Scheduling-based Coordination Function (SCF) in WLANs for High Throughput

Hojin Lee, Jaeyoung Choi, Taekyoung Kwon, and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {lumiere, jychoi}@mmlab.snu.ac.kr and {tkkwon, yhchoi}@snu.ac.kr

***Abstract*— IEEE 802.11 WLAN has been widely accepted throughout the world. However, it has large overhead due to idle backoff slots and frequent collisions depending on the number of nodes. In this paper, we propose a novel medium access control scheme, Scheduling-based Coordination Function (SCF). SCF polls the next station to be serviced by piggybacking. The selection of the next station is based on Self-Clocked Fair Queueing (SCFQ) scheduling in a distributed manner. Due to polling, SCF does not suffer from collisions and is efficient owing to the SCFQ scheme. SCF improves the system throughput up to 63.2% compared to IEEE 802.11 DCF. Comprehensive simulation is performed to compare SCF with DCF, PCF, and Fast Collision Resolution (FCR).**

I. Introduction

During the last decade, IEEE 802.11 Wireless LANs [7] have prevailed throughout the world. IEEE 802.11 specifies two functions to access the wireless medium: Distributed Coordination Function (DCF), and Point Coordination Function (PCF). DCF is mandatory and based on CSMA/CA for best-effort services while PCF is optional and a point coordinator (access point) polls the stations in a round robin manner for time-bounded services. Most of IEEE 802.11-compliant products support only DCF, because the scheduling algorithm of PCF is inefficient due to inflexible polling group management and it does not provide time-bounded service properly.

The performance of DCF is highly dependent on the number of active nodes in the network [4]. If the number of active nodes is too small, most of time slots are wasted due to idle backoff slots. On the other hand, if the number of active nodes is too large, frequent collisions may occur in spite of a contention resolution mechanism in DCF. Thus, DCF is not the optimal solution for WLANs over a wide range of the number of active nodes. PCF is devised for time-bounded services, i.e., real-time traffic. However, PCF cannot guarantee QoS parameters (delay, bandwidth, jitter, etc) strictly because the start of PCF's polling may be delayed and it has a simple round robin scheduler. To make it worse, PCF polls non-active nodes and hence is not efficient.

In order to enhance the efficiency of the collision resolution mechanism in DCF, many studies have conducted. Kwon *et al*. [5] propose the Fast Collision Resolution (FCR) to speed up the collision resolution. To reduce the average number of idle backoff slots, FCR uses smaller contention window (CW) size for nodes with successful packet transmissions and reduces the backoff slots exponentially. Choi *et al*. [1] propose the Early Backoff Announcement (EBA) to reduce collision probability. In EBA, a node announces its future backoff information. All the nodes receiving the information avoid collisions by excluding the same backoff when selecting their future backoff value. Yang *et al*. [6] propose Dual-Stage Contention Resolution (DSCR) to reduce both idle overhead and collision overhead. DSCR has two steps of contention resolution. The first step of contention resolution is performed not when the wireless medium is idle but also when other node transmits. Nodes which go through the first step of contention resolution perform the second step similarly to ones under IEEE 802.11 DCF. All of the above approaches enhance the throughput of DCF. However, since they are based on contention, they cannot eliminate collisions completely. Thus, throughput will still be degraded when the number of nodes increases.

In this paper, we propose a novel medium access control scheme, Scheduling-based Coordination Function (SCF), which polls the next node to be serviced by piggybacking. The selection of the next node is based on a Self-Clocked Fair Queueing (SCFQ) scheduling scheme [2] in a distributed manner. In SCF, each node calculates its service finish time of its pending packet by SCFQ and informs the access point (AP) of it by piggybacking into a data frame. The AP then selects the node to be serviced by the piggybacked information among nodes and polls it by piggybacking into an ack or a data frame. Due to polling, SCF does not suffer from collisions and is efficient owing to the SCFQ scheme.

The rest of this paper is organized as follows. Preliminaries are presented in Section II. The details of SCF are described in Section III. Section IV presents simulation results. Finally, the paper is concluded with the future work in Section V.

II. Preliminaries

SCF is based on emulating Self-Clocked Fair Queueing (SCFQ) in a distributed manner. We first describe the centralized SCFQ algorithm [2] which assumes the architecture depicted in Fig. 1. A virtual clock is maintained by the central coordinator, and v(t) denotes the virtual time at real time t. Let P_i^k denote the k-th packet arriving on flow i. Let L_i^k denote the length of P_i^k. Let w_i denote the weight of flow i. On arrival of a packet P_i^k at real time t, the packet is stamped with a finish tag, F_i^k, which is calculated as

$$F_i^k = \max(v(t), F_i^{k-1}) + L_i^k / w_i \quad (1)$$

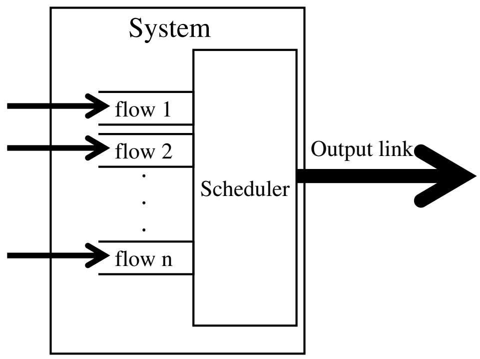

Fig. 1. The centralized architecture.

where F_i^0 = 0. Initially the virtual time is set to 0. The virtual time is updated to the finish tag of the packet which is currently transmitted on the output link. Packets stamped with finish tags are transmitted on the output link in the increasing order of their finish tags.

III. Scheduling-based Coordination Function

SCF is a polling scheme based on SCFQ. It is composed of two mechanisms: Finish Tag Registration and Next Transmitter Polling. Also, to support dynamic membership change (newly joining nodes) in polling, SCF consists of two periods: Join Period and Service Period. Detailed explanation is presented in the following.

A. Finish Tag Registration

When each packet arrives, the packet is stamped with its Finish Tag, calculated by SCFQ as equation 1. Contrary to the centralized architecture shown in Fig. 1, in WLANs, flows are distributed at each node. In equation 1, $F_i^k - 1$, L_i^k, and w_i are local variables which each node knows. On the other hand, $v(t)$ is a shared variable in WLANs. In SCF, the virtual time is maintained at the AP. As described in Section II, the virtual time is updated only when a new packet is transmitted. If a node sends a packet to the AP (uplink), the AP piggybacks the current virtual time in the following ack frame. We call this piggybacked ack frame an Ack+VT compared to a normal ack frame, which is referred to as an Ack. If the AP sends a packet to a node (downlink), the AP piggybacks the current virtual time in the data frame. We call this piggybacked data frame a Data+VT compared to a normal data frame, which is referred to as a Data. The detailed explanation of virtual time is presented in Section III-C. Consequently, each node in the network knows the current virtual time and can calculate each finish tag in a distributed manner.

If a node attempting to transmit a packet has one or more packets in its network interface queue, i.e., backlogged, the finish tag of the next data packet is piggybacked in the current data frame. We call this data frame Data+FT compared to Data.

When the AP receives a Data+FT, it registers the transmitter address and the finish tag in the Data+FT into a Finish Tag Table, which is sorted in an increasing order of the finish tag. Thus, the AP knows which nodes have data to transmit, and the service order among them in the network.

B. Next Transmitter Polling

As described in Section III-A, the AP knows which node should transmit its packet next. When the AP transmits a data frame (Data+VT) or an ack frame (Ack+VT) and its Finish Tag Table is not empty, it piggybacks the next transmitter address (NTA), or the first entry's transmitter address in the Finish Tag Table. We call these piggybacked frames Data+VT+Poll and Ack+VT+Poll, respectively.

When a node receives or overhears a Data+VT+Poll or Ack+VT+Poll, it checks whether it is selected as the next transmitter, i.e, whether the NTA in the frame is equal to its address. If the node is selected, the node transmits the next packet after short interframe space (SIFS) in the case of an Ack+VT+Poll, and after 2 x SIFS + Ack_Transmissioin_Time in the case of a Data+VT+Poll. In the case of an Ack+VT+Poll, since the duration of SIFS is the smallest among all IFSs, the selected node will access the wireless medium without contention. In the case of a Data+VT+Poll, an ack frame of the Data+VT+Poll is followed after SIFS. Thus, in order to transmit the next packet after the followed ack frame, the selected node transmits after 2 x SIFS + Ack_Transmission_Time.

After the transmission of the selected node, the Finish Tag Table in the AP is updated. If the selected node has data, the node transmits a Data+Tag at its reserved transmission time. When the AP receives the Data+Tag, it updates the corresponding entry in Finish Tag Table, whose transmitter address is equal to one in the received frame, to a new finish tag (the finish tag in the Data+Tag). Otherwise, the selected node transmits a Data without a finish tag. When the AP receives the Data, it removes the corresponding entry in the Finish Tag Table.

C. Virtual Time Maintenance

To employ SCFQ, the nodes in the network should know the current virtual time. In SCF, the virtual time is shared by piggybacking as described in Section III.A. In this section, we describe how to update the virtual time in the AP.

In SCFQ, the virtual time is set to the finish tag of a packet which is currently transmitted on the output link. Thus, the virtual time is updated only when a new packet is transmitted. In DFS [3], a node piggybacks the finish tag in a packet, and other nodes set their local virtual times to the finish tag of the packet. Consequently, DFS can emulate centralized SCFQ equivalently in a distributed manner. However, it assumes all the nodes are within the transmission range. This assumption is not applicable to a general situation in WLANs because all the nodes need to communicate only with an AP in order to use the wireless LAN service; all the nodes exist within the transmission rage of an AP. Therefore, DFS may not be effective in general WLAN environment.

On the contrary, in our scheme, an AP announces its virtual time by piggybacking as described in Section III.A. Note that all the nodes in the network can receive a frame sent by an AP. The AP sets the virtual time to the smallest finish tag in the Finish Tag Table and it is announced when the AP transmits an Ack+VT(+Poll) in response to a data frame, or when the AP sends a Data+VT(+Poll). The smallest finish tag in the Finish Tag Table is the next packet's finish tag. Thus, the virtual time in our scheme is advanced faster than the central SCFQ scheme by the amount of a packet. However, since all the nodes calculate their finish tags by the same virtual time, our scheme provides the sufficient fairness among all the nodes.

D. Join Period and Service Period

For a newly joining node to use the WLAN service, it should be able to register with the AP. However, if there are some nodes which are already registered with the AP and always have one or more packets to transmit, the newly joining node cannot register with the AP until the existing nodes finish transmitting all the packets because they are reserved by Data+VT+Poll or Ack+VT+Poll. This may lead to large latency for the newly joining node to start transmitting its packet. In the worst case, starvation can occur. To address this problem, we introduce two periods: Join Period and Service Period.

During the Service Period, only reserved nodes can access the wireless medium. Since the reserved access does not suffer from collisions and the overhead of the access is the minimum, or SIFS duration, the throughput becomes higher as the length of Service Period becomes longer. However, the access delay for a newly joining node increases as the length of the Service Period becomes larger. On the other hand, if the Service Period is too short, the overhead of piggybacking of virtual time, polled address, or a finish tag increases relatively. Thus, considering both throughput and fairness (the access delay), we set the length of Service Period (SP_LEN) as follows

$$SP\_LEN = \min(\max(SP_{min}, \alpha \times N), SP_{max})(\alpha > 1) \quad (2)$$

where SP_{min} is the minimum length of the Service Period, N is the number of the currently registered nodes in the AP, and SP_{max} is the maximum length of the Service Period. SP_{min} and SP_{max} are the system parameters. If newly joining nodes can get registered with the AP within the interval during which all the registered nodes are serviced exactly once, the network is reasonably fair. Since we focus on throughput enhancement further, the interval is relaxed to α times. However, a large access delay may be intolerable to newly joining nodes. Therefore, we set the upper bound of SP_LEN to SP_{max}. On the other hand, if the number of registered nodes is small, SP_LEN may be too short, which leads to relatively large SCF overhead. Thus, we set the lower bound of SP_LEN to SP_{min}.

The AP maintains the number of pollings (SP_PL) during the Service Period. When the AP sends a Data+VT+Poll or an Ack+VT+Poll, it increments SP_PL. If the Finish Tag Table is empty or SP_PL is equal to SP_LEN, the AP uses a Data+VT or an Ack+VT instead of a Data+VT+Poll or an Ack+VT+Poll, and resets SP_PL. If a node receives or overhears a Data+VT or an Ack+VT, it contends for the wireless medium by DCF, which means that Join Period begins. During the Join Period, the AP does not use a Data+VT+Poll or an Ack+VT+Poll for newly joining nodes to register with it. To provide newly joining nodes with higher priority, nodes serviced during the previous Service Period add CW_{min} to their chosen backoff slots. The AP keeps track of the number of packets transmitted (JP_TX) during the Join Period. If JP_TX is equal to the length of Join Period (JP_LEN), the AP resets JP_TX and starts the Service Period by transmitting a Data+VT+Poll or an Ack+VT+Poll.

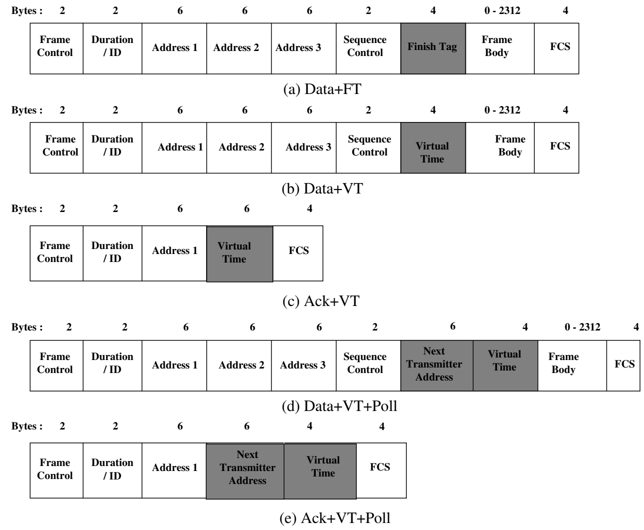

Fig. 2. Frame formats.

E. Frame Formats

In this section, we explain new frame formats for SCF. Figure 2 shows the frame formats. One or two shaded fields are added to the existing frames (Data, Ack) to piggyback SCF-related information: current virtual time, the finish tag of the next data packet, or the address of a polled node. A Data+Tag has a Finish Tag field, where the finish tag of the next data packet is recorded. This frame is used for nodes to register with the AP. An Ack+VT or a Data+VT has a Virtual Time field for the AP to announce the current virtual time to nodes in the network. These frames are transmitted only by the AP during the Join Period. During the Service Period, in order to poll one of registered nodes, the AP transmits an Ack+VT+Poll or a Data+VT+Poll. The address of a polled node is recorded in a Next Transmitter Address field. A Virtual Time field in an Ack+VT+Poll or a Data+VT+Poll is the same as one in an Ack+VT or a Data+VT. These frames are also transmitted only by the AP.

IV. Performance Evaluation

In this section, we compare SCF with DCF, PCF, and FCR [5].

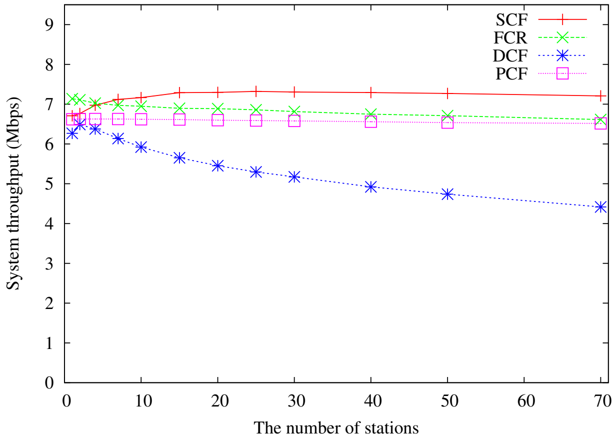

Fig. 3. The system throughput in the saturated network.

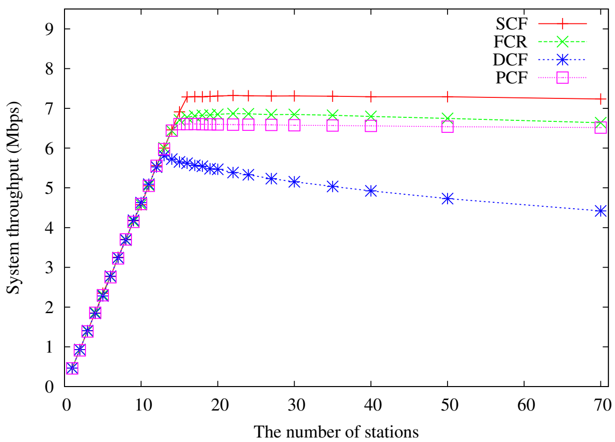

Fig. 4. The system throughput under the increasing network load.

A. Simulation Setup

In order to evaluate SCF, we use ns-2 simulator [8]. Data rate is set to 11 Mbps and Ack rate is set to 2 Mbps. All the nodes exist within the carrier sensing range of each other; there is no hidden node. We do not consider RTS/CTS handshake because it incurs relatively large overhead when high data rate is used with no hidden node situation. The superframe size (Contention Free Period Repetition Interval [7]) of PCF is set to 100 ms. SP_{min}, SP_{max}, JP_LEN, and α are set to 2, 20, 2, and 2 respectively. Throughput is measured at the MAC level excluding MAC header overhead and MAC-layer data payload length is 1500 bytes. Simulations are performed for 100 seconds unless otherwise specified.

B. Simulation Results

To evaluate the effect of contention level on the system throughput, we measure the system throughput varying the number of nodes in a saturated network condition. Throughput of four schemes are plotted in Fig. 3. Since high contention level leads to frequent collisions to contention-based access schemes, the throughput of DCF decreases dramatically (by 31.9%). Although FCR is contention-based, the throughput reduction of FCR is not much (7.3%) due to its enhanced collision resolution algorithm. On the other hand, the throughput of PCF is almost constant because a polling-based MAC does not suffer from collisions. The throughput of SCF increases as the number of nodes increases up to 15, and then is kept almost constantly. This is because the throughput of SCF depends on SP_LEN, which is the function of the number of nodes. Since SCF is polling-based, it also does not suffer from collisions even when the number of node is large. The enhancement of throughput under SCF is 63.2%, 10.6%, 9.0% compared to the throughput under DCF, PCF, and FCR, respectively in high contention level.

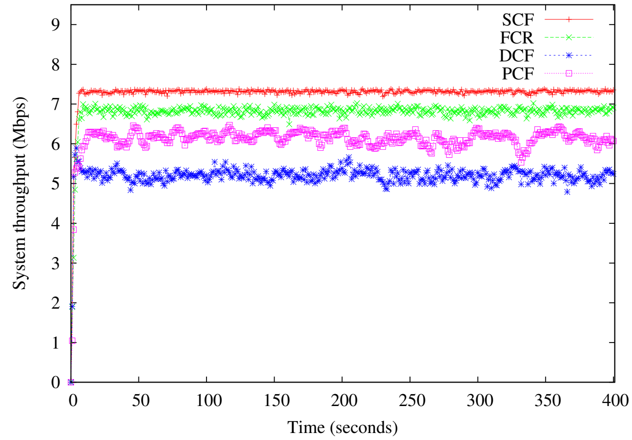

Fig. 5. The system throughput in an ON/OFF traffic scenario.

Fig. 4. shows the system throughput when the packet arrival rate of each node is fixed as 480 Kbps. The arrival rate of data packets in each node is Poisson process that average packet inter-arrival time is 25 ms. The throughput of all the access schemes increases until the network load increase at a certain point. Under DCF, the throughput is reduced severely as the network is overloaded beyond that point. On the other hand, PCF, FCR, and SCF exhibit stable throughput beyond that point. Moreover, the capacity of SCF is highest among all the access schemes and thus the throughput of SCF is largest among all the schemes in overload situations. Overall, SCF extends the capacity of WLANs by 17.0%, 11.1%, and 6.6% compared to that of DCF, PCF, and FCR, respectively.

We also consider ON/OFF traffic scenario where the traffic has two periods: ON (480 Kbps CBR traffic) and OFF (idle). The lengths of ON and OFF are exponentially distributed and the averages of them are five seconds. The number of nodes is 40 and the simulation is conducted for 400 seconds. Throughput with this scenario is shown in Fig. 5. Each point in Fig. 5 represents the throughput for one second interval. The throughput of SCF is also highest in this scenario and more stable than other schemes. This is because SCF is polling-based and only active nodes are polled. FCR and DCF are contention-based schemes. In the case of contention-based schemes, the time-varying membership change of active

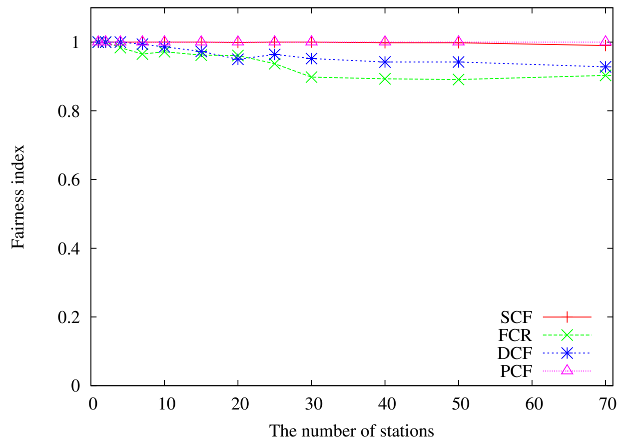

Fig. 6. The short-term fairness.

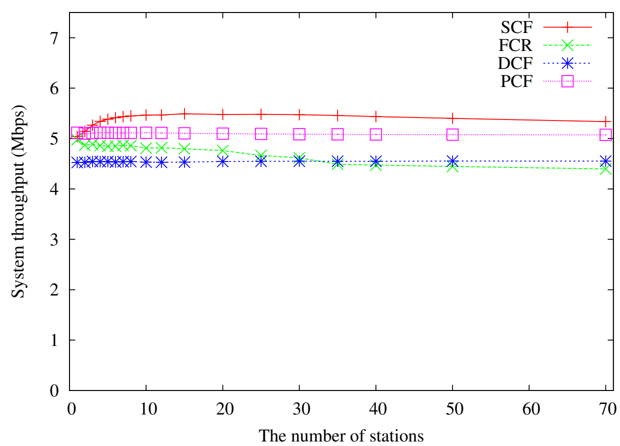

Fig. 7. The system throughput of TCP

nodes affects the system throughput. PCF polls all nodes regardless of whether the node has packets to transmit or not. Thus, the crucial wireless bandwidth is wasted due to polling inactive nodes. Therefore, the system throughput under PCF is fluctuated in ON/OFF traffic.

To evaluate short-term fairness of SCF, we measure the throughput of 10 seconds interval in the saturated network and calculate the fairness using Jain's fairness index [10], which is defined as:

$$F(x) = \left[\sum_{i=1}^{n} x_i\right]^2 / n \sum_{i=1}^{n} x_i^2 \qquad (3)$$

where n is the number of flows and x_i denotes the measured throughput of flow i. As the fairness index approaches to one, it implies that the use of resource by each node is fairer. Figure 6 shows the short-term fairness indices of SCF, FCR, DCF and PCF. It shows that as the number of nodes increases, the fairness indices of DCF and FCR decrease but those of SCF and PCF are maintained near one. This is because the backoff algorithm of DCF and FCR favors the previous successful node while SCF and PCF is polling-based.

Lastly, we measure the throughput of TCP. Contrary to UDP, TCP controls its traffic load by its congestion control algorithm. Figure 7. shows that DCF's throughput degradation is not severe as the number of nodes increases in contrast with the UDP traffic case as shown earlier. The throughput curve of DCF is almost flat. This is because under DCF with TCP traffic the number of average active nodes is about two irrespective of the number of TCP source nodes [9], and the contention among two nodes is too low. Thus, time slots will be wasted by idle backoff slots. On the other hand, SCF polls the only active node without idle backoff slots and hence enhances TCP throughput also.

V. Conclusions

In this paper, we propose a novel medium access control scheme, Scheduling-based Coordination Function (SCF). SCF polls the next transmitter by piggybacking and the selection of the next transmitter is based on the Self-Clocked Fair Queueing scheme in a distributed manner. SCF enhances the system throughput up to 63.2% in high contention level and extends the capacity of WLANs up to 17.0% compared to IEEE 802.11 DCF because it does not suffer from collisions and idle backoff slots. In future work, we have a plan to analyze the effect of the system parameters (SP_{min}, SP_{max}, JP_LEN, and α). Also, we will extend SCF to support QoS by setting w_i properly

Acknowledgement

This work was supported in part by the Brain Korea 21 project of the Ministry of Education, 2006, Korea.

References

[1] Jaehyuk Choi, Joon Yoo, Sunghyun Choi and Chongkwon Kim, "EBA: An Enhancement of the IEEE 802.11 DCF via Distributed Reservation," *IEEE Transaction on Mobile Computing*, vol. 4, no. 2, Mar./Apr. 2005.

[2] S. Jamaloddin Golestani,"A Self-Clocked Fair Queueing Scheme for Broadband Applications," *Proc. IEEE INFOCOM '94*, 1994.

[3] Nitin H. Vaidya, Paramvir Bahl, and Seema Gupta,"Distributed Fair Scheduling in a Wireless LAN," *Proc. ACM MOBICOM '00*, 2000.

[4] Giuseppe Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal of Selected Area on Communications*, vol. 18, no. 3, Mar. 2000.

[5] Younggoo Kwon, Yuguang Fang and Haniph Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," *Proc. IEEE INFOCOM '03*, 2003.

[6] Xue Yang, and Nitin H. Vaidya, "A Wireless MAC Protocol Using Implicit Pipelining," *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, Mar. 2006.

[7] IEEE Std. 802.11-1999, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, Reference number ISO/IEC 8802-11:1999(E), IEEE Std 802.11, 1999 edition, 1999.

[8] "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns/, online link.

[9] Sungwoong Choi, Kihong Park, and Chong-kwon Kim, "On the Performance Characteristics of WLANs: Revisited," *Proc. ACM Sigmetrics '05*, 2005.

[10] Raj Jain, Dah-Ming Chiu and William Hawe, "A Quantative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems," *Digital Equipment Corporation, Technical Report TR-301*, Sep 1984.