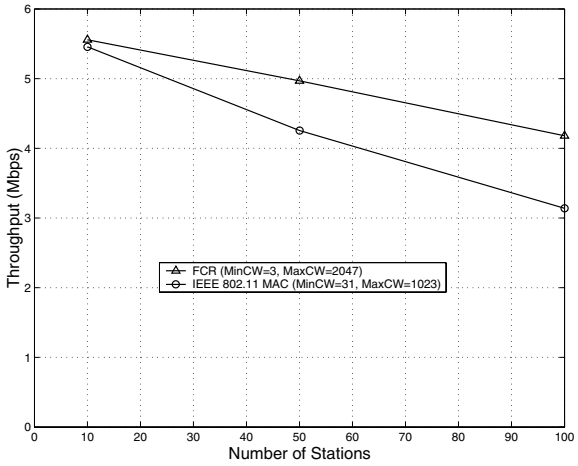


(c) Delay distribution for 100 stations wireless LAN

Figure 4: Delay Performance Results



(a) Throughput result for FTP traffic sources

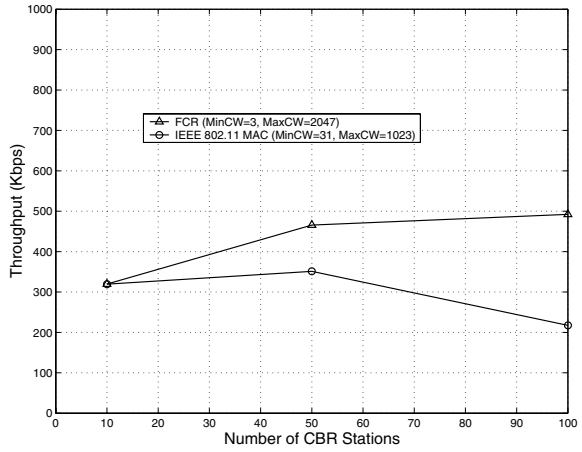


(b) Throughput result for bursty CBR traffic sources

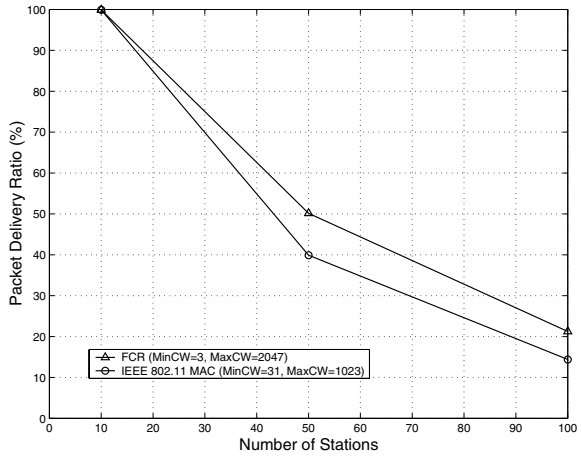
Figure 5: Throughput Results on Transport Layer

5. CONCLUSIONS

In this paper, we propose a new contention-based medium access control algorithms, namely, the fast collision resolution (FCR) algorithm. The FCR algorithm can achieve high throughput performance while preserving the implementation simplicity in wireless local area networks. In the FCR algorithm, each station changes the contention window size upon both successful packet transmissions and collisions (i.e., upon detecting a start of busy period) for all active stations in order to redistribute the backoff timers to actively avoid potential future collisions. Due to this operation, each station can quickly resolve collisions. Other ideas we incorporate in the FCR are using much smaller minimum contention window size compared 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 cycle, which contributes to the throughput improvement. To very the effectiveness for co-operation with the transport layer protocols such TCP and UDP, extensive simulation studies are performed



(a) Throughput result for voice traffic sources



(b) Packet Delivery Ratio

Figure 6: Performance Results on Voice CBR Traffic

for throughput, delay distribution and packet delivery ratio on the transport layer as well as MAC layer. The simulation results show that the proposed FCR algorithm gives significant performance improvement over the IEEE802.11 MAC algorithm, and support the transport layer protocols with high efficiency.

6. REFERENCES

[1] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," UCLA Computer Science Department Technical Report 990027, May 1999.

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

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

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

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

[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. on Multimedia*, Vol.1, NO.2, 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," *IEEE Workshop on High Performance Switching and Routing*, pp.245-249, 2001.

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

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

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

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

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

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

[15] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer Extension in the 2.4 GHz Band, *IEEE*, 1999.

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

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

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

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

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

[21] W. R. Stevens, "TCP/IP Illustrated," Vol. 1, Addison Wesley, 1994.

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

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

[24] G. Xylomenos and G. C. Polyzos, "TCP and UDP performance over a wireless LAN," *Proceedings of the IEEE INFOCOM '99*, pp. 439-446, March 1999.

[25] S. Xu and T. Saadawi, "Does IEEE 802.11 MAC Protocol Work Well in Multi-hop Wireless Ad Hoc Networks?," *IEEE Communication Magazine*, Jun. 2001.