MR-FQ: A Fair Scheduling Algorithm for Wireless Networks with Variable Transmission Rates

You-Chiun Wang*, Yu-Chee Tseng*, Wen-Tsuen Chen†, and Kun-Cheng Tsai‡

*Department of Computer Science and Information Engineering
National Chiao-Tung University, Hsin-Chu, 30010, Taiwan
†Department of Computer Science
National Tsing-Hua University, Hsin-Chu, 30043, Taiwan
‡Networks and Multimedia Institute, Institute for Information Industry, Taiwan

Abstract— **Wireless networks are characterized by bursty and location-dependent errors. Although many fair scheduling methods have been proposed to address these issues, most of them assume a simple two-state channel model, where a channel can be either good or bad. However, the situation is not so pessimistic since different modulation techniques can be adapted to different channel conditions. Multi-rate transmission is a common technique for wireless networks nowadays. This leads to a dilemma: should fairness be built based on the amount of medium time that a user utilizes or the amount of services that a user receives? In this work, we propose a** *Multi-rate wireless Fair Queueing (MR-FQ)* **algorithm that allows a flow to transmit at different rates according to its channel condition and lagging degree. MR-FQ takes both time and service fairness into account. It not only guarantees fairness and bounded delays for packet flows, but also increases the overall system throughput.**

Index Terms — **Communication Network, Fair Scheduling, Multi-rate Communication, Quality of Service (QoS), Wireless Network**

I. INTRODUCTION

We have seen huge growth of wireless services over recent years. The increasing importance of real-time applications demands provision of QoS and fair channel access among multiple flows over a shared, error-prone wireless channel. In wired networks, many fair scheduling methods [1], [2], [3], [4] have been proposed to bound delays of packet transmission. However, wireless channels are characterized by three features that distinguish themselves from wired networks: (i) serious bursty errors, (ii) location-dependent errors, and (iii) multi-rate communication capability. Bursty errors may break a flow's services, while location-dependent errors may let error-free flows receive more services than they deserve, thus violating the fairness. A wireless channel may provide different transmission rates to stations depending on channel qualities (e.g., IEEE 802.11a and b support 16 and 4 rates, respectively). Due to these reasons, existing wired solutions may not be suitable for the wireless networks [5].

Many fair scheduling algorithms have been proposed to address the features (i) and (ii) of wireless networks. In *IWFQ* [6], each packet is assigned a *finish tag*, and the scheduler always serves the error-free packet with the smallest tag. When a flow suffers from errors, all its packets' tags are unchanged. So when the flow exits from errors, its packets may have smaller tags, thus achieving the compensation purpose. In *CIF-Q* [7], services allocated to error flows will transfer to error-free flows. Fairness is achieved by dispatching compensation services to the former proportional to their weights. In *SBFA* [8], a fraction of bandwidth is reserved for the compensation purpose, and a number of virtual servers called *LTFS* are created for flows that experienced errors. The reserved bandwidth will be used to compensate these flows recorded in LTFS. In *TD-FQ* [9], flows are separated into real-time and non-real-time ones. The scheduler gives higher priorities to real-time flows to reduce their queuing delays, while still maintains fairness and bounded delays for all flows.

However, the feature (iii) has not been well addressed in these work. Most of them assume that the channel is either in a *good (error-free)* or *bad (error)* state. Transmissions in a good state will succeed, but completely fail in a bad state. In fact, the situation is not so pessimistic since different modulation techniques can be adapted to different channel conditions. The PHY of IEEE 802.11a and b are well-known examples, which can provide multi-rate transmission capabilities [10], [11]. A simpler modulation (and thus a higher data rate) can be used when SNR is sufficiently high, while a more complicated modulation (and thus a lower rate) can still be used under a bad channel [12]. Adopting multi-rate transmissions poses several challenges to fair queuing. First, there is a mismatch between the amount of service that a client receives and the amount of time that a server actually serves a client. To transmit the same amount of data, a client using a lower rate takes longer time than one using a higher rate. So the concept of virtual time may need to be redefined. Second, when a flow suffered from a bad channel exits from error, it may take a different amount of time for the system to compensate the flow depending on its channel condition, thus making the design of compensation difficult. Third, the system performance may be degraded if there are many low-rate flows.

In this work, we propose a new scheduling algorithm called *Multi-rate wireless Fair Queueing (MR-FQ)* in a wireless network with a TDMA MAC protocol and multi-rate communication capability. MR-FQ adjusts a flow's transmission rate based on its channel condition and lagging degree. A flow can transmit at a lower rate only if it is lagging up to a certain degree. Specifically, the more serious a flow is lagging, the lower rate it can use. Such differentiation can take care of both
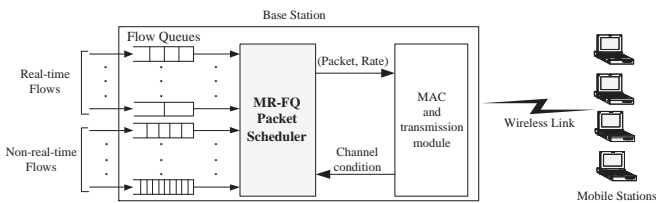
Fig. 1. System architecture of MR-FQ.

Fig. 2. The tradeoff between service fairness and time fairness.

fairness and system performance, and lower-rate flows will not prolong other flows' delays. Besides, MR-FQ follows the idea in [9] by separating real-time flows from non-real-time ones and compensates real-time lagging flows with higher priorities to reduce their delays. However, such treatment of real-time flows will not starve non-real-time flows. Thus, MR-FQ can satisfy the delay-sensitive property of real-time applications, while still maintain fairness and bounded delays for all flows.

Some works have addressed the similar issue by adjusting flows' weights, but the concept of using multiple rates are not exploited. *ELF* [13] adjusts each flow's weight by the flow's error rate, up to a maximum defined by that flow's *power factor*. In *CS-WFQ* [14], each flow i is assigned a fair share ϕ_i and a time-varying factor $f_i(t)$. The latter is to adjust the former based on error rates. In *CAFQ* [15], the weight of each flow i is adjusted by a factor $M(\Phi_i)^a$, where $M(\Phi_i)$ reflects the channel states and $0 \leq M(\Phi_i) \leq 1$. The works in [16], [17] address the multi-rate issue, but they focus on assigning codes in CDMA networks.

The remainder of this paper is organized as follows. Section II presents our MR-FQ algorithm. Section III presents some simulation results. Conclusions are drawn in Section IV.

II. THE MR-FQ ALGORITHM

A. System Model

We consider a base station (BS) as in Fig. 1. Packets arriving at the BS are classified into real-time and non-real-time traffic and dispatched into different queues. These flows are sent to the *MR-FQ packet scheduler*, which schedules flows and transmits the head-of-line (HOL) packet of the selected flow to the *MAC and transmission (MT) module*. The MT module can transmit at n rates $\hat{C}_1 > \hat{C}_2 > \cdots > \hat{C}_n$. It also measures the current channel condition to each mobile station and determines the most appropriate rate (including zero, which means that the channel is too bad) to communicate with the station (several works [12], [18] have addressed the rate selection problem, but this is out of scope of this work). The information of the best rate is also fed back to the scheduler for making decision. For simplicity, we assume that BS has immediate knowledge of the best rate for each station.

B. Service Fairness vs. Time Fairness

With multi-rate communication, the concept of fairness can be defined in two ways: *service fairness*, which means that the difference between services received by any two flows is bounded, and *time fairness*, which means that the difference between amounts of transmission time of any two flows is
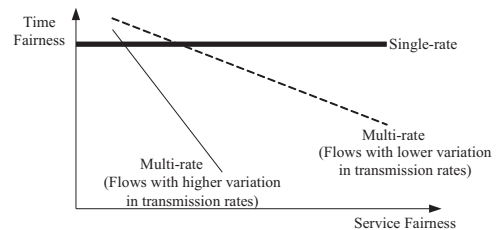
bounded. Formally, let w_i be the weight of flow i, and $\Phi_i^s(t_1, t_2)$ and $\Phi_i^t(t_1, t_2)$ be the amount of services and the amount of time that flow i receives/utilizes during the time interval $[t_1, t_2)$, respectively. For any two flows i and j,

$$\left| \frac{\Phi_i^s(t_1, t_2)}{w_i} - \frac{\Phi_j^s(t_1, t_2)}{w_j} \right| \leq \sigma_s, \quad (1)$$

holds during any $[t_1, t_2)$ if service fairness is desired , and

$$\left| \frac{\Phi_i^t(t_1, t_2)}{w_i} - \frac{\Phi_j^t(t_1, t_2)}{w_j} \right| \leq \sigma_t, \quad (2)$$

holds during any $[t_1, t_2)$ if time fairness is desired, where σ_s and σ_t are small, non-negative numbers.

We observe that in a single-rate environment, Eqs. (1) and (2) are equivalent. However, in a multi-rate environment, Eqs. (1) and (2) may not be satisfied at the same time. If service fairness is desired, flows using lower rates will occupy more time. On the contrary, if time fairness is desired, flows using higher rates will transmit more data. The concept is illustrated in Fig. 2. When the rates used by stations exhibit higher variation, the tradeoff between service and time fairness is more significant (solid line). When the variation is lower, the tradeoff is less significant (dashed line). When the variation is 0, this degenerates to the single-rate case (thick line).

C. Scheduling Policy

Fig. 2 leads to the following guidelines when designing MR-FQ. First, virtual time is redefined based on the concept of time fairness. However, we will differentiate flows based on their lagging degrees. A flow can use a lower transmission rate only if it is suffering from a higher lagging degree. In this way, we can take care of service fairness, so the system performance would not be hurt when there are too many low-rate stations.

In MR-FQ, each flow i is assigned a *weight* w_i to represent the ideal fraction of bandwidth that the system commits to it. For each flow i, we maintain a *virtual time* v_i to record the nominal services received by it, and a *lagging index* lag_i to record its credits/debits. The former is used to compete with other flows for services, while the latter is used to arrange compensation services. Flow i is called *leading* if $lag_i < 0$, called *lagging* if $lag_i > 0$, and called *satisfied* if $lag_i = 0$. Further, a flow is called *backlogged* if its queue is nonempty, called *unbacklogged* if its queue is empty, and called *active* if it is backlogged or unbacklogged but leading. Note that MR-FQ only selects active flows to serve. When an unbacklogged but leading flow is chosen, its service will actually be transferred

to another flow for compensation purpose. Besides, whenever a flow i transits from unbacklogged to backlogged, its virtual time v_i is set to $max\{v_i, min_{j \in A}\{v_j\}\}$, where A is the set of all active flows.

Fig. 3 outlines the scheduling policy of MR-FQ. First, the active flow i with the smallest virtual time v_i is selected. If flow i is backlogged, the *Rate Selection Scheme* is called to compute the best rate r to transmit for flow i. If the result is $r \leq 0$, it means that either flow i has a very bad channel or its current lagging degree does not allow it to transmit (refer to Section II-C.1). Otherwise, if flow i is non-leading, the HOL packet of flow i is served, and its virtual time is updated as:

$$v_i = v_i + \left(\frac{l_p}{w_i} \times \frac{\hat{C}_1}{r} \right), \qquad (3)$$

where l_p is the length of the packet. Note that the ratio $\frac{\hat{C}_1}{r}$ reflects the concept of time fairness. The amount of increase in v_i is inverse to the transmission rate r. So if a lower r is used, the less competitive flow i will be in the next round.

If flow i is leading, the *Graceful Degradation Scheme* will be activated to check if it is eligible for the service (refer to Section II-C.2). In case that flow i gives up its service due to an empty queue, a bad channel condition, or a rejection by the Graceful Degradation Scheme, the service will be transferred to the *Compensation Scheme* to select another flow j to serve (refer to Section II-C.3). If the scheme fails to select any flow, this service is just wasted. If the scheme still selects flow i to serve, we send its HOL packet and update v_i by Eq. (3). If another flow j ($\neq i$) is chosen, flow j's packet will be sent and the values of v_i, lag_i, and lag_j are updated as:

$$\begin{aligned} v_i &= v_i + l_{p'}/w_i, \\ lag_i &= lag_i + l_{p'}, \\ lag_j &= lag_j - l_{p'}, \end{aligned}$$

where p' is the packet being sent. Note that in this case we charge flow i by increasing its virtual time, but credit (respectively, debit) to lag_i (respectively, lag_j). Since flow i is not actually served, the first equation is equivalent to Eq. (3) with $r = \hat{C}_1$.

After the scheduler serves a flow i, it checks the queue size. If flow i changes to unbacklogged but is still lagging, we give its credit to other flows that are in debt proportional to their weights, i.e., for each flow k such that $lag_k < 0$, we set

$$lag_k = lag_k + lag_i \times \frac{r_k}{\sum_{lag_m < 0} r_m}.$$

Then we reset $lag_i = 0$. This is because the flow does not need the credit any more [19].

Below, we introduce the three schemes, Rate Selection Scheme, Graceful Degradation Scheme, and Compensation Scheme. Table I summarizes notations used in MR-FQ.

1) Rate Selection Scheme: When a backlogged flow i is selected, the Rate Selection Scheme will be invoked to choose a suitable transmission rate according to its lagging degree and channel condition. The idea is to permit different ranges of
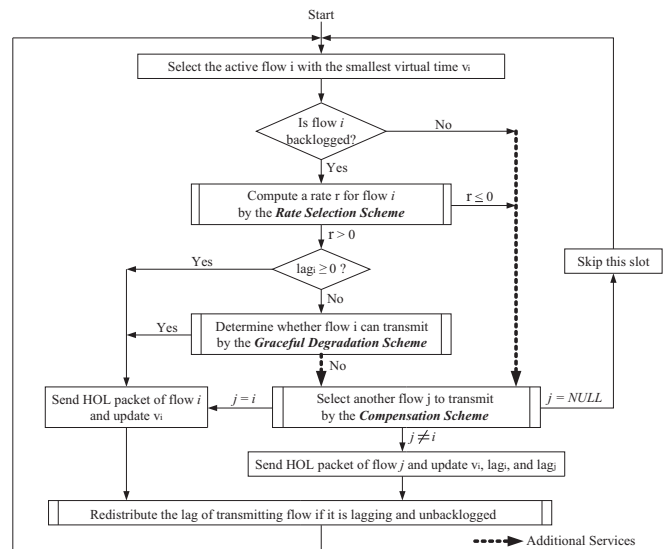

Fig. 3. The MR-FQ algorithm.

TABLE I
SUMMARY OF NOTATIONS USED IN MR-FQ.

Notations	Definition
$\hat{C}_1 - \hat{C}_n$	all transmission rates
w_i	weight of flow i
v_i	virtual time of flow i
lag_i	credits/debits of flow i
$\delta_1 - \delta_{n-1}$	thresholds to distinguish lagging degrees of flows
s_i	service index of flow i when $lag_i < 0$
α_R, α_N	degradation ratios for real-time and non-real-time flows
L_R, L_N	real-time lagging flows and non-real-time lagging flows
W_R, W_N	weights of L_R and L_N, respectively
V_R, V_N	additional services received by L_R and L_N, respectively
B	bound of differences of services for L_R and L_N
c_i	additional services received by a lagging flow i
f_i	additional services received by a non-lagging flow i

transmission rates according to flow i's normalized lag $\frac{lag_i}{w_i}$. To help a seriously lagging flow to alleviate its huge lag, we allow it to use a larger range of rates. Specifically, we set up $n - 1$ levels of lagging thresholds $\delta_1, \delta_2, \cdots, \delta_{n-1}$. A flow with a normalized lag exceeding δ_i will be allowed to use a rate as low as $\hat{C}_{i+1}$, $i \leq n - 1$. Fig. 4 shows the mapping of lagging degrees to allowable transmission rates. If flow i's current best rate falls within the allowable range, this rate will be returned. Otherwise, a negative value will be returned to indicate a failure. For example, if flow i satisfies $\delta_2 < \frac{lag_i}{w_i} \leq \delta_3$ and its current best rate is $\hat{C}_2$, $\hat{C}_2$ will be returned. If the current best rate is $\hat{C}_5$, a negative value will be returned.

2) Graceful Degradation Scheme: When a leading flow i is selected, the Graceful Degradation Scheme is triggered to check its leading amount. A leading flow is allowed to receive an amount of extra service proportional to its normal services. Specifically, when a flow i transits to leading, we set up a parameter $s_i = \alpha \cdot v_i$, where $\alpha \in [0, 1]$ is a constant. Later on, flow i's virtual time will be increased each time when it is selected by the scheduler. Let v_i' be flow i's current virtual time

lagging degrees	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$	$\cdots$	$\hat{C}_{n-1}$	$\hat{C}_n$
$\frac{lag_i}{w_i} \le \delta_1$	✓					
$\delta_1 < \frac{lag_i}{w_i} \le \delta_2$	✓	✓				
$\delta_2 < \frac{lag_i}{w_i} \le \delta_3$	✓	✓	✓			
⋮				⋮		
$\delta_{n-2} < \frac{lag_i}{w_i} \le \delta_{n-1}$	✓	✓	✓	✓	✓	
$\delta_{n-1} \le \frac{lag_i}{w_i}$	✓	✓	✓	$\cdots$	✓	✓

Fig. 4. The mapping of lagging degrees to allowable transmission rates in the Rate Selection Scheme.

Fig. 5. Dispatching additional services in the Compensation Scheme.

when it is selected. We allow flow i to be served if $s_i \le \alpha v_i'$. If so, s_i is updated as $s_i + l_p/r_i$, where l_p is the length of the packet. Intuitively, a leading flow i can enjoy approximately $\alpha(v_i' - v_i)$ services. Moreover, we substitute the above α by a parameter α_R for real-time flows, and by α_N for non-real-time flows. We set $\alpha_R > \alpha_N$ to distinguish their priorities.

3) Compensation Scheme: When the selected flow cannot be served, the Compensation Scheme is triggered. Fig. 5 shows how to dispatch such *additional services (AS)*. Flows are prioritized based on the following rules: (i) lagging flows have a higher priority over non-lagging flows, (ii) flows that can use higher rates to transmit have a higher priority over flows that can use lower rates, and (iii) among lagging flows of the same best rate, real-time flows and non-real-time ones will share the services according to some ratio. In the rule (iii), when dispatching AS to lagging flows, we keep track of the services received by real-time ones and non-real-time ones. Let $L_R = L_R^1 \cup L_R^2 \cup \cdots \cup L_R^n$ (where the superscript k means that the best rate is $\hat{C}_k$) be the set of real-time, lagging flows, and $L_N = L_N^1 \cup L_N^2 \cup \cdots \cup L_N^n$ the set of non-real-time, lagging flows. To let real-time lagging flows receive more AS without starving non-real-time lagging flows, we assign weights W_R and W_N to L_R and L_N, respectively, to control the fractions of AS they can receive, where $W_R > W_N$. A virtual time V_R (respectively, V_N) is used to record the normalized AS received by L_R (respectively, L_N). Flows in Fig. 5 is checked from left to right. When both L_R^k and L_N^k are non-empty, the service is given to L_R if $V_R \le V_N$, and to L_N otherwise. When only one of L_R^k and L_N^k is non-empty, the service will be given to that one, independent of the values of V_R and V_N. When a flow in L_R receives the service, V_R is updated as

$$V_R = min\left\{ V_R + \frac{l_p}{W_R}, \frac{B + V_N W_N}{W_R} \right\}, \qquad (4)$$

where B is a constant to bound the difference between V_R and V_N. Similarly, when a flow in L_N receives the service, V_N is updated as

$$V_N = min\left\{ V_N + \frac{l_p}{W_N}, \frac{B + V_R W_R}{W_N} \right\}. \qquad (5)$$

Note that to avoid that $V_R \gg V_N$ (respectively, $V_N \gg V_R$), which may cause flows in L_R (respectively, L_N) to starve, we set up a bound $|V_R W_R - V_N W_N| \le B$. This is reflected by the second term in the righthand side of Eqs. (4) and (5). When the scheduler selects either L_R^k or L_N^k, it distributes AS
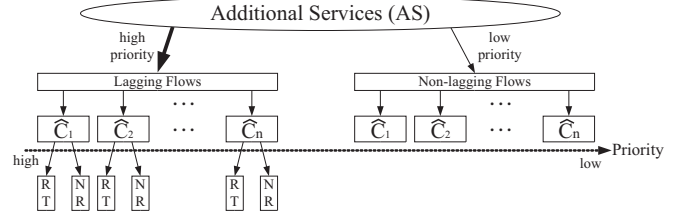
proportional to the weights of flows in that set. Specifically, for each flow i, we maintain a *compensation virtual time* c_i to keep track of the normalized amount of AS received by it. The scheduler selects the flow i with the smallest c_i to serve, and updates c_i as $c_i + \left(\frac{l_p}{w_i} \times \frac{\hat{C}_1}{\hat{C}_k} \right)$. Initially, when a flow i newly enters L_R or L_N, its c_i will be set to $max\{c_i, min_{j \in L}\{c_j\}\}$, where L can be either L_R or L_N, depending on its traffic type.

If there is no lagging flow in the previous stage, the service is returned back to the originally selected flow if it is a leading flow but rejected by the Graceful Degradation Scheme. Otherwise, the services will be given to a non-lagging flow that can use the highest rate. In case of a tie, MR-FQ will dispatch the services proportional to some weights. Specifically, each flow i is assigned an *extra virtual time* f_i to keep track of the normalized amount of AS received by flow i when $lag_i \le 0$. When a backlogged flow i that can send becomes non-lagging, f_i is set to

$$f_i = max\{f_i, min\{f_j \,|\text{flow } j \text{ is backlogged, non-lagging}$$
$$\text{and can send}, j \ne i\}\}.$$

The scheduler selects the flow i with the smallest f_i to serve. When flow i receives the service, f_i is updated as $f_i + \left(\frac{l_p}{w_i} \times \frac{\hat{C}_1}{r} \right)$, where r is the current rate for flow i.

III. SIMULATION RESULTS

In this section, we present some experimental results to verify the effectiveness of the proposed algorithm. We mix real-time and non-real-time traffics together, and observe the packet dropping ratios of real-time flows and the throughput of non-real-time flows. We compare CIF-Q [7], TD-FQ [9], and the proposed MR-FQ. We adopt the IEEE 802.11b as the MAC protocol, which provides 11, 5.5, 2, and 1 Mb/s transmission rates. Ten flows are used, as shown in Table II. The first six flows are real-time flows, which represent three traffic models: voice, video, and CBR. The voice is modeled as an ON-OFF process, where the average duration of ON and OFF periods are set to 2.5 and 0.5 seconds, respectively. During an ON period, packets are generated with fixed intervals. No packet is generated during an OFF period. In a video traffic, packets arrive in a Poission fashion. The last four flows are non-real-time FTP flows, and their traffics are modeled as greedy sources whose queues are never empty. For error scenarios, we use T_{good} and T_{bad} to adjust the average time when a channel stays in good and bad states, respectively. When the channel is in the good state, the flow can use 11 Mb/s to transmit.

When the channel is in the bad state, the best transmission rate that a flow can use is randomly selected from 5.5, 2, 1, and 0 Mb/s. The total simulation time in each experiment is 30 minutes. For CIF-Q, we set its parameter $\alpha = 0.5$, while for TD-FQ and MR-FQ, we set their parameters $\alpha_R = 0.8$ and $\alpha_N = 0.2$, respectively. In TD-FQ, the weights assigned to lagging sets are $W_R : W_N = 3 : 1$, $W_R^S : W_R^M = 3 : 1$, and $W_N^S : W_N^M = 3 : 1$. In MR-FQ, we set $W_R : W_N = 3 : 1$.

TABLE II
TRAFFIC SPECIFICATION OF THE FLOWS USED IN THE EXPERIMENT.

Flow	Rate	Error scenario
voice1	64 Kb/s	T_{good}=8 sec., T_{bad}= 1.5 sec.
voice2	32 Kb/s	T_{good}=5 sec., T_{bad}= 1 sec.
video1	2 Mb/s	T_{good}=8 sec., T_{bad}= 1.5 sec.
video2	1 Mb/s	T_{good}=5 sec., T_{bad}= 1 sec.
CBR1	512 Kb/s	T_{good}=8 sec., T_{bad}= 1.5 sec.
CBR2	256 Kb/s	T_{good}=5 sec., T_{bad}= 1 sec.
FTP1	2 Mb/s	T_{good}=9.5 sec., T_{bad}= 0.5 sec.
FTP2	2 Mb/s	T_{good}=8 sec., T_{bad}= 1.5 sec.
FTP3	2 Mb/s	T_{good}=5sec., T_{bad}= 1 sec.
FTP4	2 Mb/s	T_{good}=3 sec., T_{bad}= 1 sec.

The packet dropping ratios of real-time flows and throughput of non-real-time flows are shown in Table III, where the deadline of a packet is set to twice of the average packet interarrival time. From Table III, we can observe that the packet dropping ratios of real-time flows in MR-FQ are smaller than those in CIF-Q and TD-FQ. This is because MR-FQ allows flows in a bad state to transmit packets using lower rates, while other schemes treat such flows as suffering from errors. The similar effect can be observed from the throughput of non-real-time flows.

TABLE III
EXPERIMENTAL RESULTS (DROPPING RATIO AND THROUGHPUT).

Flow	CIF-Q	TD-FQ	MR-FQ
voice1	24.42%	21.73%	19.25%
voice2	49.14%	46.71%	41.18%
video1	33.87%	33.32%	30.52%
video2	36.08%	35.75%	32.55%
CBR1	25.91%	23.94%	21.79%
CBR2	30.01%	24.22%	22.06%
FTP1	1.57 Mb/s	1.54 Mb/s	1.61 Mb/s
FTP2	1.54 Mb/s	1.53 Mb/s	1.59 Mb/s
FTP3	1.48 Mb/s	1.49 Mb/s	1.52 Mb/s
FTP4	1.36 Mb/s	1.39 Mb/s	1.43 Mb/s

IV. CONCLUSIONS

We have addressed the problem that many existing wireless fair scheduling algorithms ignore the multi-rate transmission capability of wireless networks. A new algorithm, MR-FQ, is proposed to solve this problem. By taking both time fairness and service fairness into account, MR-FQ allows a flow to transmit at different rates based on its channel condition and lagging degree, thus increasing the system performance. Simulation results have shown that MR-FQ incurs less packet dropping for real-time flows and has larger throughput for non-real-time flows when compared to CIF-Q and TD-FQ.

REFERENCES

[1] 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. on Networking*, vol. 1, no. 3, pp. 344–357, 1993.
[2] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *Journal of Internetworking Research and Experience*, vol. 1, pp. 3–26, 1990.
[3] P. Goyal, H. M. Vin, and H. Chen, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," in *Conf. on Applications, technologies, architectures, and protocols for computer communications*, 1996, pp. 157–168.
[4] J. C. R. Bennett and H. Zhang, "$\mathcal{W}F^2Q$: worst-case fair weighted fair queueing," in *INFOCOM*, vol. 1, 1996, pp. 120–128.
[5] Y. Cao and V. O. K. Li, "Scheduling algorithms in broadband wireless networks," in *IEEE Proceedings of the IEEE*, vol. 89, no. 1, 2001, pp. 76–87.
[6] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. on Networking*, vol. 7, no. 4, pp. 473–489, 1999.
[7] T. S. E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," in *INFOCOM*, vol. 3, 1998, pp. 1103–1111.
[8] P. Ramanathan and P. Agrawal, "Adapting packet fair queuing algorithms to wireless networks," in *Int'l Conf. on Mobile Computing and Networking*, 1998, pp. 1–9.
[9] Y. C. Wang, S. R. Ye, and Y. C. Tseng, "A fair scheduling algorithm with traffic classification in wireless networks," in *Int'l Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, 2004, pp. 502–509.
[10] LAN/MAN Standards Committee of the IEEE Computer Society, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHZ Band," *IEEE Std 802.11a*, 1999.
[11] LAN/MAN Standards Committee of the IEEE Computer Society, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," *IEEE Std 802.11b*, 1999.
[12] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media sccess for multirate ad hoc networks," in *Int'l Conf. on Mobile Computing and Networking*, 2002, pp. 24–35.
[13] D. A. Eckhardt and P. Steenkiste, "Effort-limited fair (ELF) scheduling for wireless networks," in *INFOCOM*, vol. 3, 2000, pp. 1097–1106.
[14] P. Lin, B. Benssou, Q. L. Ding, and K. C. Chua, "CS-WFQ: a wireless fair scheduling algorithm for error-prone wireless channels," in *Proceedings of Computer Communications and Networks*, 2000, pp. 276–281.
[15] L. Wang, Y. K. Kwok, W. C. Lau, and V. K. N. Lau, "Channel capacity fair queueing in wireless networks: issues and a new algorithm," in *IEEE Int'l Conf. on Communications*, 2002, pp. 3116–3120.
[16] A. Stamoulis and G. B. Giannakis, "Packet fair queueing scheduling based on multirate multipath-transparent CDMA for wireless networks," in *INFOCOM*, vol. 3, 2000, pp. 1067–1076.
[17] A. Stamoulis, N. D. Sidiropoulos, and G. B. Giannakis, "Time-varying fair queueing scheduling for multicode CDMA based on dynamic programming," *IEEE Tran. on Wireless Communications*, vol. 3, pp. 512–523, 2004.
[18] K. Balachandran, S. R. Kadaba, and S. Nanda, "Channel quality estimation and rate adaptation for cellular mobile radio," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1244–1256, 1999.
[19] Y. Yi, Y. Seok, T. Kwon, Y. Choi, and J. Park, "$\mathcal{W}^2F^2Q$: packet fair queuing in wireless packet networks," in *ACM Int'l Workshop on Wireless Mobile Multimedia*, 2000, pp. 2–10.