



Performance Analysis for a New Medium Access Control Protocol in Wireless LANs

YOUNGGOO KWON

Samsung Electro-Mechanics Central R&D Center, 314 Maetan 3-dong, Paldal-gu, Kyunggi-do, Suwon, 442-743, Korea

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, USA

Abstract. One fundamental issue in high-speed wireless local area networks (LANs) is to develop efficient medium access control (MAC) protocols. In this paper, we focus on the performance improvement in both MAC layer and transport layer by using a novel medium access control protocol for high-speed wireless LANs deploying carrier sense multiple access/collision avoidance (CSMA/CA). We first present a recently proposed distributed contention-based MAC protocol utilizing a Fast Collision Resolution (FCR) algorithm and show that the proposed FCR algorithm provides high throughput and low latency while improving the fairness performance. The performance of the FCR algorithm is compared with that of the IEEE 802.11 MAC algorithm via extensive simulation studies on both MAC layer and transport layer. The results show that the FCR algorithm achieves a significantly higher efficiency than the IEEE 802.11 MAC and can significantly improve transport layer performance.

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

1. Introduction

Wireless local area networks provide a practical approach for providing high-speed data services in limited geographical areas with reasonable performance for mobility and multi-access characteristics. The medium access control (MAC) protocol is one of the main factors which determine the performance of wireless LANs. A good MAC protocol for wireless local area networks (LANs) should provide an efficient mechanism to share limited spectrum resources together with simplicity of implementation. Wireless MAC protocol research based on distributed contention resolution 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) (see [3,9,12] 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 [15,16]. 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 [4–6] 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 [3,4,6,11,12].

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 degradation on the performance of the transport layer [27,28]. The overheads of MAC layer may cause many retransmissions of segments that 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 carried not only on MAC layer, but also transport layer.

Although many innovative distributed contention-based MAC protocols have been proposed for wireless LANs, it

* Corresponding author.

E-mail: fang@ece.ufl.edu

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 study 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 deferring stations due to prior loss in the contention cycles, 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. The proposed efficient contention scheme in the FCR algorithm will improve the performance both in MAC layer and transport layer. We attempt to keep the proposed distributed contention-based MAC easily implementable in high-speed wireless local area networks.

This paper is organized as follows. In section 2, we briefly describe the IEEE 802.11 MAC protocol and the transport layer protocols: TCP and UDP. Then, we present, in section 3, the newly proposed fast collision resolution (FCR) algorithm. Performance evaluations via simulative study for FCR algorithm on both MAC layer and transport layer are presented in section 4. In the final section, we present the conclusions.

2. Backgrounds

2.1. IEEE 802.11 Medium Access Control (MAC)

IEEE 802.11 MAC protocol is based on the carrier sense multiple access/collision avoidance (CSMA/CA). Its basic operations 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 checks the medium status using the carrier sensing mechanism. If the medium is idle, the transmission may proceed. If the medium is busy, the station will defer until the medium is sensed idle for a distributed coordination function inter-frame space (DIFS) and the backoff procedure is then invoked. The station will set its

backoff timer to a random backoff time based on the current contention window size (CW):

$$\text{Backoff Time (BT)} = \text{Random}() \cdot a\text{SlotTime}, \quad (1)$$

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

After DIFS idle time, the station performs the backoff procedure using the carrier sensing mechanism to determine whether there is any activity during each backoff slot. If the medium is determined to be idle during a particular backoff slot, then the backoff procedure will decrement its backoff time by a slot time ($BT_{\text{new}} = BT_{\text{old}} - a\text{SlotTime}$). 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 then 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 size CW for the source station will be reset to the initial (minimum) value $\min CW$. If the transmission is not successfully completed (i.e., the source station does not receive the ACK after SIFS), the contention window size will be increased (in the IEEE 802.11 DSSS $CW = 2^{n+5} - 1$, *retry counter* $n = 0, \dots, 5$), beginning with the initial value $\min CW$, up to the maximum value $\max CW$ (in the IEEE 802.11 DSSS, $\min CW = 31$ and $\max CW = 1023$). 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 end 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, acknowledgment of correctly received data, and multiplexing and demultiplexing sessions. Applications and end users of the TCP/IP suite employ one of two protocols in transport layer: the *transmission control protocol* (TCP) or the *user datagram protocol* (UDP) [24]. We briefly explain the basic functionalities 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 the support of diverse flow and congestion controls as well as error recovery. If the data segments and acknowledgments are lost, i.e., the sender does not receive an acknowledgment for a data segment within a predetermined timeout interval, it retransmits the data segment. Therefore, the design strat-

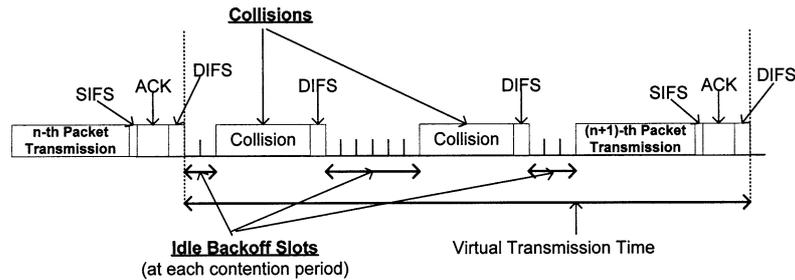


Figure 1. Basic operations of CSMA/CA.

egy for timeout and retransmission has been the main issue to improve the TCP performance [24].

When delivering a large amount of data, a sender should decide the transfer speed based on the receiver's buffer status and packet loss to avoid network congestion. Slow start is the procedure that can control the amount of data in-transit between the sender and the receiver. It controls the rate that new packets are transferred into the network based on the rate of returned acknowledgments from the receiver. 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 acknowledges with advertisement of its receiving window size that allows the amount of data the sender can transmit. After the acknowledgment is received, the sender can send additional segments 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 window 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 a packet gets lost because of packet damages in transit or network congestion, TCP flow control or TCP congestion control algorithms will be activated. 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, it sets the congestion window size to one segment and the slow start threshold as one-half of the current window (but at least two segments). When a new data segment 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 a packet loss is detected, TCP slow start and congestion avoidance are performed, which may degrade the end-to-end throughput severely. To overcome this performance degradation, fast retransmit and fast recovery have been proposed to speed up the recovery of the connection. Fast retransmit and fast recovery detect a segment loss by monitoring duplicate acknowledgments. When a segment is lost, TCP at the receiver will keep sending ACK segments indicating the next expected sequence number corresponding 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 [24]. UDP protocol is used in conjunction with internet protocol (IP) used in the network layer, 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 connection-oriented TCP. Under UDP, 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 reaches the receiving host, UDP delivers the data in the segment to the corresponding application process [24].

3. Fast collision resolution

Two major factors affecting the throughput performance in the IEEE 802.11 MAC protocol are transmission failures (we only consider failures due to packet collisions) and the idle slots due to backoff, 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) [4,6]:

$$\rho = \bar{m}(E[N_c](E[B_c] \cdot t_s + \bar{m} + DIFS) + (E[B_c] \cdot t_s + \bar{m} + SIFS + ACK + DIFS))^{-1}, \quad (2)$$

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

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

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

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

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:

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

where B_i is the backoff timer of station i . From this equation, (4) can be achieved only if

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

This can be interpreted as that station i has the probability of packet transmission of 1 at current contention period when it

has backoff timer is zero while station i has the backoff timer ∞ when others are transmitting.

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. Whenever a node is in successful transmissions, a significantly smaller backoff timer is chosen while other nodes should choose much significantly large backoff timers to have the 0- ∞ effect. To avoid the dominance of the medium by one single node, the time of successive use of the medium by one single node is limited to address the fairness issue. To reduce the number of idle slots due to the fact that the node in transmission does not have enough packets while other nodes may have large backoff timers, we introduce a fast backoff timer reduction algorithm into the FCR.

From such intuitive arguments, we conclude that to achieve high throughput with good fairness, 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 sizes 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.
4. *Limit the successive transmissions by any node.* One way to achieve this is to set an upper bound on the num-

ber of packets successively transmitted without any failures. Whenever a node reaches its limit, it will pick up the maximum contention window size and regenerate the backoff timer.

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, similar to the operations in IEEE 802.11 binary backoff collision resolution algorithm, has the following salient characteristics:

1. Use much smaller initial (minimum) contention window size $\min CW$ than the IEEE 802.11 MAC.
2. Use much larger maximum contention window size $\max CW$ 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.

Items 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) or senses a busy medium at initial access. In the FCR algorithm, the contention window size of a station increases 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 the situation 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 ($\max CW$) 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., $BT_{\text{new}} = BT_{\text{old}} - aSlotTime$ (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 $(\min CW + 1) \cdot 2 - 1$ consecutive idle slots being detected, its backoff timer should be decreased much faster (say, exponentially fast), i.e., $BT_{\text{new}} = BT_{\text{old}} - BT_{\text{old}}/2 = BT_{\text{old}}/2$ (if $BT_{\text{new}} < aSlotTime$, then $BT_{\text{new}} = 0$) or the backoff timer is decreased by a half. For example, if a station has the backoff timer 2047, hence its backoff time is $BT = 2047 \cdot aSlotTime$, which will be decreased by a slot time at each idle slot until the backoff timer reaches 2040 (we assume that $(\min CW + 1) \cdot 2 - 1 = 7$ or $\min CW = 3$). After then, if the idle slots continue, the backoff timer will be decreased by one half, i.e., $BT_{\text{new}} = BT_{\text{old}}/2$ 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 $BT = 1020 \cdot aSlotTime$ on the 8th idle slot, $BT = 510 \cdot aSlotTime$ on the 9th idle slot, $BT = 255 \cdot aSlotTime$ 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 $18 \cdot aSlotTime$ 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., $CW = \min(\max CW, CW \cdot 2)$, $BT = \text{uniform}(0, CW - 1) \cdot aSlotTime$, where $\text{uniform}(a, b)$ 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 $\min CW$ and a random backoff time

value will be chosen accordingly, i.e., $CW = \min CW$, $BT = \text{uniform}(0, CW - 1) \cdot aSlotTime$. 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 $\max CW$ and a random backoff time value will be chosen as follows: $CW = \max CW$, $BT = \text{uniform}(0, CW - 1) \cdot aSlotTime$.

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: $CW = \min(\max CW, CW \cdot 2)$, $BT = \text{uniform}(0, CW - 1) \cdot aSlotTime$.

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.

We remark our scheme FCR bears some similarity to the opportunistic scheduling developed in the literature where transmissions adapts to the channel condition (see [22] and references therein): when the physical channel is good, more data will be transmitted with higher speed. For example, in [22], upon detecting a good channel, the transmitter will change the modulation rate so that more packets can be transmitted during the reserved time period secured by the Network Allocation Vector (NAV), which will obviously increase the throughput without affecting other node transmissions. The integration of this idea with our proposed scheme will be investigated in the future.

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 based on direct sequence spread spectrum (DSSS) technologies. The parameters used in the simulations are shown in table 1, which are based on the IEEE 802.11b network configurations [15]. The

Table 1
Network configuration.

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

transmission rates for data and ACK frame are 11 Mbps and 2 Mbps, respectively. We evaluate the performance of the proposed FCR algorithm on both MAC layer and transport layer.

We first assume that the best-effort data packets are always available at all stations (i.e., in the saturated state) as commonly used in most performance study. In the simulations, the packet lengths for the best-effort data packets are geometrically distributed with parameter q [6]:

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

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

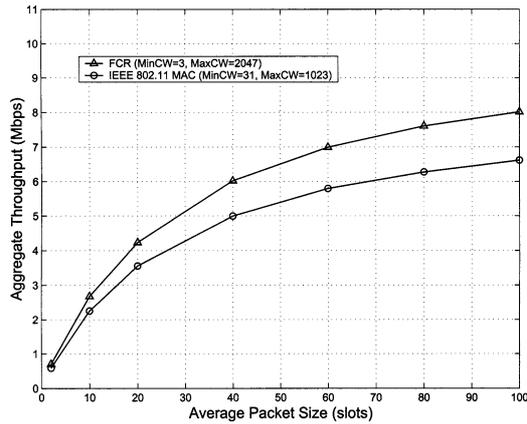
$$\bar{m} = \frac{t_s}{1 - q} \text{ (}\mu\text{s)}$$

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

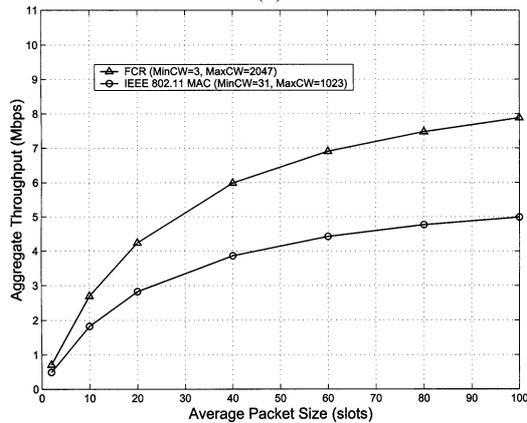
In our study, we assign the maximum successive packet transmission limit of the FCR algorithm as 10. All simulations run for 100 seconds simulation time.

We first study the performance on the MAC layer. Figures 2(a)–2(c) show the throughput results for the IEEE 802.11 MAC and FCR 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 implies that 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 ($\min CW = 31$). 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(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.

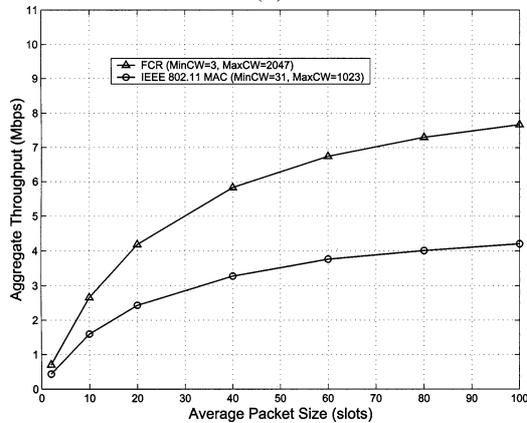
Next, we investigate the performance of both algorithms (IEEE 802.11 MAC and FCR) when the network load varies. This is important because an MAC protocol works well at saturated situation may not work well in less loaded scenarios. Figure 3 shows, respectively, the throughput vs. the offered load for the IEEE 802.11 MAC and the FCR algorithm for 10, 50, 100 stations with the average packet transmission time (i.e., the average packet length) of 40 slots (i.e., $q = 0.975$). We use a traffic generator with Poisson distribution to provide the offered load in this simulation. The normalized aggregate



(a)



(b)



(c)

Figure 2. Throughput for various number of stations. (a) 10 BE data stations. (b) 50 BE data stations. (c) 100 BE data stations.

throughput is shown vs. the offer load. 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. This is due to the fact that the FCR incorporates the dynamic nature of medium access: adapt the medium access to the status of the medium. For the IEEE 802.11 MAC, the throughput performance is severely affected by the traffic load and the number of stations in the wireless LANs.

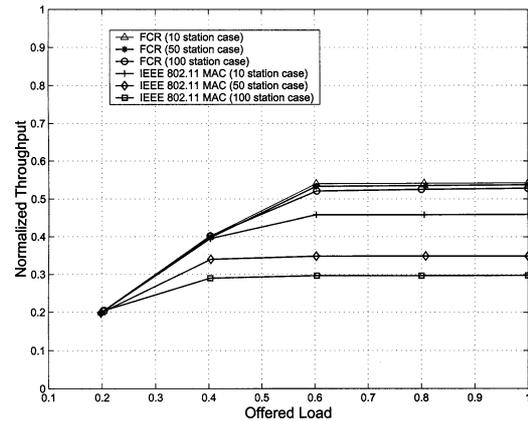
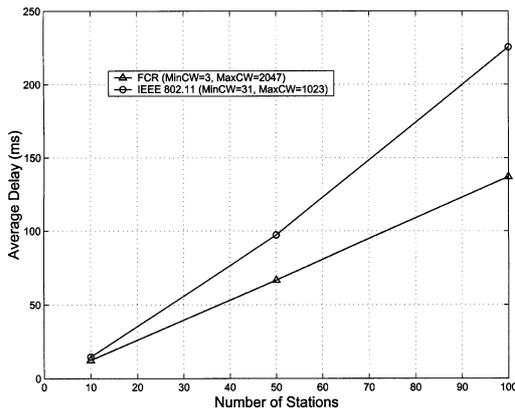


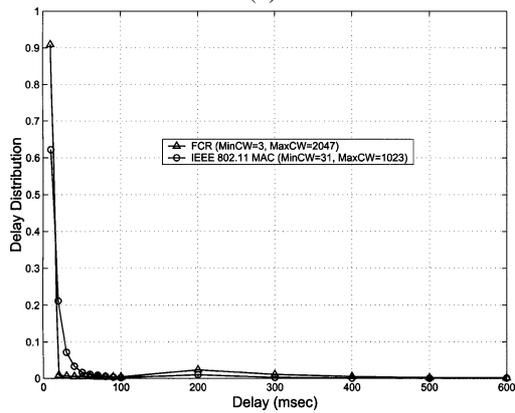
Figure 3. Throughput vs. offered load.

To show our FCR does resolve collisions faster than the IEEE 802.11 MAC does, we carry out the analysis for the packet delay for 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 here is the time period from the time when a packet arrives from the higher layer to the MAC layer to the time it is successfully transmitted to the intended receiving station. It does not take any queuing delay as a station will not generate another packet until the current packet has been successfully transmitted. Figure 4(a) shows the average delay for the IEEE 802.11 MAC and the FCR algorithm for 10, 50, and 100 stations, respectively. Figures 4(b) and 4(c) show the packet delay distributions for 10 and 100 stations, respectively. We have not applied the 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 ms while the remaining 9% packets spread over 10 ms to over 600 ms in delay. However, the IEEE 802.11 MAC transmits 62% packets within 10 ms, 21% packets in the range from 10 ms to 20 ms, 7% packets in the range from 20 ms to 30 ms, and so on. In figure 4(c), the FCR algorithm transmits 88% of all packets successfully within 10 ms, while the IEEE 802.11 MAC transmits only 18% packets within 10 ms, 16% packets in the range from 10 ms to 20 ms, 12% packets in the range from 20 ms 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 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.

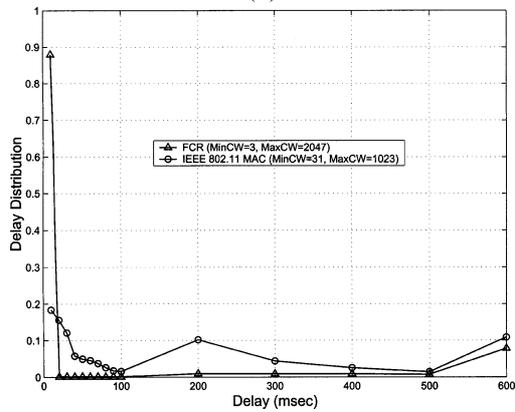
Next, we study the performance on the transport layer. It is well known that the TCP performance degrades significantly in the wireless environments due to the physical layer [27]. Recently, the effect of MAC protocol on the transport layer performance in wireless LANs has also received intensive attention in the last few years [27,28]. Thus, it is also interesting how our proposed FCR performs on the transport layer. In our simulation study, we use the GloMoSim network simulator [1]. We check the performance results on the transport



(a)



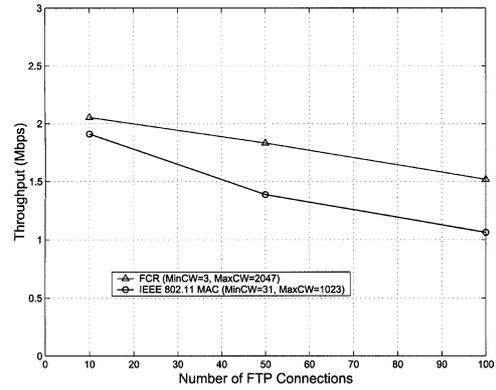
(b)



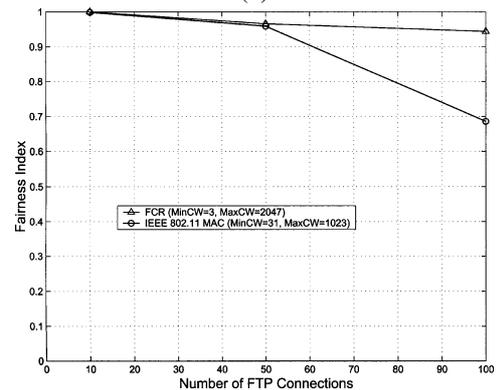
(c)

Figure 4. Delay performance results. (a) Average delay. (b) Delay distribution for 10 stations. (c) Delay distribution for 100 stations.

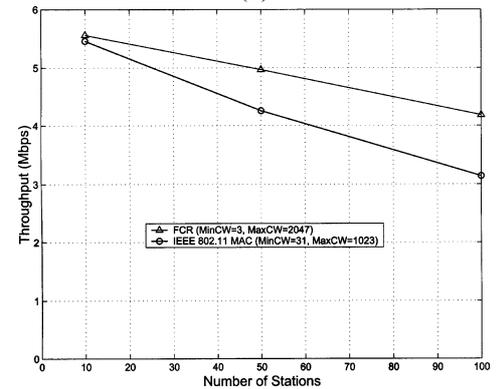
layer by using different MAC layer protocols: IEEE 802.11 and FCR. In figure 5, the throughput and fairness index for TCP and UDP connections are shown. The throughput, fairness and packet delivery ratio for the general voice traffic are shown in figure 6. In figures 5(a) and 5(b), the throughput and fairness results are 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 seconds. The FCR algorithm improves the throughput of FTP connections about 15–35% compared to the result of the IEEE 802.11 MAC.



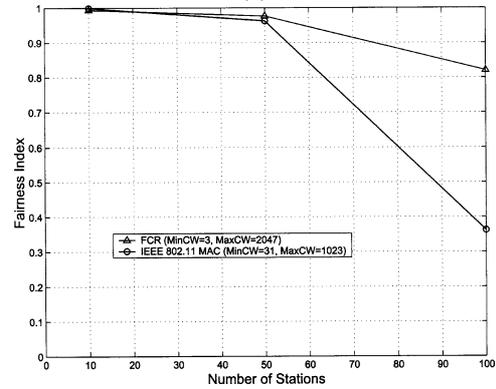
(a)



(b)



(c)



(d)

Figure 5. Throughput results on transport layer. (a) Throughput result for FTP traffic sources. (b) Fairness Index for FTP traffic sources. (c) Throughput result for bursty CBR traffic sources. (d) Fairness Index for bursty CBR traffic sources.

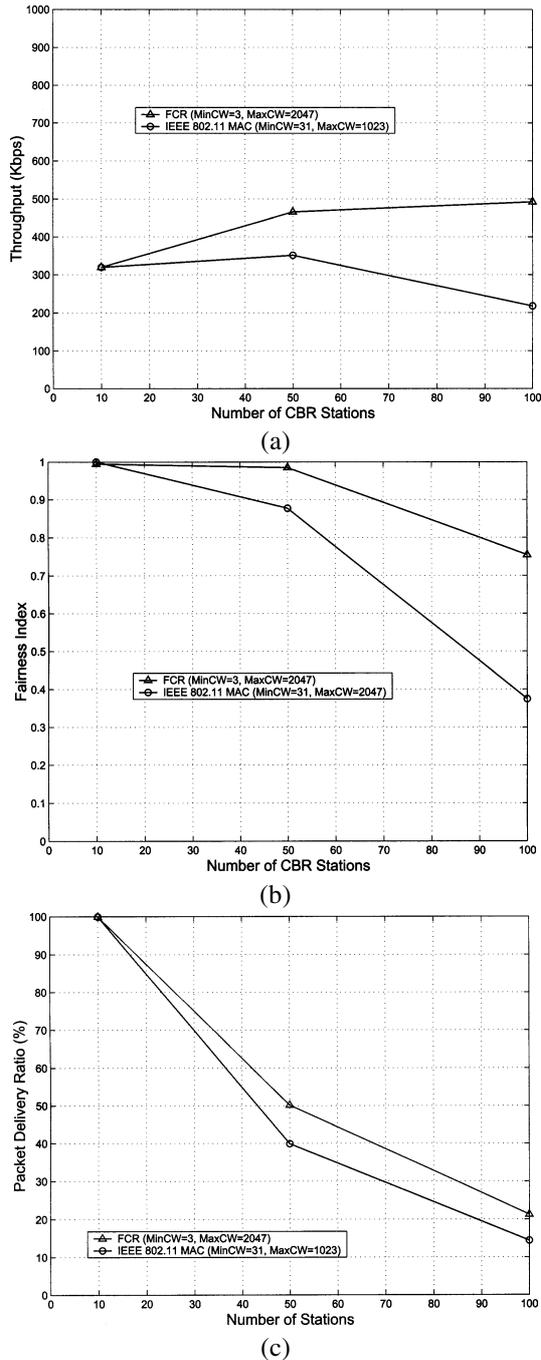


Figure 6. Performance results on voice CBR traffic. (a) Throughput result for voice traffic sources. (b) Fairness Index for voice traffic sources. (c) Packet delivery ratio.

The Jain's fairness index [17] is used to evaluate the degree of fairness for each algorithm, which is defined as follows:

$$FairnessIndex = \frac{(\sum_f T_f / \phi_f)^2}{n \sum_f (T_f / \phi_f)^2}, \quad (7)$$

where n is the number of flows, T_f is the throughput of flow f , ϕ_f is the weight of flow f . Given the proportional share ϕ_f of the medium access, the higher the index, the better in terms of fairness. Since in our scenario, all nodes are treated

equal, hence we can choose $\phi_f = 1$. It can be shown by Cauchy-Schwartz's inequality that $FairnessIndex \leq 1$, and the equality holds if and only if all nodes have the same throughput. Thus, the intuition behind this index is that the higher the fairness index (i.e., closer to 1), the better in terms of fairness. From figure 5, we observe that the FCR algorithm keeps a high level of fairness while the fairness of the IEEE 802.11 MAC is poor as the number of FTP connections increases.

Figures 5(c) and 5(d) show the throughput and fairness results for bursty CBR stations with the UDP operation. CBR stations generate 1460 byte packets at every 1 ms. The performance results of the UDP operation for bursty CBR traffic show similar pattern to those of the TCP operation for FTP connections. In figures 6(a)–6(c), the throughput, fairness and packet delivery ratio are shown for voice traffic stations (32 Kbps CBR traffic) as the number of stations increases up to 100 stations. The overhead come from the contention procedure of the large number of voice stations with small size of voice packets (120 byte payload) result in the low throughput of figure 6(a).

In figures 5 and 6, we observe that the FCR algorithm improves the performance (aggregate throughput, fairness, and 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 study, we can see that the efficient collision resolution scheme in the FCR algorithm can also significantly improve the performance at higher layers.

Finally, we observe that we have taken multiple features into our new protocol FCR: assigning smaller minimum contention window size for nodes with successful transmissions, active backoff for deferring nodes, and fast decrease for back-off timer count-down. One interesting question is how each factor affects the performance. More elaborative evaluation of this new protocol will be investigated in the future.

5. Conclusions

In this paper, we present 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 show the effectiveness of FCR in cooperation with the transport layer protocols such as TCP and

UDP, extensive simulation studies are performed for throughput, fairness, 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 IEEE 802.11 MAC algorithm, and support the transport layer protocols with high efficiency.

Acknowledgements

The authors would like to express their sincere gratitude to the anonymous reviewers for the constructive comments, which not only greatly improve the quality of this paper, but also help the authors' further research along this direction. This paper is the extension of a paper presented at ACM MSWiM'2002 held at Atlanta, Georgia on September 28, 2002. This work was supported in part by US National Science Foundation under grant ANI-0093241 and ANI-0220287, and US Office of Research under grant N000140210464 and N000140210554.

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] A. Banchs, X. Perez, M. Radimirsch and H.J. Stutgen, Service differentiation extensions for elastic and real-time traffic in 802.11 wireless LAN, in: *IEEE Workshop on High Performance Switching and Routing* (2001) pp. 245–249.
- [3] V. Bharghavan, MACAW: A media access protocol for wireless LAN's, in: *SIGCOMM'94*, London, England (August 1994) pp. 212–225.
- [4] V. Bharghavan, Performance evaluation of algorithms for wireless medium access, in: *IEEE International Computer Performance and Dependability Symposium IPDS'98* (1998) pp. 142–149.
- [5] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, *IEEE Journal on Selected Areas in Communications* 18(3) (2000) 535–547.
- [6] F. Cali, M. Conti and E. Gregori, Dynamic tuning of the IEEE 802.11 Protocol to achieve a theoretical throughput limit, *IEEE/ACM Transactions on Networking* 8(6) (2000) 785–799.
- [7] 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 Transactions on Multimedia* 1(2) (1999) 187–201.
- [8] A. Chandra, V. Gummalla and J.O. Limb, Wireless medium access control protocols, *IEEE Communications Surveys* 3(2) (2000) 2–15.
- [9] B.P. Crow, I. Widjaja, J.G. Kim and P.T. Sakai, IEEE 802.11 wireless local area networks, *IEEE Communications Magazine* 35 (September 1997) 116–126.
- [10] J. Deng and R.S. Chang, A priority scheme for IEEE 802.11 DCF access method, *IEICE Trans. Communications* E82-B(1) (1999) 96–102.
- [11] HIPERLAN Type 2 Standard, ETSI (2000).
- [12] C. Fullmer and J. Garcia-Luna-Aceves, Floor acquisition multiple access (FAMA) for packet-ratio networks, in: *Proc. SIGCOMM'95*, Cambridge, MA (1995) pp. 262–273.
- [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* 37(8) (1989) 885–890.
- [14] P. Goyal, H.M. Vin and H. Cheng, Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks, *IEEE/ACM Transactions on Networking* 5(5) (1997) 690–704.
- [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] R. Jain, A. Duresi and G. Babic, Throughput fairness Index: An explanation, in: *ATM Forum/99-0045* (February 1999).
- [18] K. Kim, S. Shin and K. Kim, A novel MAC scheme for prioritized services in IEEE 802.11a wireless LAN, in: *ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, Joint 4th IEEE International Conference* (2001) pp. 196–199.
- [19] Y. Kwok and V.K.N. Lau, A quantitative comparison of multiple access control protocols for wireless ATM, *IEEE Transactions on Vehicular Technology* 50(3) (2001) 796–815.
- [20] Y. Kwon, Y. Fang and H. Latchman, Improving transport layer performance by using a novel medium access control protocol in wireless LANs, in: *Proc. of ACM MSWiM*, Atlanta, GA (2002).
- [21] 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 Conference*, Phoenix, AZ (1997).
- [22] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly, Opportunistic media access for multirate ad hoc networks, in: *Proc. of ACM MobiCom'2002*, Atlanta, GA, USA (September 2002).
- [23] 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* 17(8) (1999) 1353–1368.
- [24] W.R. Stevens, *TCP/IP Illustrated* Vol. 1 (Addison-Wesley, 1994).
- [25] N.H. Vaidya, P. Bahl and S. Gupta, Distributed fair scheduling in a wireless LAN, in: *Proc. Mobicom'2000*, Boston, MA, USA (August 2000).
- [26] M. Veeraraghavan, N. Cocker and T. Moors, Support of voice services in IEEE 802.11 wireless LANs, in: *Proc. of IEEE INFOCOM'2001*, Vol. 1, (2001) pp. 488–497.
- [27] G. Xylomenos and G.C. Polyzos, TCP and UDP performance over a wireless LAN, in: *Proceedings of the IEEE INFOCOM'99* (March 1999) pp. 439–446.
- [28] S. Xu and T. Saadawi, Does IEEE 802.11 MAC Protocol work well in multi-hop wireless ad hoc networks?, *IEEE Communication Magazine* 39 (2001) 130–137.



Younggoo Kwon received the B.S. and M.S. degrees in electrical engineering at Korea University, Seoul, Korea in 1993 and 1996, respectively, and a Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering from University of Florida, Gainesville, in August 2002. He is a senior member research staff at Samsung Electro-Mechanics Central R&D Center, Suwon, Korea. His current research interests are in 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. He has authored/coauthored more than 10 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 wireless LAN and PLC.

E-mail: younggoo.kwon@samsung.com

Yuguang Fang received the B.S. and M.S. degrees in mathematics from Qufu Normal University, Qufu, Shandong, China, in 1984 and 1987, respectively, and a Ph.D. degree in systems and control engineering from the Department of Systems, Control, and Industrial Engineering at Case Western Reserve University, Cleveland, OH, in January 1994, and a Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering at Boston University, MA, in May 1997.

From June 1997 to July 1998, he was a visiting assistant professor in the Department of Electrical Engineering at the University of Texas at Dallas. From July 1998 to May 2000, he was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology, Newark. Since May 2000, he has been an assistant professor in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville. His research interests span many areas including wireless networks, mobile computing, mobile communications, automatic control, and neural networks. He has published over eighty (80) technical papers in refereed journals and conferences. He is an editor for IEEE Transactions on Communications, an editor for IEEE Transactions on Wireless Communications, an editor for ACM Wireless Networks, an editor for IEEE Wireless Communications Magazine, an area editor for ACM Mobile Computing and Communications Review, and an editor for Wiley International Journal on Wireless Communications and Mobile Computing. He was an editor for the IEEE Journal on Selected Areas in Communications: Wireless Communications Series. He has also been actively involved with many professional conferences such as ACM MobiCom'02, ACM MobiCom'01, IEEE INFOCOM'04, INFOCOM'03, INFOCOM'00, INFOCOM'98, 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).
E-mail: fang@ece.ufl.edu

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, UK, in 1986. He was the 1983 Jamaica Rhodes Scholar. 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 and is Director of the Laboratory for Information Systems and Telecommunications and Co-director of the Research Laboratory for Control System and Avionics. He is also an Associate Editor for the IEEE Transactions on Education. Dr. Latchman 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 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* (McGraw-Hill, New York) and *Linear Control Systems – A First Course* (Wiley, New York).
E-mail: latchman@list.ufl.edu