Inter-Frame Space (IFS)-based Distributed Fair Queuing for Proportional Fairness in IEEE 802.11 WLANs

Jeng Farn Lee, Wanjiun Liao, and Meng Chang Chen

***Abstract*- In this paper, we study fair queuing in the IEEE 802.11 Distributed Coordination Function (DCF). We propose an IFS-based Distributed Fair Queuing (IDFQ) mechanism to provide proportional fairness service for IEEE 802.11 WLANs. IDFQ is designed to emulate Self-Clocked Fair Queuing (SCFQ) in a distributed manner. It eliminates the backoff process as implemented in existing work and introduces a new mechanism to assign the inter-frame space 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 with 802.11 MAC. The performance of IDFQ is validated by ns-2 simulations. 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.**

Keywords: *distributed fair queuing, IEEE 802.11 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 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 are based on static priority assignment [2-4]. Their typical approach is tuning one of the three 802.11 MAC parameters to enable priority-based service: Contention Window (CW), Backoff Interval (BI), and Inter-Frame Space (IFS). The higher priority traffic is then assigned a smaller parameter value so as to reduce the waiting time in channel access. These fixed priority protocols, however, may starve low priority traffic. To solve this problem, fair queuing is required [5].

Jeng Farn Lee and Wanjiun Liao are with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan.
Meng Chang Chen, and also Jeng Farn Lee, are with the Institute of Information Science, Academia Sinica, Taipei, Taiwan.

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 f_i, which indicates the weight of flow i for sharing the channel capacity. Let $W_{i,GPS}(t_1,t_2)$ denote the amount of workload received by flow i in time interval $[t_1,t_2]$. A GPS server guarantees the following equation for each flow which is continuously backlogged over the interval $[t_1,t_2]$,

$$\frac{W_{i,GPS}(t_1,t_2)}{W_{j,GPS}(t_1,t_2)} \geq \frac{f_i}{f_j} \quad i,j=1, 2, \ldots, \text{n}. \qquad (1)$$

A flow is *backlogged* if it has packets waiting for transmission in the queue or its packet is being transmitted. Let $B(t)$ denote the set of the backlogged flows at time t, and C be the link capacity. The service rate $r\_GPS_i(t)$ of each backlogged flow i at time t is then expressed by

$$r\_GPS_i(t) = \frac{f_i}{\sum_{j \in B(t)} f_j} C . \qquad (2)$$

The challenge of providing fair queuing in WLANs is in that service provisioning must be fully distributed. This renders existing centralized mechanisms (e.g., [7] for wired networks and [8] for wireless networks in which base stations are involved) 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 Medium Access Control (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.

A. Distributed Coordination Function (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 but introducing the "inter frame space" (IFS) and the backoff interval 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 a transmission.

Once the station enters the backoff process, a Backoff Interval (BI), uniformly distributed in the interval [0, CW] is selected, where CW (i.e., Contention Window) falls between the minimum ($CW_{\min}$) and maximum ($CW_{\max}$) contention windows. The backoff interval 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 a transmission. A collision occurs if more than one station transmits at the same slot. In this case, the collided stations cannot hear an ACK from their receivers. The CWs of the collided stations are doubled until $CW_{\max}$ is reached. CW is reset to $CW_{\min}$ 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 *v(t)* denote the virtual clock at real time *t*, P_i^k be the k^{th} 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 f_i the assigned weight of flow *i*. A start tag S_i^k and a finish tag F_i^k are associated with each packet P_i^k, given by

$$\begin{aligned} S_i^k &= \max\{v(A_i^k), F_i^{k-1}\} \\ F_i^k &= S_i^k + \frac{L_i^k}{f_i} \end{aligned} \tag{3}$$

The system virtual clock is initialized to 0, 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 backoff interval BI_i for each wireless station *i* is expressed by

$$BI_i = \left\lfloor \left\lfloor Scaling\_Factor \times \frac{L_i}{f_i} \right\rfloor \times r \right\rfloor, \quad (4),$$

where L_i is the size of the head of line frame, f_i is the assigned weight of station *i*, and r is a random variable uniformly distributed in [0.9,1.1] for preventing collisions. The ratio between L_i and f_i in (4) is based on the finish tag in (3), and the *Scaling_Factor* allows the choice of an appropriate scale for the virtual time in DFS. DFS, however, requires a complicated backoff interval mapping scheme to improve its throughput performance as the duration of the backoff interval is inversely proportional to the weight of a station. This mapping scheme may (*i*) cause many BI_i to be mapped to the same value, resulting in more collisions, and (*ii*) require backoff intervals to be "recalculated" after each packet transmission for maintaining fairness. Moreover, there is a trade-off 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 contention window (CW) of each station as follows:

$$CW_j = \frac{CW_1 - 1}{f_j} + 1, \tag{5}$$

where CW_j and f_j are the CW and the weight of stations of class *j*, respectively, assuming the weights of all stations of class 1 are 1. CW_1 is an 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 keeps sensing the channel and monitoring the activities on the wireless medium such that each station can learn (*i*) the traffic class to which a successfully-transmitted frame belongs and (*ii*) 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 when a frame is transmitted. The deficit counter value is then mapped to an appropriate IFS value at each station. At time *t*, the IFS for wireless station *i* can be expressed by:

$$IFS_i = DIFS - a\,\frac{DC_i(t)}{Q} \times random(1.0, b), \tag{6}$$

where a is a scaling factor to ensure IFS_i falling between PIFS and DIFS, and b is a value larger than 1.

In DDRR, unlike DFS and P-MAC, no backoff algorithm is employed. Collisions are avoided by randomizing the IFS values. This is accomplished by multiplying a by a random number between 1 and b 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 transmission at a high rate. This may cause starvation to other stations because, in DDRR, the deficit counters of stations continue to accumulate when stations are inactive, and at the time they start transmissions, their IFS values are determined according to the accumulated deficit counters, without reference to the normalized (global) service amount in the system. Moreover, when the deficit counters of some stations become zero due to their sending rates exceeding 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 a PIFS and a 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 at the same time slot will collide no matter which station starts transmission earlier at 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 mechanism implemented in IDFQ. The collisions are avoided by an adaptive mechanism plus randomizing each IFS value. Furthermore, it avoids the implementation problem encountered by most IFS-based mechanisms such as DDRR.

The rest of the paper is organized as follows. In Sec. II, the proposed mechanism for distributed fair queuing in 802.11 MAC is described. In Sec. 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, the paper is concluded in Sec. IV.

II. IDFQ: AN IFS-BASED DISTRIBUTED FAIR QUEUEING MECHANISM IN IEEE 802.11 WLAN

In this section, we describe the proposed IFS-based Distributed Fair Queuing (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 $v_i(t)$ as a function of real time t, and $v_i(0) = 0$. 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[2] a data or ACK frame with finish tag F, say, at time t, and $v_i(t) = \max\{v_i(t), F\}$. As such, the virtual clock can be ensured to grow monotonically because frames with larger finish tags may be transmitted earlier than those with smallest finish tags due to the randomization adopted in IDFQ described below. Note that since all wireless stations are associated with the same access point, their local virtual clocks are equal. 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, $L_{\max}$ and $f_{\min}$ be the maximum frame size and the minimum weight in the system, respectively. The IFS of station i is expressed by

$$IFS_i = \lceil \Delta_i \times b \rceil \times a\text{slottime} \tag{7}$$

where

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

the randomization factor b, uniformly distributed between 0.9 and 1.1, is introduced to prevent collisions, and *a*slottime is 20 *us*, which corresponds to a slot time in IEEE 802.11b. a = $L_{\max} / f_{\min}$, which is the maximum value of $F_i - v_i(t)$; the parameters k and *Scaling_Factor* are used to choose an appropriate value for the inter-frame space of each station in IDFQ: k is used to scale up the value of $(F_i - v_i(t))/a$ when $(F_i - v_i(t))/a$ is negative, and *Scaling_Factor* is used

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

when $(F_i - v_i(t))/a$ is non-negative. The value of *Scaling_Factor* is initialized to a pre-defined 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 will have the smallest value of $F_i - v_i(t)$ because all stations in the system have the same $v_i(t)$ such that the station with the smallest $F_i - v_i(t)$ will be transmitted next. We then divide $F_i - v_i(t)$ by a, the maximum value of $F_i - v_i(t)$, to bound the value of Δ_i. If a station does not hear the data or ACK frame due to 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 when the network is ready for service, a frame with a larger finish tag may be transmitted earlier as we use a randomization factor ß to avoid collisions in the distributed environment. When this happens, the value of $F_i - v_i(t)$ may be less than 0. We will show below that even so, the value of $|F_i - v_i(t)|$ is less than or equal to a . Consequently, $(F_i - v_i(t))/a$ stays between -1 and 1; thus the value of IFS is bounded.

Lemma 1: If $F_i - v_i(t) \ge 0$, then $F_i - v_i(t) \le \frac{L_{\max}}{f_{\min}}$.

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 size. Thus, F_i is equal to $v_i(t_i) + L_i / f_i$ at t_i according to (3), and

$$F_i - v_i(t_i) = v_i(t_i) + \frac{L_i}{f_i} - v_i(t_i) = \frac{L_i}{f_i} \qquad (8)$$

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

$$\begin{aligned} F_i &= v_i(t_i) + \frac{L_i}{f_i} - v_i(t') \\ &\le v_i(t_i) + \frac{L_i}{f_i} - v_i(t_i) \qquad (9) \\ &= \frac{L_i}{f_i}. \end{aligned}$$

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

Lemma 2: If $F_i - v_i(t) < 0$, then $v_i(t) - F_i \le \frac{L_{\max}}{f_{\min}}$.

Proof: Let P_i be the head of line frame of station *i*. $F_i - v_i(t) < 0$ because DIFS uses b 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, $v_i(t)$, which is the finish tag of station j after P_j is transmitted, will be lager than F_i. Therefore, we would like to obtain the maximum value of $F_j - F_i$ that satisfies (10) and (11).

$$\begin{cases} \frac{F_i - v_i(t)}{a} \times b_1 > \frac{F_j - v_i(t)}{a} \times b_2 & (10) \\ F_i < F_j & (11) \end{cases}$$

Since $F_j - F_i$ has the maximum value when $b_1 \ge b_2$, from (10), we have

$$\begin{aligned} &\frac{F_i - v_i(t)}{a} \times b_1 > \frac{F_j - v_i(t)}{a} \times b_2 \\ &=> (F_i - v_i(t)) > \frac{b_2}{b_1}(F_j - v_i(t)). \qquad (12) \end{aligned}$$

From (11) and (12), we have

$$F_j - v_i(t) > (F_i - v_i(t)) > \frac{b_2}{b_1}(F_j - v_i(t)). \qquad (13)$$

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

$$\begin{aligned} 0 < F_j - F_i &< \frac{b_1 - b_2}{b_1}(F_j - v_i(t)) \\ &\le \frac{b_1 - b_2}{b_1} \frac{L_{\max}}{f_{\min}} \le \frac{L_{\max}}{f_{\min}} \end{aligned} \qquad (14)$$

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

Theorem 1: The value of $(F_i - v_i(t))/a$ stays between -1 and 1.

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

Since the value of $(F_i - v_i(t))/a$ 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 may occur. Thus, we need to enlarge this range to a reasonable interval. Since the value of

$(F_i - v_i(t))/a$ is between -1 and 1, to improve the system throughput of IDFQ, we do not use the same scaling value for both cases of $(F_i - v_i(t))/a < 0$ and $(F_i - v_i(t))/a \geq 0$ because the number of stations with negative $(F_i - v_i(t))/a$ should remain small in the system and most of the stations are supposed to have positive $(F_i - v_i(t))/a$. When $(F_i - v_i(t))/a < 0$, the preferable setting of the scaling slope *k* is a small integer, say less than 5; when $(F_i - v_i(t))/a \geq 0$, the setting of *Scaling_Factor* will significantly influence the throughput performance of IDFQ. A larger *Scaling_Factor* value will lead to a larger inter-frame space for which stations must wait before each transmission, but with 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 inter-frame space of the station that will transmit next, rather than those of all stations. The station that will transmit next will have the smallest $(F_i - v_i(t))/a$ value, and the value is very close to 0 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 3, and *Scaling_Factor* is 32. The value of Δ_i when $(F_i - v_i(t))/a = 1$ is equal to *Scaling_Factor*+3.

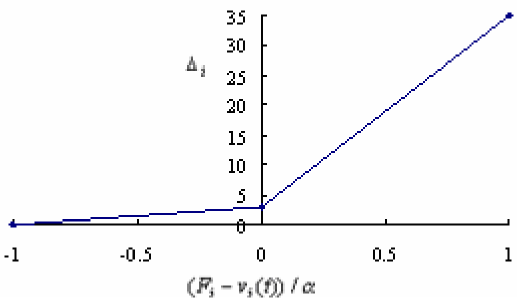


Figure 1. The mapping function Δ_i.

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

III. PERFORMANCE EVALUATION

In this section, we conduct simulations based on ns-2 [13] 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 11Mbps, and the link capacity between the AP and the router is set to 100Mbps. Each wireless sender generates CBR UDP traffic at a rate of 8Mbps to a receiver in the wired network. The packet including the IP header is uniformly distributed between 500 and 2304 bytes in length and the sampling interval is fixed at 0.5 seconds. We assume that frames are error-free in our simulations.

The parameter setting of each mechanism is described as follows. The values of *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 [9] is used in DFS in the simulation.

$$BI_i = y(\Delta) = \begin{cases} \Delta & \text{if } \Delta < Threshold \\ \left\lfloor \sqrt{Threshold \times \Delta} \right\rfloor & \text{otherwise} \end{cases} \qquad (15)$$

where Δ is the backoff interval obtained in (4) and the *Threshold* is a constant parameter. The value of b 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., equation (7)) are set to 200 and 3, respectively.

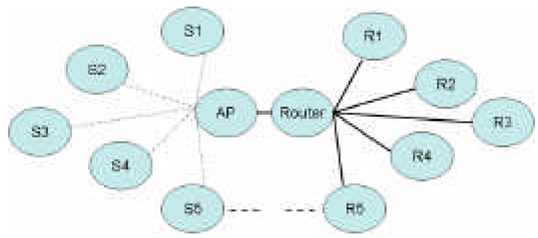


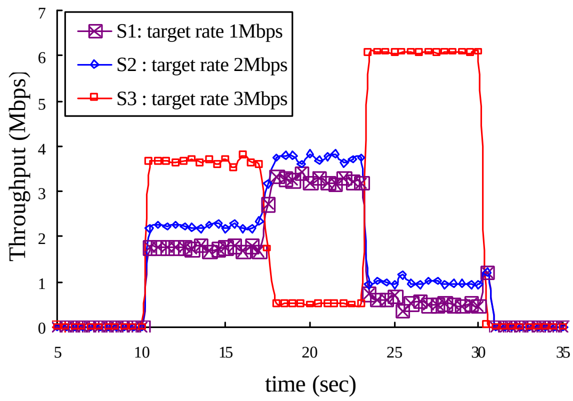
Figure 2. Simulation topology

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

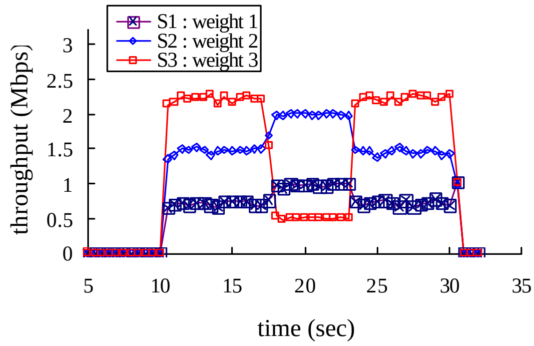
$$FI = \frac{m(\frac{R_i}{f_i})}{m(\frac{R_i}{f_i}) + s(\frac{R_i}{f_i})}, \qquad (16)$$

where m and s are the mean and the standard deviation of R_i / f_i over all data flows. The fairness index is a real value between 0 and 1. The closer to 1 the fairness index, the fairer. The stability is referred to as the fluctuation degree of the throughput of each flow in each scheduling discipline.

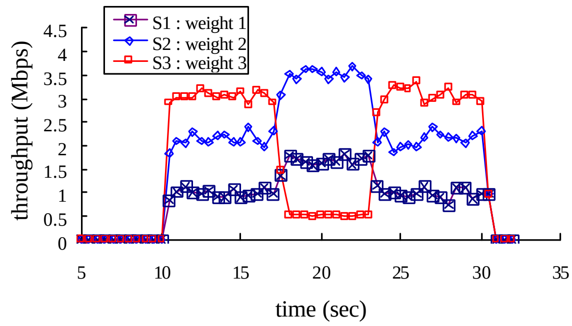
We first compare the fairness for different mechanisms with respect to network load variations. We consider three stations sending packets in the system: S1, S2, and S3. The target rates of these three stations in DDRR are set to 1Mbps, 2Mbps, and 3Mbps, corresponding to S1, S2, and S3, respectively. The assigned weights of S1: S2: 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 8Mbps to 0.5Mbps at time 17 and 0.5Mbps to 8Mbps at time 23. Fig. 3(a) shows that stations S1 and S2 fail to receive their desired service share after time 23 when DDRR is used. This is because with DDRR each node maintains the deficit counter locally, such that the DC value of S3 will be accumulated when S3 sends at a lower rate. The fairness indices of DDRR, DFS, P-MAC and IDFQ during time 10 to 17 are 0.80, 0.97, 0.96, and 0.99, respectively. We observe that DDRR has the worst fairness index during time 10 to 17 because all stations share the link capacity equally when their deficit counters become zero. Figs. 3(b) and 3(c) show 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.



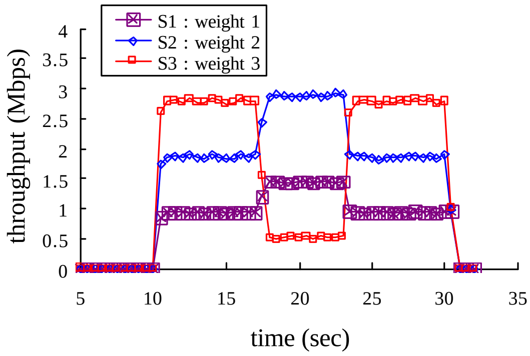
(a) DDRR



(b) DFS



(c) P-MAC

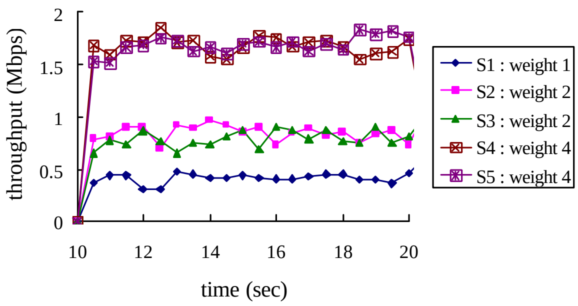


(d) IDFQ

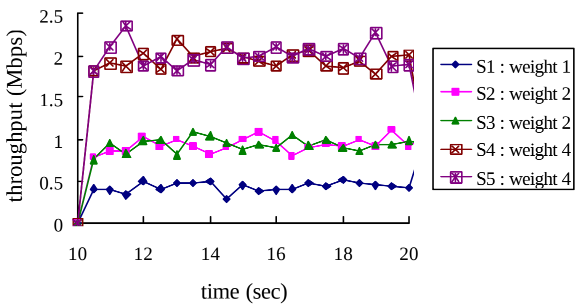
Figure. 3. Throughputs when sending rates change.

Next, we compare proportional fairness provided by different mechanisms. In this simulation, there are five wireless senders with weights 1:2:2:4:4, corresponding to S1, S2, S3, S4, and S5, respectively. Note that DDRR is not considered in the following simulations due to its implementation problem. Fig. 4 shows the throughput and fairness of each flow for each scheduling discipline. We observe that all 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.

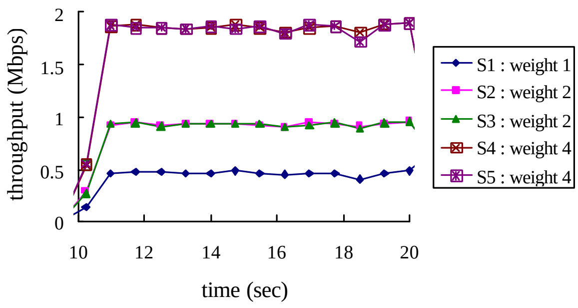
Typically, in contention-window based mechanisms, such as P-MAC, a random backoff interval is selected from a uniform distribution over the interval [0, CW] to avoid collisions, which, however, may result in a larger variance in the values of backoff intervals. In IFS-based mechanisms, the randomization factor is usually very small (for example, [0.9, 1.1] in IDFQ), thus leading to a smaller fluctuation in throughput performance. DFS, a backoff interval based mechanism, also uses a small randomization factor as in IDFQ to avoid collisions. But compared with DFS, IDFQ provides better fairness and stability, explained as follows. With SCFQ, the virtual start and finish tags of frames are only used to determine the transmission ordering of frames, and the frame with the smallest finish tag should be transmitted immediately. However, the frames with larger finish tags (or frame sizes) in DFS must be waiting for a larger backoff interval before being transmitted. On the other hand, IDFQ maps the finish tags of frames to their inter-frame space. As a result, frames with the smallest finish tag have priority to be transmitted next after waiting for a small inter-frame space time. Therefore, IDFQ is relatively stable and fair.



(a) DFS (*FI* = 0.96)

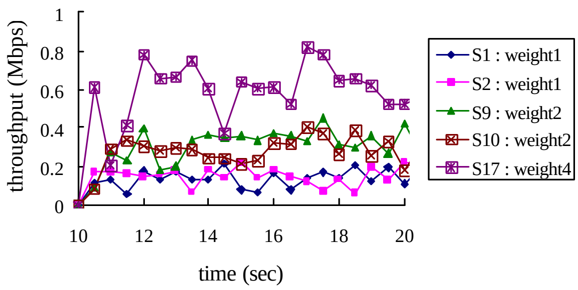


(b) P-MAC (*FI* = 0.94)

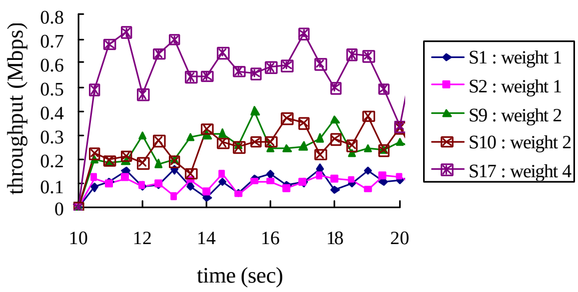


(c) IDFQ (*FI* = 0.99)

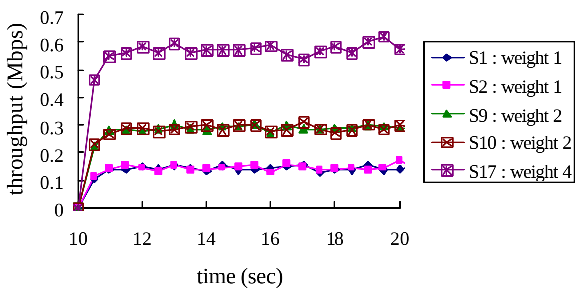
Figure. 4. The throughputs and fairness indices of three schemes with respect to the fairness performance



(a) DFS (*FI = 0.93*)

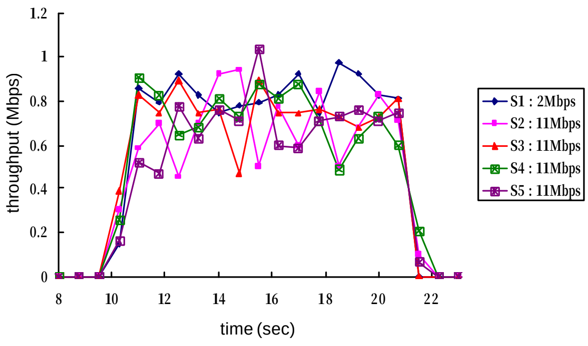


(b) P-MAC (*FI = 0.90*)

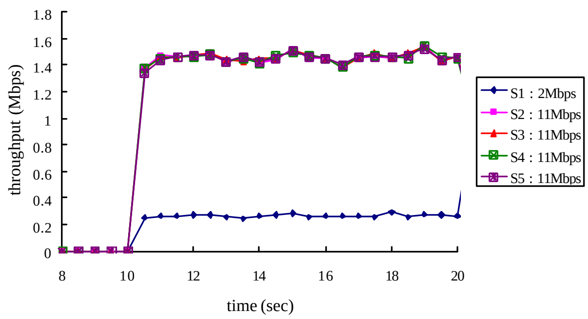


(c) IDFQ (*FI = 0.99*)

Figure. 5. The throughputs and fairness indices of three schemes with respect to the scalability performance.



(a) 802.11b DCF



(b) IDFQ

Figure 6. Throughput of flows when sending rates of stations are different

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 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 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. in [14]. 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 2Mbps, the throughput of all stations in the system will degrade to 2Mbps. 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 weighted 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 2Mbps, and the others at a rate of 11Mbps. 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 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 an IFS-based Distributed Fair Queuing (IDFQ) mechanism for proportional fairness in IEEE 802.11 WLANs. Unlike existing work based on backoff intervals for collision resolutions, the proposed mechanism chooses appropriate IFS values for flows with different weights, and applies an adaptive mechanism plus some randomization for avoiding collisions. Unlike existing IFS-based mechanisms, such as DDRR, which suffers 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 multi-hop wireless network based on IEEE 802.11 MAC. We will also conduct theoretical analysis of the proposal IDFQ mechanism with respect to the throughput performance.

REFERENCES

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

[2] I. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE," *IEEE INFOCOM 2001*.

[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, 1999, pp. 96-102

[4] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi and E. Knightly, "Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints," *ACM Mobicom 2001*

[5] W. Pattara Atikom, P. Krishnamurthy, and S. Banerjee, "Distributed Mechanisms for Quality of Service in Wireless LANs," *IEEE Wireless Communications*, Vol. 10, Issue 3, pp. 26-34, June 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 Transactions on Networking*, Vol. 1, No. 3, pp.344-357, June 1993.

[7] S. Golestani. "A Self-Clocked Fair Queueing Scheme for Broadband Applications." *IEEE INFOCOM'94*, Toronto, CA, Jane 1994

[8] S. Lu, T. Nandagopal, and V. Bharghavan, "A Wireless Fair Service Algorithm for Packet Cellular Networks," *ACM Mobicom* 1998.

[9] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in Wireless LAN," *ACM Mobicom 2000*.

[10] Daji Qiao and Kang G. Shin, "Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WALN under the DCF," *ACM IWQoS 2002*

[11] Wasan Pattara-Atikom, Sujata Banerjee, and Prashant Krishnamurthy, "Starvation Prevention and Quality of Service in Wireless LANs." *IEEE WPMC* 2002.

[12] M. Shreedhar and G. Varghese, "Efficient Fair Queuing Using Deficit Round-robin," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, 1996, pp. 375-85.

[13] ns-2, available at http://www.isi.edu/nsnam/ns/

[14] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. "Performance Anomaly of 802.11b," *IEEE INFOCOM* 2003.