Interframe-Space (IFS)-Based Distributed Fair Queuing for Proportional Fairness in IEEE 802.11 WLANs

Jeng Farn Lee, Wanjiun Liao, *Senior Member, IEEE*, and Meng Chang Chen

***Abstract*—In this paper, we propose an interframe-space (IFS)-based distributed-fair-queuing (IDFQ) mechanism to provide proportional fairness service for IEEE 802.11 wireless local-area networks (WLANs). IDFQ is designed to emulate self-clocked fair queuing in a distributed manner. It eliminates the backoff process as implemented in existing work and introduces a new mechanism to assign the IFS value to each station. IDFQ is immune from the implementation problem suffered by existing IFS-based mechanisms and is adaptive to the collision state in the system. Moreover, it can be used to eliminate the performance-anomaly problem of 802.11 medium-access control. The performance of IDFQ is validated by ns-2 simulations. The simulation results show that IDFQ supports fairness service for flows in proportion to their weights and outperforms existing mechanisms in terms of fairness and stability, rendering IDFQ an excellent candidate to provide weighted fairness in IEEE 802.11 WLANs.**

***Index Terms*—Distributed fair queuing (DFS), IEEE 802.11 wireless local-area networks (WLANs).**

I. INTRODUCTION

IEEE 802.11 wireless local-area network (WLAN) with the distributed coordination function (DCF) [1] is the dominant wireless medium to access the Internet. In DCF, mobile stations contend for channel access. All the stations operate independently and share channel bandwidth equally. Since DCF provides best effort service only, the natural evolution is to provide differentiated service for traffic of different demands. There have been many proposals providing service differentiation for wireless LANs but most of them are based on static priority assignment [2]–[4]. Their typical approach is tuning one of the three 802.11 medium-access control (MAC) parameters to enable priority-based service: contention window (CW), backoff interval (BI), and interframe space (IFS). The higher priority traffic is then assigned a smaller parameter value so as to reduce the waiting time for channel access. These fixed priority protocols, however, may starve low-priority traffic. To solve this problem, fair queuing is required [5].

With fair queuing, different flows contending for a shared link can be allocated bandwidth in proportion to their "weights." The typical approach of fair queuing is to emulate the generalized-processor-sharing (GPS) service discipline [6]. GPS is a fluid system in which traffic is infinitely divisible and all traffic streams can receive service simultaneously. In GPS, each flow i is assigned a positive real number ϕ_i, which indicates the weight of flow i for sharing the channel capacity. Let $W_{i,\mathrm{GPS}}(t_1, t_2)$ denote the amount of workload received by flow i in time interval $[t_1, t_2]$. The GPS server guarantees the following equation for each flow, which is continuously backlogged over the interval $[t_1, t_2]$:

$$\frac{W_{i,\mathrm{GPS}}(t_1,t_2)}{W_{j,\mathrm{GPS}}(t_1,t_2)} \geq \frac{\phi_i}{\phi_j}, \qquad i,j = 1,2,\ldots,n. \tag{1}$$

A flow is backlogged if it has packets waiting for transmission in the queue. Let $B(t)$ denote the set of backlogged flows at time t and C be the link capacity. The service rate $r\_\mathrm{GPS}_i(t)$ for each backlogged flow i at time t is then expressed by

$$r\_\mathrm{GPS}_i(t) = \frac{\phi_i}{\sum\limits_{j\in B(t)} \phi_j} C. \tag{2}$$

The challenge of providing fair queuing in WLANs is in that the service provisioning must be fully distributed. This renders existing centralized mechanisms (e.g., the study in [7] for wired networks and [8] for wireless networks) inappropriate. There have been many scheduling disciplines proposed for weighted fairness in the link or MAC layer of IEEE 802.11, including distributed fair queuing (DFS) [9], priority-based fair MAC (P-MAC) [10], and distributed deficit round robin (DDRR) [11]. Their typical approach is applying fair queuing to one of the three 802.11 MAC parameters (i.e., BI, CW, and IFS). In what follows, we first review 802.11 DCF and describe the related work. We then state the problem to solve in this paper.

Manuscript received February 23, 2005; revised December 30, 2005, March 3, 2006, May 16, 2006, and May 29, 2006. This work was supported by the National Science Council (NSC), Taiwan, R.O.C., under the Center Excellence Grants NSC95-2752-E-002-006-PAE and NSC95-2221-E-002-066. The review of this paper was coordinated by Dr. W. Zhuang.

J. F. Lee is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C., and also with the Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan, R.O.C. (e-mail: kunimi@iss.sinica.edu.tw).

W. Liao is with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: wjliao@ntu.edu.tw).

M. C. Chen is with the Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan, R.O.C. (e-mail: mcc@iis.sinica.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2007.895489

A. DCF in 802.11

IEEE 802.11 DCF is based on carrier-sense multiple access with collision avoidance (CSMA/CA), similar to CSMA/CD for Ethernet, to avoid collisions. Each wireless station wishing to transmit monitors the wireless channel. When the channel becomes idle, the station waits for a DCF IFS (DIFS) plus a backoff process before transmitting.

Once the station enters the backoff process, a BI, uniformly distributed in the interval [0, CW], is selected, where the CW falls between the minimum $(\mathrm{CW}_{\min})$ and maximum $(\mathrm{CW}_{\max})$ CWs. The BI is decremented by one after each time slot on an idle medium. When the channel is busy, the backoff process is frozen and resumed after the channel is idle again. Eventually, the backoff counter reaches zero, and the station starts the transmission. A collision occurs if more than one station transmits at the same slot. In this case, the collided stations cannot hear an Acknowledgement (ACK) from their receivers. The CWs of the collided stations are doubled until $\mathrm{CW}_{\max}$ is reached. The CW is reset to $\mathrm{CW}_{\min}$ after each successful transmission.

B. Related Work on Weighted Fair Scheduling in 802.11

1) DFS: DFS is designed to emulate self-clocked fair queuing (SCFQ) [7] in a distributed manner. In SCFQ, a virtual clock is maintained by the central coordinator in the system. Let $\nu(t)$ denote the virtual clock at real time t, P_i^k be the kth packet of flow i, and A_i^k be the real arrival time of P_i^k. Let L_i^k denote the packet size of P_i^k and ϕ_i, the assigned weight of flow i. The start tag S_i^k and the finish tag F_i^k, associated with each packet P_i^k, are given by

$$S_i^k = \max\left\{\nu\left(A_i^k\right), F_i^{k-1}\right\}$$
$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i}. \tag{3}$$

The system virtual clock is initialized to zero and is updated to be the finish tag of the packet being transmitted. In SCFQ, packets are transmitted in increasing order of finish tags. Ties are broken arbitrarily.

To emulate SCFQ in which the frame with the smallest finish tag is transmitted first, each transmitted frame in DFS is stamped with a finish tag, based on which the BI of each competing mobile station is chosen. The BI_i for each wireless station i is expressed by

$$\mathrm{BI}_i = \left\lfloor \left\lfloor \text{Scaling\_Factor} \times \frac{L_i}{\phi_i} \right\rfloor \times \rho \right\rfloor, \tag{4}$$

where L_i is the size of the head-of-line frame, ϕ_i is the assigned weight of station i, and ρ is a random variable uniformly distributed in [0.9, 1.1] for preventing collisions. The ratio between L_i and ϕ_i in (4) is based on the finish tag in (3), and the Scaling_Factor allows the choice of an appropriate scale for the backoff interval in DFS. DFS, however, requires a complicated BI mapping scheme to improve its throughput performance, as the duration of BI for each station is inversely proportional to the weight of the station. This mapping scheme may 1) cause many BI_i to be mapped to the same value, resulting in more collisions, and 2) require BIs to be "recalculated" after each packet transmission for maintaining fairness. Moreover, there is a tradeoff between throughput and fairness in selecting the value of the Scaling_Factor, causing this scheme impractical.

2) P-MAC: In P-MAC, the weighted fairness is achieved by adjusting the CW of each station as follows:

$$\mathrm{CW}_j = \frac{\mathrm{CW}_1 - 1}{\phi_j} + 1, \tag{5}$$

where CW_j and ϕ_j are the CW and the weight for stations of class j, respectively, assuming that the weights of all stations of class 1 are one. CW_1 is the optimal value properly selected to reflect the number of stations contending for the wireless medium, such that P-MAC can maximize the aggregate throughput. However, P-MAC requires that each wireless station keep sensing the channel and monitoring the activities on the wireless medium, such that each station can learn 1) the traffic class to which a successfully transmitted frame belongs and 2) the number of wireless stations in each traffic class. Moreover, the proportional fairness service achieved by P-MAC is through controlling the medium-access probability of each station but without considering the frame size. This may lead to unfairness when the frame size is not fixed.

3) DDRR: DDRR is based on deficit round robin (DRR) [12] to translate user requirements into the IFS parameter of 802.11 MAC. Each station i is allocated a service quantum Q bits every T_i seconds such that Q/T_i is the desired throughput. The deficit counter (DC) of each station i (denoted by DC_i) is increased at a rate of Q bits every T_i seconds, and is decreased by the size of the frame transmitted. The DC value is then mapped to an appropriate IFS value for each station. At time t, the IFS for wireless station i can be expressed by

$$\mathrm{IFS}_i = \mathrm{DIFS} - \alpha\frac{\mathrm{DC}_i(t)}{Q} \times \mathrm{random}(1.0, \beta), \tag{6}$$

where α is a scaling factor to ensure that IFS_i falls between Point coordination function InterFrame Space (PIFS) and DIFS, and β is a value larger than one. In DDRR, unlike DFS and P-MAC, no backoff process is employed. Collisions are avoided by randomizing the IFS values. This is accomplished by multiplying α by a random number between one and β, as in (6).

DDRR, however, has a potential fairness problem. This problem happens when a station, which has accumulated a high DC value (due to transmitting at a rate lower than the desired throughput), starts transmitting at a high rate. This may starve stations because, in DDRR, the DCs of stations continue to accumulate when the stations are inactive, and at the time they start transmissions, their IFS values are determined according to the accumulated DCs without reference to the normalized (global) service amount in the system. Moreover, when the DCs of some stations become zero, since their sending rates exceed their assigned target rates, these stations will share the link capacity equally. This further renders the system unable to provide proportional fairness. Improper settings of parameters may also degrade the performance of DDRR.

Most of all, DDRR has an implementation problem due to its lack of consideration of the physical-layer limitations. With DDRR, each node contends for the channel with a different IFS value bounded by PIFS and DIFS. Since there is only one slot-time difference between PIFS and DIFS, and the slot-time duration in IEEE 802.11 is determined by the physical layer which corresponds to the minimum carrier-sensing time period plus the transmission/receiving turnaround time period, stations transmitting in the same time slot will collide no matter which station starts transmission earlier in that slot.

C. Problem Specification

In this paper, we study fair queuing for IEEE 802.11 WLANs with DCF. In particular, we propose an IFS-based distributed-fair-queuing (IDFQ) mechanism for providing proportional fairness service in conformance with the 802.11 standard. To emulate SCFQ in a distributed manner, each head-of-line frame is stamped with a finish tag. The user requirements are then mapped to the IFS of 802.11 DCF. A smaller finish tag is mapped to a smaller IFS value, and thus, the frame with the smallest tag can be transmitted first. There is no backoff process implemented in IDFQ. Collisions are avoided by an adaptive mechanism plus randomizing the IFS values. Furthermore, it avoids the implementation problem encountered by most IFS-based mechanisms, such as DDRR.

The rest of this paper is organized as follows. In Section II, the proposed mechanism for distributed fair queuing in 802.11 MAC is described. In Section III, the performance of the proposed IDFQ is compared with the existing distributed fair queuing disciplines for 802.11 via the ns-2 simulator. Finally, this paper is concluded in Section IV.

II. IDFQ MECHANISM IN IEEE 802.11 WLAN

In this section, we describe the proposed IDFQ for IEEE 802.11 MAC. With IDFQ, an appropriate IFS value is chosen in proportion to the finish tag of the head-of-line frame. Note that the description of IDFQ in this paper is based on the parameters of 802.11b. It can be easily extended to other 802.11 variations (e.g., 802.11a/g).

In IDFQ, each station i maintains a local virtual clock $\nu_i(t)$ as a function of real time t, and $\nu_i(0) = 0$. Each transmitted frame is stamped with a finish tag. The calculation of the tag is based on (3), except that the tag is calculated when the frame is at the head of line instead of at the arrival time to the system. The virtual clock is updated whenever station i transmits or hears[1] a data or ACK frame with finish tag F at time, say, t and $\nu_i(t) = \max\{\nu_i(t), F\}$. As such, the virtual clock can be ensured to grow monotonically even though the frames with larger finish tags may be transmitted earlier than those with smallest finish tags due to the randomization adopted in IDFQ described next. Note that, since all wireless stations are associated with the same access point, their local virtual clocks are identical. Therefore, the global clock in the system can be maintained.

The finish tag obtained above is then mapped to the IFS parameter of 802.11. A smaller finish tag is mapped to a smaller IFS value such that the frame with the smallest tag is more likely to be transmitted first due to the smallest waiting time. Let F_i be the finish tag of the head-of-line frame for station i, and $L_{\max}$ and $\phi_{\min}$ be the maximum frame size and the minimum weight in the system, respectively. The IFS of station i is expressed by

$$\mathrm{IFS}_i = \lceil \Delta_i \times \beta \rceil \times a \text{ slot time}, \tag{7}$$

where

$$\Delta_i = \begin{cases} \left(\frac{F_i - \nu_i(t)}{\alpha} + 1\right) \times k, & \frac{F_i - \nu_i(t)}{\alpha} < 0 \\ \left(\frac{F_i - \nu_i(t)}{\alpha} \times \text{Scaling\_Factor} + k\right), & \text{otherwise} \end{cases}.$$

The randomization factor β, uniformly distributed between 0.9 and 1.1, is introduced to prevent collisions, and a slot time is 20 μs, which corresponds to a slot time in IEEE 802.11b. $\alpha = L_{\max}/\phi_{\min}$, which is the maximum value of $F_i - \nu_i(t)$; the parameters k and Scaling_Factor are used to choose an appropriate value for the IFS of each station in IDFQ: k is used to scale up the value of $(F_i - \nu_i(t))/\alpha$ when $(F_i - \nu_i(t))/\alpha$ is negative, and the Scaling_Factor is used when $(F_i - \nu_i(t))/\alpha$ is non-negative. The value of Scaling_Factor is initialized to a predefined value and is multiplied by the number of transmission attempts to account for the number of stations involved in collisions. It is reset to the initial value when each frame is transmitted successfully.

The station with the smallest finish tag has the smallest value of $F_i - \nu_i(t)$, because all the stations in the system have the same $\nu_i(t)$, such that the station with the smallest $F_i - \nu_i(t)$ will be transmitted next. We then divide $F_i - \nu_i(t)$ by α, the maximum value of $F_i - \nu_i(t)$, to bound the value of Δ_i. If a station does not hear the data or ACK frame due to the physical-layer errors, it cannot update its virtual clock. This station will be at a disadvantage in channel contention due to its having a smaller system virtual clock. However, once this station hears a data or ACK frame correctly, it can update its virtual clock and be compensated in subsequent contentions since it has a smaller finish tag than other stations in the network. Therefore, failure in updating the system virtual clock due to frame errors may cause short-term unfairness, but long-term fairness can still be guaranteed in IDFQ. Note that, while the frame with the smallest finish tag is usually transmitted first when the network is ready for service, a frame with a larger finish tag may be transmitted earlier as we use a randomization factor (β) to avoid collisions in the distributed environment. When this happens, the value of $F_i - \nu_i(t)$ may be less than zero. We will show below that even so, the value of $|F_i - \nu_i(t)|$ is less than or equal to α. Consequently, $(F_i - \nu_i(t))/\alpha$ stays between -1 and 1; thus the value of IFS is bounded.

Lemma 1: If $F_i - \nu_i(t) \geq 0$, then $F_i - \nu_i(t) \leq L_{\max}/\phi_{\min}$.

Proof: Let t_i be the time when the frame of wireless station i reaches the head of queue, and let L_i be the frame

---

[1] The access point or wireless stations will attach the finish tag of the received frame on the ACK frame to help maintain the system virtual clock.

size. Thus, F_i is equal to $\nu_i(t_i) + L_i/\phi_i$ at t_i according to (3), and

$$F_i - \nu_i(t_i) = \nu_i(t_i) + \frac{L_i}{\phi_i} - \nu_i(t_i) = \frac{L_i}{\phi_i}. \tag{8}$$

Let t' denote the time that the system is ready for transmitting the next frame after t_i. Since the virtual clock grows nondecreasingly, we have $\nu_i(t') \geq \nu_i(t_i)$. Thus, the value of $F_i - \nu_i(t)$ after t_i is given by

$$\begin{aligned} F_i &= \nu_i(t_i) + \frac{L_i}{\phi_i} - \nu_i(t') \\ &\leq \nu_i(t_i) + \frac{L_i}{\phi_i} - \nu_i(t_i) \\ &= \frac{L_i}{\phi_i}. \end{aligned} \tag{9}$$

Consequently, if $F_i - \nu_i(t) \geq 0$, the maximum value of $F_i - \nu_i(t)$ is $L_{\max}/\phi_{\min}$. ■

Lemma 2: If $F_i - \nu_i(t) < 0$, then $\nu_i(t) - F_i \leq L_{\max}/\phi_{\min}$.

Proof: Let P_i be the head-of-line frame of station i. $F_i - \nu_i(t) < 0$, because DIFS uses β to avoid collisions, and thus, the head-of-line frame P_i with finish tag F_i is transmitted after another frame, say, P_j with a larger finish tag F_j. As a result, $\nu_i(t)$, which is the finish tag of station j after P_j is transmitted, is larger than F_i. Therefore, we would like to obtain the maximum value of $F_j - F_i$ that satisfies the following:

$$\begin{cases} \dfrac{F_i - \nu_i(t)}{\alpha} \times \beta_1 > \dfrac{F_j - \nu_i(t)}{\alpha} \times \beta_2 & (10) \\ F_i < F_j. & (11) \end{cases}$$

Since $F_j - F_i$ has the maximum value when $\beta_1 \geq \beta_2$ and β_1 has a positive value from (10), we have

$$\begin{aligned} &\frac{F_i - \nu_i(t)}{\alpha} \times \beta_1 > \frac{F_j - \nu_i(t)}{\alpha} \times \beta_2 \\ &\qquad => (F_i - \nu_i(t)) > \frac{\beta_2}{\beta_1}(F_j - \nu_i(t)). \end{aligned} \tag{12}$$

From (11) and (12), we have

$$F_j - \nu_i(t) > (F_i - \nu_i(t)) > \frac{\beta_2}{\beta_1}(F_j - \nu_i(t)). \tag{13}$$

Subtracting $F_j - \nu_i(t)$ on the both sides of (13) and according to Lemma 1, we have

$$0 < F_j - F_i < \frac{\beta_1 - \beta_2}{\beta_1}(F_j - \nu_i(t)) \leq \frac{\beta_1 - \beta_2}{\beta_1}\frac{L_{\max}}{\phi_{\min}} \leq \frac{L_{\max}}{\phi_{\min}}. \tag{14}$$

Consequently, if $F_i - \nu_i(t) < 0$, the maximum value of $\nu_i(t) - F_i$ is $L_{\max}/\phi_{\min}$. ■

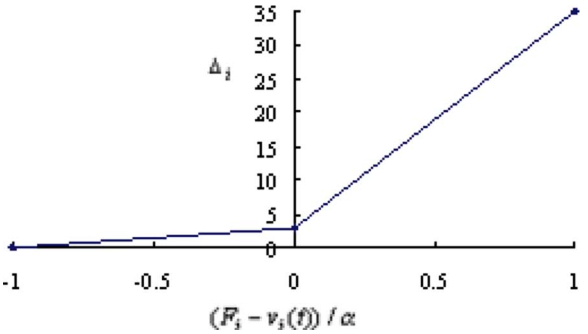

Fig. 1. Mapping function Δ_i.

Theorem 1: The value of $(F_i - \nu_i(t))/\alpha$ stays between -1 and 1.

Proof: The value of $|F_i - \nu_i(t)|$ is less than or equal to $L_{\max}/\phi_{\min}$ according to Lemmas 1 and 2. Thus, $(F_i - \nu_i(t))/\alpha$ is between -1 and 1. ■

Since the value of $(F_i - \nu_i(t))/\alpha$ is a real number between -1 and 1, it is impractical to use this small range multiplied by one slot time to determine the IFS values of different stations in 802.11 WLANs, as many collisions would occur. Thus, we need to enlarge this range to a reasonable interval. Since the value of $(F_i - \nu_i(t))/\alpha$ is between -1 and 1, to improve the system throughput of IDFQ, we do not use the same scaling value for both cases of $(F_i - \nu_i(t))/\alpha < 0$ and $(F_i - \nu_i(t))/\alpha \geq 0$ because the number of stations with negative $(F_i - \nu_i(t))/\alpha$ should remain small in the system and most of the stations are supposed to have a positive value of $(F_i - \nu_i(t))/\alpha$. When $(F_i - \nu_i(t))/\alpha < 0$, the preferable setting of the scaling slope k is a small integer, for example, less than five; when $(F_i - \nu_i(t))/\alpha \geq 0$, the setting of the Scaling_Factor will significantly influence the throughput performance of IDFQ. A larger Scaling_Factor value will lead to a larger IFS, for which stations must wait before each transmission starts, but with a reduced collision probability. A collision occurs if the difference between the smallest IFS and the second smallest IFS is less than a slot time. In IDFQ, we are only concerned with the IFS value of the station that will transmit next, rather than those of all stations. The station that will transmit next will have the smallest value of $(F_i - \nu_i(t))/\alpha$, and the value is very close to zero in general. Therefore, a larger Scaling_Factor value, such as 200, is suggested in IDFQ. However, if the frame size in the network is fixed and all stations are assigned the same weight, a smaller Scaling_Factor, such as 32, is preferred. Fig. 1 shows the mapping function Δ_i, where k is three and the Scaling_Factor is 32. The value of Δ_i when $(F_i - \nu_i(t))/\alpha = 1$ is equal to Scaling_Factor $+ k$.

Note that the mapping function of Fig. 1 cannot be used directly in the BI-based mechanisms, because the operations of the BI and IFS are different. In DCF, the BI value is frozen when the channel is busy, and the backoff entity resumes its countdown of the BI value when the channel is idle again. A frame with a large BI may no longer have a large BI after one or two transmissions. Thus, it is not a good idea to use the BI value as the waiting time to emulate SCFQ in our algorithm.

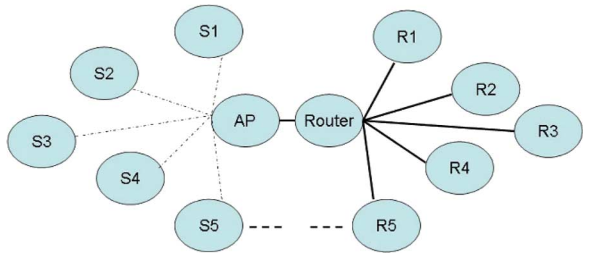

Fig. 2. Simulation topology.

III. PERFORMANCE EVALUATION

In this section, we conduct simulations based on ns-2 to evaluate the performance of four different fair-queuing mechanisms in the MAC layer of IEEE 802.11 WLANs, including DDRR, DFS, P-MAC, and IDFQ. Fig. 2 gives the simulation topology. The wireless link rate is 11 Mb/s, and the link capacity between the AP and the router is set to 100 Mb/s. Each wireless sender generates constant bit rate User Datagram Protocol (UDP) traffic at a rate of 8 Mb/s to a receiver in the wired network. The packet size including the IP header is uniformly distributed between 500 and 2304 B in length and the sampling interval is fixed at 0.5 s. We assume that the frames are error-free in our simulations.

The parameter setting of each mechanism is described as follows. The values of the Scaling_Factor and Threshold in DFS are set to 0.02 and 80, respectively, the same setting as in [9]. The following square-root mapping scheme proposed in the study in [9] is used in DFS in the simulation.

$$\mathrm{BI}_i = \psi(\Delta) = \begin{cases} \Delta, & \text{if } \Delta < \text{Threshold} \\ \lfloor \sqrt{\text{Threshold} \times \Delta} \rfloor, & \text{otherwise} \end{cases} \tag{15}$$

where Δ is the BI value obtained in (4), and the Threshold is a constant parameter. The value of β in DDRR is set to 1.9. Note that, in order to show the fairness problem of DDRR, we assume that the transmitted signals can be heard by all nodes with no propagation delay, no turnaround (from sense to transmit) time, and no sensing time in the implementation of DDRR. However, these assumptions may not hold in practice. Thus, the result can be regarded as the best performance achievable by DDRR. The Scaling_Factor and k in the mapping function of IDFQ [i.e., (7)] are set to 200 and 3, respectively.

We compare the fairness and stability of each mechanism. The fairness is measured with respect to the fairness index FI defined in [10]. Let R_i and ϕ_i be the average throughput and the weight of flow i, respectively. The fairness index is then defined as follows:

$$\mathrm{FI} = \frac{\mu\left(\frac{R_i}{\phi_i}\right)}{\mu\left(\frac{R_i}{\phi_i}\right) + \sigma\left(\frac{R_i}{\phi_i}\right)}, \tag{16}$$

where μ and σ are the mean and the standard deviation of R_i/ϕ_i over all data flows. The fairness index is a real value between zero and one. The closer to one the fairness index, the fairer. The stability is referred to as the fluctuation degree of the throughput of each flow in each scheduling discipline.

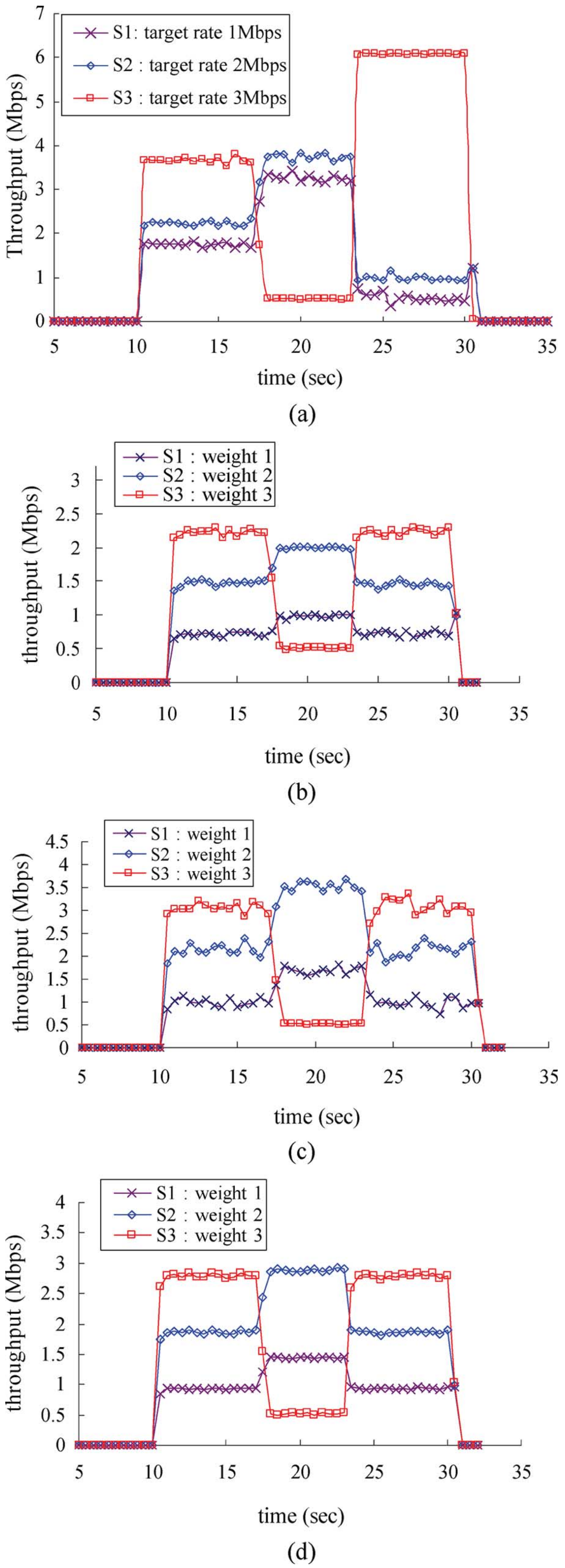

Fig. 3. Throughputs when the sending rates are changed. (a) DDRR. (b) DFS. (c) P-MAC. (d) IDFQ.

We first compare the fairness for different mechanisms with respect to network-load variations. We consider three stations sending packets in the system: S1, S2, and S3. The target rates of these three stations in DDRR are set to 1–3 Mb/s, corresponding to S1–S3, respectively. The assigned weights of $\mathrm{S1}:\mathrm{S2}:\mathrm{S3}$ in the rest of the mechanisms are set to $1:2:3$. Fig. 3 plots the simulation results in the case that S3 changes its sending rate from 8 to 0.5 Mb/s at time 17 and from 0.5 to 8 Mb/s at time 23. Fig. 3(a) shows that the stations S1 and S2 fail to receive their desired service shares after time 23 when DDRR is used. This is because, with DDRR, each node maintains the DC value locally, such that the DC value of S3 will be accumulated when S3 sends at a lower rate. The fairness indexes of DDRR, DFS, P-MAC, and IDFQ during time 10–17 are 0.80, 0.97, 0.96, and 0.99, respectively. We observe that DDRR has the worst fairness index during time 10–17, because all stations share the link capacity equally when their DCs become zero. Fig. 3(b) and (c) shows that DFS and P-MAC can provide fair service only to a limited extent (with fluctuations in each curve). Fig. 3(d) shows that, with IDFQ, all stations share the link capacity according to their weights and the curves are much smoother. Thus, it can achieve excellent fairness and stability.

Next, we compare proportional fairness provided by the different mechanisms. In this simulation, there are five wireless senders with weights $1:2:2:4:4$, corresponding to S1–S5, respectively. Note that DDRR is not considered in the following simulations due to its implementation problem. Fig. 4 shows the throughput and the fairness of each flow for each scheduling discipline. We observe that all the three mechanisms provide proportional fairness for flows with different weights. Of these three mechanisms, IDFQ provides the best stability and fairness (i.e., 0.999) due to its perfect emulation of SCFQ in distributed environments.

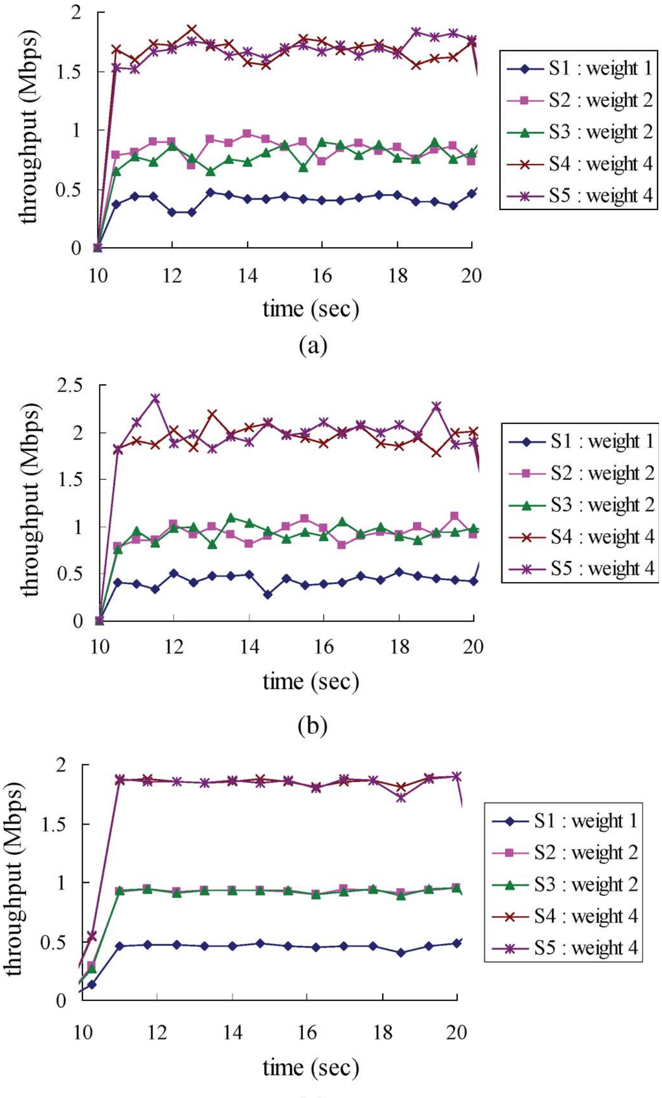

Fig. 4. Throughputs and fairness indices of three schemes with respect to the fairness performance. (a) DFS $(\mathrm{FI} = 0.96)$. (b) P-MAC $(\mathrm{FI} = 0.94)$. (c) IDFQ $(\mathrm{FI} = 0.99)$.

In P-MAC, a CW-based mechanism, a random BI value is selected uniformly from the interval [0, CW] so as to avoid collisions. This randomization, however, may result in a larger variance in the values of BIs and, thus, larger throughput fluctuation. In DFS, a BI-based mechanism, the randomization factor is usually very small (for example, [0.9, 1.1]), thus leading to a smaller fluctuation in throughput performance. IDFQ, an IFS-based mechanism, also uses a small randomization factor as in DFS to avoid collisions. However, compared with DFS, IDFQ provides better fairness and stability, which is explained as follows. Although both DFS and IDFQ employ a randomization factor to avoid collisions, the randomization factors will cause different effects on these two mechanisms. Taking a frame size of 1400 B (i.e., the mean size of frames in our simulations) and a weight of 0.1, as an example, the BI value for a frame in DFS, according to (4), ranges from 252 to 308, corresponding to a difference of 56, which will cause throughput fluctuation. Even if the mapping function (15) is adopted to improve the DFS's throughput performance, which results in a difference of 14 (i.e., $156 - 142$), the small range of possible BIs will still introduce throughput fluctuation, as the transmission time of the frame is only 101 (slot time). In addition, the range of possible BIs are affected by the frame size and the assigned weight of that station. In summary, any range of possible BIs in DFS will introduce throughput fluctuation, and may cause a change in the transmission ordering of frames in SCFQ, resulting in short-term unfairness. In contrast, in IDFQ, although frames with higher finish tags have larger values of Δ_i, the value Δ_i is bounded by Scaling_Factor $+\ k$, regardless of the frame size the weight. For example, the value of Δ_i in IDFQ is bounded by 203 (i.e., $200 + 3$); however, the maximum value of BI in (4) of DFS is $2304^*0.02/0.01 = 4600$, if the weight of a station is 0.01, as in our simulations. Moreover, a wider range of the IFS values for frames with higher finish tags introduced by the randomization factor will not have negative impacts on our mechanism, since these frames are not the candidates to be transmitted next. The next transmitted frame will have a small value of Δ_i and, thus, have a small possible range of the IFS values. Therefore, the randomization process has less effect on the throughput and fairness fluctuation in IDFQ, i.e., IDFQ is relatively stable and fair.

We then show the scalability of each mechanism with 20 wireless stations in the system. The numbers of stations assigned weights 1, 2, and 4 are 8, 8, and 4, respectively.

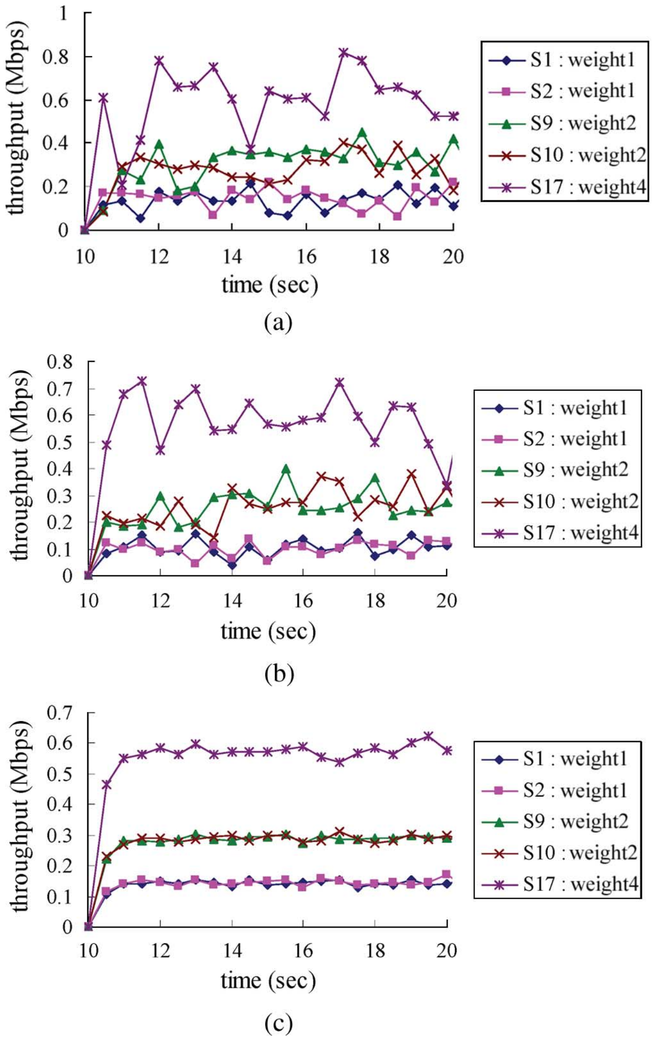

Fig. 5. Throughputs and fairness indices of three schemes with respect to the scalability performance. (a) DFS (FI $= 0.93$). (b) P-MAC (FI $= 0.90$). (c) IDFQ (FI $= 0.99$).

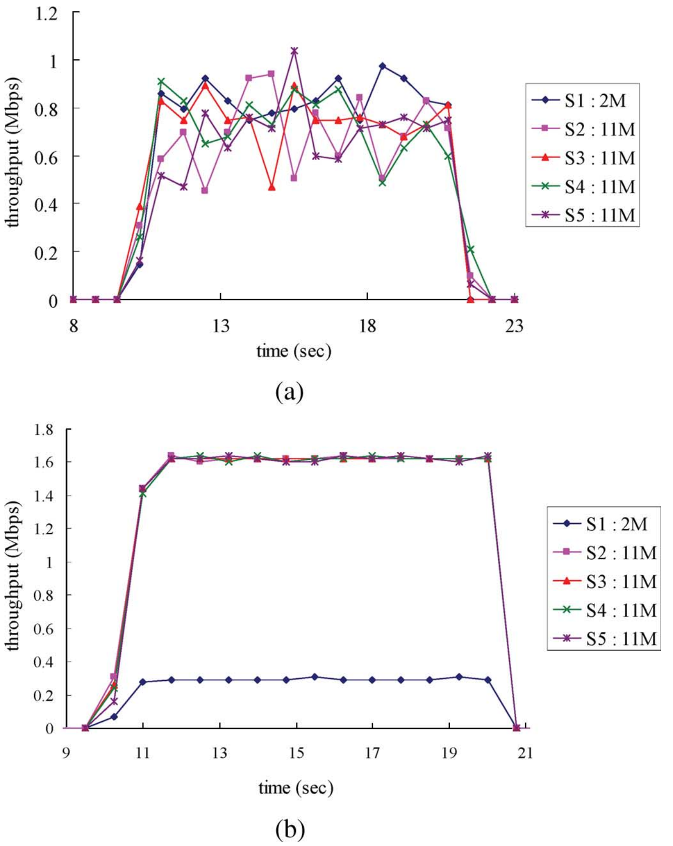

Fig. 6. Throughput of flows when sending rates of stations are different. (a) 802.11b DCF. (b) IDFQ.

Fig. 5 shows the throughputs of the 1st, the 2nd, the 9th, the 10th, and the 17th senders for each scheduling discipline. IDFQ provides the best fairness and stability for flows with both the identical and different weight assignments. We observe that the aggregate throughput of IDFQ is not worse than those of DFS and P-MAC, irrespective of the number of wireless stations in the network.

Finally, we study the performance anomaly in 802.11 reported by Heusse *et al.* [13]. The performance anomaly occurs when some wireless stations operate at lower rates than others due to signal fading and interference. When such anomaly happens, the throughputs of all stations are degraded considerably. For example, if a station transmits at 2 Mb/s, the throughput of all stations in the system will degrade to 2 Mb/s. This anomaly results from the fact that 802.11 DCF guarantees an equal-access probability for all stations in the long term. Consequently, when one station operates at a lower bit rate, it tends to capture the channel for a longer time. This behavior causes stations with higher rates to transmit at a lower rate, which is undesirable in practice. This anomaly can be remedied if we provide a weighed fair service discipline in the system and set the weight of wireless stations according to their bit rates. Fig. 6 plots the simulation result with the setting that S1 operates at a rate of 2 Mb/s and the others at a rate of 11 Mb/s. We observe that the original 802.11 DCF does suffer from the performance anomaly, while the proposed IDFQ is immune to this problem and can still ensure the demand of each flow. Note that all the service disciplines that can provide proportional bandwidth allocation among wireless stations, such as DFS and P-MAC, can avoid this performance anomaly. Due to space limitations, we only show the result of IDFQ due to its best performance.

IV. CONCLUSION

In this paper, we have proposed a mechanism for proportional fairness in IEEE 802.11 WLANs. Unlike existing work based on the BI values for collision resolutions, the proposed mechanism chooses appropriate IFS values for flows with different weights and applies an adaptive mechanism plus some randomization to avoid collisions. Unlike existing IFS-based mechanisms, such as DDRR, which suffer from implementation problems, the proposed IDFQ mechanism takes the physical characteristics of wireless channel into consideration and designs a mapping function to improve network throughput. We compare IDFQ with several existing distributed service disciplines through simulations. The results show that the proposed mechanism outperforms other service disciplines significantly in terms of fairness, stability, scalability, and aggregate throughput.

In this paper, we consider only single-hop IEEE 802.11 wireless networks. In the future, we will extend our mechanism to provide end-to-end proportional fairness to flows in multihop wireless network based on IEEE 802.11 MAC. We will also conduct theoretical analysis for the proposed IDFQ mechanism with respect to the throughput performance.

REFERENCES

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11, Aug. 1999.

[2] I. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM*, 2001, pp. 209–218.

[3] J. Deng and R. S. Chang, "A priority scheme for IEEE 802.11 DCF access method," *IEICE Trans. Commun.*, vol. E82-B, no. 1, pp. 96–102, 1999.

[4] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed multi-hop scheduling and medium access with delay and throughput constraints," in *Proc. ACM Mobicom*, 2001, pp. 200–209.

[5] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, "Distributed mechanisms for quality of service in wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 26–34, Jun. 2003.

[6] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[7] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *Proc. IEEE INFOCOM*, Toronto, CA, Jun. 1994, pp. 636–646.

[8] S. Lu, T. Nandagopal, and V. Bharghavan, "A wireless fair service algorithm for packet cellular networks," in *Proc. ACM Mobicom*, 1998, pp. 10–20.

[9] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in wireless LAN," in *Proc. ACM Mobicom*, 2000, pp. 167–178.

[10] D. Qiao and K. G. Shin, "Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF," in *Proc. ACM IWQoS*, 2002, pp. 227–236.

[11] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy, "Starvation prevention and quality of service in wireless LANs," in *Proc. IEEE WPMC*, 2002, pp. 1078–1082.

[12] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Trans. Netw.*, vol. 4, no. 3, pp. 375–385, Jun. 1996.

[13] M. Heusse, F. Rousseau, and G. Berger-Sabbatel, "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM*, 2003, pp. 836–843.

Jeng Farn Lee received the B.S. and M.S. degrees from the Department of Information Management, National Taiwan University, Taipei, Taiwan, R.O.C., in 1998 and 2000, respectively, where since 2003, he has been working toward the Ph.D. degree at the Department of Electrical Engineering.

Since 2001, he has been with the Institute of Information Science, Academia Sinica, Taipei, as a Research Assistant. His current research interests include QoS networking, scheduling, and wireless-access network.

Wanjiun Liao (S'96–M'97–SM'06) received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, in 1997.

In 1997, she was with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, as an Assistant Professor, and since August 2005, as a Full Professor. Her research interests include wireless, multimedia, and broadband-access networks.

Dr. Liao is currently an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE TRANSACTIONS ON MULTIMEDIA. She served as the Technical Program Committee chair/cochair of many international conferences, including the IEEE INFOCOM 2004 as a Tutorial Cochair, IEEE Globecom 2005 Symposium on Autonomous Networks as the Technical Program Vice Chair, and IEEE Globecom 2007 General Symposium as a Technical Program Cochair. She has received many research awards. Papers she coauthored with her students received the Best Student Paper Award at the First IEEE International Conference on Multimedia and Expo (ICME) in 2000 and the Best Paper Award at the First IEEE International Conference on Communications, Circuits, and Systems (ICCCAS) in 2002. She also received the K. T. Li Young Researcher Award from the Association for Computing Machinery in 2003 and a Distinguished Research Award from the National Science Council, Taiwan, in 2006.

Meng Chang Chen received the B.S. and M.S. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1979 and 1981, respectively, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1989.

In 1989, he was with AT&T Bell Laboratories as a member of the Technical Staff and led several projects in the area of data quality of distributed databases for mission critical systems. From 1992 to 1993, he was an Associate Professor in the Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan. Since 1993, he has been with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where since July 1996, he has held Associate Research Fellowship and from August 1999 to July 2002, he assumed the responsibility of Deputy Director. His current research interests include QoS networking, wireless network, multimedia systems and transmissions, operating systems, and data and knowledge engineering.