A Fair Distributed Packet Scheduling Algorithm for Wireless LANs

Huei-Wen Ferng, Chung-Fan Lee, Jeng-Ji Huang, Jin-Hui Lin, Hung-Yan Gu, and Ge-Ming Chiu

Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei 106, Taiwan
E-mail: hwferng@mail.ntust.edu.tw

***Abstract*— In this paper, a packet scheduling algorithm called distributed elastic round robin (DERR) suitable for operation in a distributed environment is proposed to provide fair scheduling for the IEEE 802.11 wireless LAN. By simulations, we show that DERR possesses slightly improved performance in throughput and delay and exhibits better fairness than distributed deficit round robin (DDRR), which is a previously proposed fair scheduling algorithm in the literature.**

I. INTRODUCTION

The increasing popularity of mobile devices, e.g., cellular phone, notebook, personal digital assistant (PDA) etc., has indicated that a new era of mobile computing and wireless networks is coming soon. In contrast with the troublesome problem in planning the layout of a wireline system, wireless communication is convenient in setup, flexible in space arrangement, easy in extension, and so on. This makes the wireless LAN (WLAN) mainly based on the IEEE 802.11 standard become one of most widely used wireless systems.

In IEEE 802.11, the transmission rate is limited to 2 Mbps only, while extended versions specified in 1999 are able to provide 54 Mbps for IEEE 802.11a and 11 Mbps for IEEE 802.11b, respectively. In common with most wireless systems, WLAN has scarce bandwidth. Thus, it is an important issue to effectively utilize the transmission medium, to maximize the utilization, and to provide stable services so as to offer realtime multimedia in the WLAN and satisfy users' demands. Several existing applications require certain degree of quality of service (QoS), e.g., throughput, delay etc. However, the QoS requirements cannot be guaranteed simply using the distributed coordination function (DCF) in IEEE 802.11 WLAN, which is performed at the medium access control (MAC) layer for asynchronous bulk data transfer.

In recent years, many schemes to improve the DCF have been proposed to enhance the performance of WLAN. These schemes intend to provide different service classes for various throughput requirements using priorities. However, they often sacrifice service demands of users with lower priorities. To solve this problem, schemes focused on fairness are then proposed. Previous works concerning fair scheduling include distributed fair scheduling (DFS) [2], distributed weighted fair queueing (DWFQ) [3], and DDRR [4]. To achieve fairness, DFS proposes to adjust the backoff interval, while DWFQ intends to adapt the contention window. Although fairness is accomplished under both schemes, they cannot avoid the backoff procedure which can cause throughput and delay to fluctuate. On the other hand, DDRR employs random inter-frame space (IFS) mapping (so does our proposed DERR). This makes DDRR have better performance in terms of throughput and delay.

In this paper, a new scheduling mechanism, i.e., DERR is proposed. DERR is a distributed version of ERR [1] which is a fair scheduling mechanism used in centralized wireline environments. DERR utilizes the quantity of $allowance$ like ERR in scheduling of traffic flows based on the knowledge of their previous transmissions in the one-hop ad hoc environment to fairly distribute system resources. In this paper, we also compare the performance of DDRR and DERR via simulations and witness that the DERR not only improves the throughput and delay but also has an edge over the DDRR in fairness.

To investigate the fairness in more detail, we have derived upper bounds of the fairness for DERR and DDRR. However, due to the limitation on paper length, the derivation is not included in this paper but it can be found in [6].

The rest of this paper is organized as follows. Section II introduces some relevant mechanisms. After that, we illustrate the proposed DERR in Section III. The simulation results and discussions are presented in Section IV. Finally, concluding remarks are given in Section V.

II. RELEVANT MECHANISMS

In this section, basic operations for DCF of IEEE 802.11 and the DDRR scheduler are briefly reviewed.

A. Basic Operations of IEEE 802.11

Two coordination functions, i.e., centralized and distributed, are used at the MAC layer of IEEE 802.11 for coordinating among hosts to decide which one to transmit next. The centralized one, i.e., point coordination function (PCF) is controlled by a point coordinator, which can be either an access point or a host selected in advance. On the other hand, DCF is a distributed function that employs carrier-sense multiple access/collision avoidance (CSMA/CA) to provide asynchronous

0-7803-8521-7/04/$20.00 (C) 2004 IEEE

data transmission without the aid of a coordinator in the determination of transmission order.

B. The DCF

As mentioned previously, there are two coordination functions at the MAC layer of IEEE 802.11. Data can be transmitted in either the contention mode under DCF or the contention-free mode under PCF. The DCF is the most basic function for multiple access. All hosts are required to equip with this function no matter they are operating in an ad hoc or an infrastructure network. The underlying mechanism of the DCF is CSMA/CA, which performs carrier sense and collision avoidance simultaneously and is best suitable for delay-insensitive asynchronous data transfer.

Before accessing a medium, a host has to perform the carrier sense. When the idle medium is sensed, the host first waits for a period of DCF IFS (DIFS) prior to proceeding with a backoff procedure. The backoff interval is chosen within [0, CW], where CW is the contention window. Due to the fact that it is almost impossible for two hosts to have the same backoff interval, the packet collision probability would be low. The backoff timer is decreased only when the medium is idle. As the timer goes down to zero and the medium is still idle, a packet is then allowed to be transmitted. After transmission (no matter it is successful or not), the backoff procedure must be performed again before the next transmission is performed. It should be noted that the value of CW is doubled till a maximum $\mathrm{CW}_{\max}$ is reached when a transmission attempt fails in order to reduce packet collision probability. In Fig. 1, the DCF mechanism is shown.

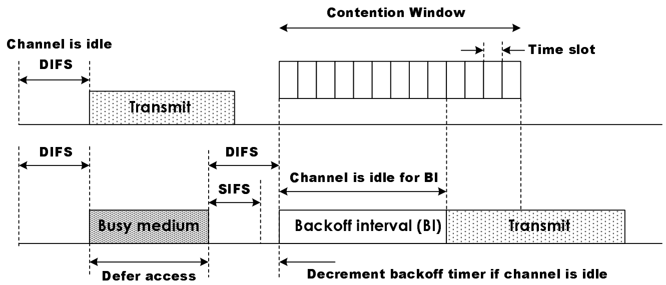

Fig. 1. The DCF Mechanism.

C. Review of DDRR

The main idea of DDRR is to apply deficit round robin (DRR) into the IEEE 802.11 MAC. Firstly, all hosts are categorized into several service classes according to their throughput requirements and a service quantity of Q bits (per T_i second) is assigned to each host. Obviously, a host belonging to a higher service class has a larger Q. A deficit counter DC_i^j is then assigned to host j in class i. Its value is increased at a rate of Q bits per T_i seconds and is decreased by the size of a packet after it is sent. Clearly, Q/T_i is equal to the desired throughput of class i.

When a host has packets to transmit, it first performs carrier sense; if the idle medium is sensed, the DC_i^j is then randomly mapped into an IFS. After waiting for a period of the IFS and the medium is still idle, the packet is then transmitted; otherwise, the whole process repeats again. In order to conform to IEEE 802.11, the range of the IFS should fall between DIFS and PIFS. DDRR thus removes the backoff procedure. This makes it capable of reducing the variation in the performance of throughput and delay mainly caused by the backoff procedure. In addition, DDRR utilizes the DC_i^j in scheduling. This not only avoids packet collision but also prevents the starvation of low priority hosts.

III. THE PROPOSED DERR

DERR is a modified and distributed version of ERR used in centralized wireline networks. Similar to DDRR, DERR intends to apply ERR into IEEE 802.11 MAC by making some changes to DCF. Based on information about last transmission of each traffic flow, DERR dynamically adjusts the transmission order among all flows. In one-hop ad hoc networks, DERR improves the performance of throughput and delay and achieves better fairness as compared to DDRR.

Suppose that there is no coordinator in a network and all hosts are categorized into service classes according to their QoS requirements. In DERR, the amount of bits allowed to be sent after acquiring the medium is called $allowance$. The value of allowance for host j in class i at time t is denoted by $A_i^j(t)$. The allowance is adjustable and the actual amount of data transmitted each time is not less than the allowance. This results in an extra amount of data transmitted which is called $excess$. It is then used in adjusting allowance. Let $E_i^j(t')$ and $F_i^j(t')$ be the excess and the total amount of data sent at time t'. The relation between the excess and the allowance is shown as follows:

$$E_i^j(t') \quad = \quad F_i^j(t') - A_i^j(t'). \tag{1}$$

After waiting for a period equal to an IFS, the host can continue transmitting data till the total bits transmitted exceeds the allowance if the medium is idle; that is, the service received by the host has already been more than its fair share. The excess is recorded by each host and a $T_{E,i}$ is assigned to each class i. The ratio of $E_i^j(t')/T_{E,i}$ is relevant to the throughput requirement of class i. For time t $(t > t')$, the allowance has the following relation:

$$A_i^j(t) = \frac{E_i^j(t')}{T_{E,i}}(t - t') - E_i^j(t'). \tag{2}$$

The allowance is dynamically adjusted between two consecutive transmissions and the longer a host waits for its next transmission, the larger the allowance will be. Thus, a low priority host would eventually have a chance to send its data and the starvation problem is prevented. In addition, the allowance also relates to the last excess. That is, a large excess in the last transmission leads to a small allowance. This is quite fair in the allocation of system resources.

The calculation of allowance and excess in the proposed DERR is similar to that of allowance and surplus count in ERR since both schemes adopt adaptive allowance to achieve fairness. However, the allowance is obtained for DERR using (2) which is different from that in ERR.

In order to apply the above idea into IEEE 802.11 DCF, the allowance is randomly mapped into an IFS. The detailed operation is described as follows. First of all, when a host has data to send and the medium is idle, it waits for an interval of IFS. If the medium is still idle after its waiting, the host then begins to transmit its packets; otherwise, it has to wait for another IFS. Therefore, no backoff procedure is performed. As for the IFS for host j in class i, it is got using the following mapping:

$$IFS_i^j(t) = DIFS - \alpha_E A_i^j(t) rand(1, \beta_E), \tag{3}$$

where α_E is a constant and is used to adjust the IFS in DERR such that it is compatible to IEEE 802.11. The IFS is obtained through multiplying the allowance by a random value between 1 and β_E (> 1). So, even if some hosts may possess the same allowance, their IFSs are different. This effectively avoids possible packet collisions in the original IEEE 802.11 operation. The DERR IFS is confined within the interval between PIFS and DIFS and it is shown in Fig. 2.

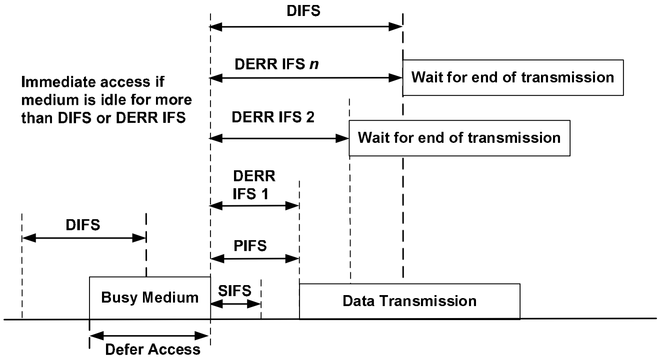

Fig. 2. The IFS of DERR.

Both DERR and DDRR employ mapping approaches to avoid collisions. However, deficit count (DC) in DDRR is used in the mapping, while allowance is used in DERR. Since the definition and the range of DC and allowance are different, there exist different scaling factors between them. In the following, we use subscripts $_D$ and $_E$ to differentiate factors for DDRR and DERR, respectively, if necessary.

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, the performance of DERR and DDRR is compared via simulations written in C programming language. We first examine the performance of throughput and delay under both mechanisms, and then compare the fairness performance by using two performance metrics: fairness index [5] and throughput/weight ratio. In the following, simulations are conducted in a 2 Mbps ad hoc WLAN, in which no access point exists and no RTS/CTS is employed. Unless specifically mentioned, the following parameters are used throughout all simulation runs: 8 hosts in the system, 100 kbps for data arrival rate of each host, 10% overhead added to packet headers (This results in the data arrival rate of 110 Kbps at the MAC layer), 1000 bytes for the length of every data packet sent, $\beta_D = \beta_E = 1.9$, $K_E = K_D = 10^{-6}$ (These two factors can be related to weights w_i equal to $K_D/T_{D,i}$ for DDRR and $K_E/T_{E,i}$ for DERR, where $T_{D,i}$ and $T_{E,i}$ denote time periods.), and $Q = 80$ bits.

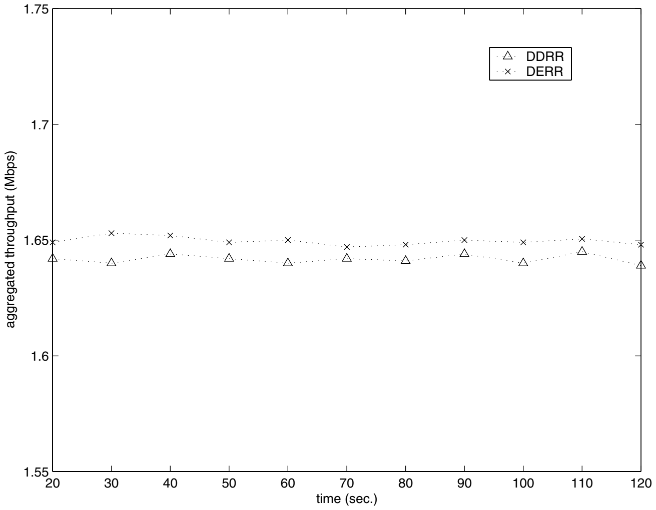

Fig. 3. Aggregated throughput for DDRR and DERR.

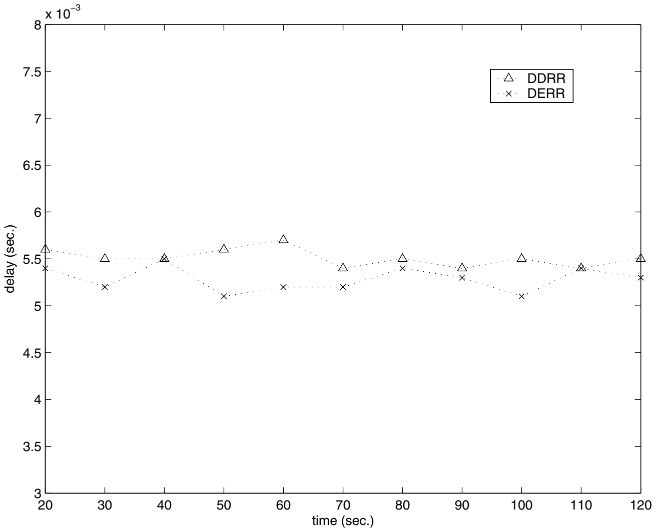

Fig. 4. Delay time for DDRR and DERR.

A. Comparison of Throughput and Delay

Figs. 3 and 4 show the aggregated throughput and average delay vs. simulation time, respectively. After running for a few seconds, an equilibrium state is reached. As shown in these two figures, DERR slightly improves DDRR in terms of throughput and delay and the improvements are about 1.2% and 3.3%, respectively, where the inevitable performance fluctuation is mainly due to the employed contention mechanisms. Although DERR indeed improves throughput and delay of

DDRR, the performance gain is marginal. However, we shall show that the fairness for DERR is much better than that of DDRR in the following subsection.

B. Comparison of Fairness

To gauge fairness, the *fairness index* is defined in [5] as shown in the following:

$$fairness\ \ index = \frac{\left(\sum_f \frac{S_f}{w_f}\right)^2}{\sum_f \sum_f \left(\frac{S_f}{w_f}\right)^2}, \tag{4}$$

where S_f and w_f represent the throughput and the weight of traffic flow f. It is clear that the closer fairness index approaches to 1, the better a fair scheduler is.

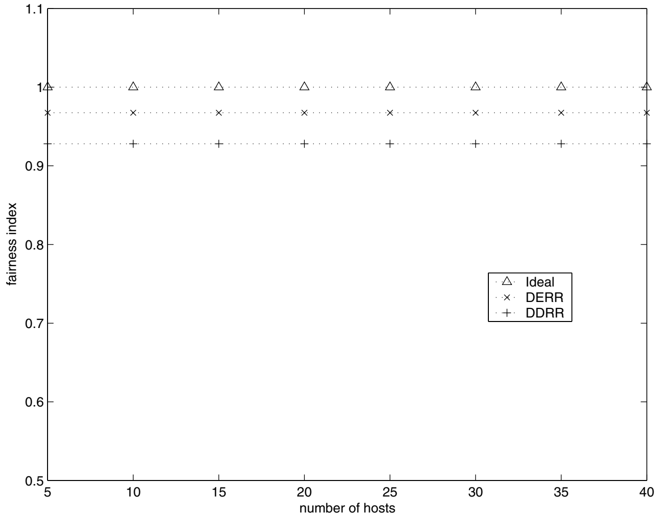

Fig. 5. Fairness index.

Let us first compare fairness performance between two mechanisms using the fairness index. Hosts are classified into five groups having weights equal to 0.02, 0.03, 0.05, 0.1, and 0.8. As shown in Fig. 5, the fairness index is 0.968 for DERR and 0.929 for DDRR. We note that the reciprocal of the difference of fairness indices between each scheme and the ideal scheme can show how fair a scheme is. Thus, they are 31.25 for DERR and 14.08 for DDRR. Obviously, DERR is more fair than DDRR with 122% improvement in terms of the reciprocal of difference. The improvement of our proposed DERR is mainly due to that allowance is used to dynamically adjust transmission order. Fig. 5 also shows that the increase in the number of hosts has little impact on the fairness performance.

Next, we focus on investigating throughput/weight ratio under the two mechanisms. The ratio is obtained by dividing the mean throughput of a class by its corresponding weight. Four classes with weights equal to 0.02, 0.03, 0.05, and 0.9 are assumed in the system and each class includes two hosts. In Figs. 6 and 7, throughput/weight ratios are plotted for DDRR and DERR, respectively. Obviously, DERR much more concentrates than DDRR. To illustrate how the concentration is, we calculate variances of throughput/weight ratios for DDRR and DERR. The variance is 0.018 for DDRR and 0.00063 for DERR. This definitely suggests that DERR has an edge over DDRR in fairness since the variance of DDRR is 29 times larger than that of DERR.

Thus, we demonstrate that DERR has much better fairness than DDRR using both the fairness index and the ratio of throughput over weight.

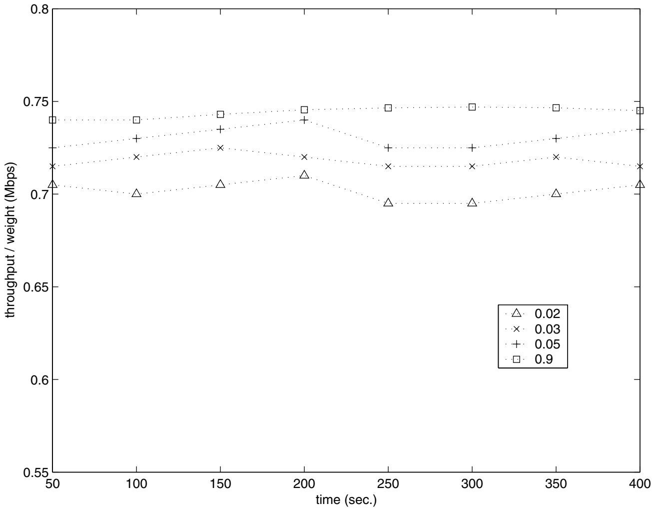

Fig. 6. Throughput/weight for DDRR.

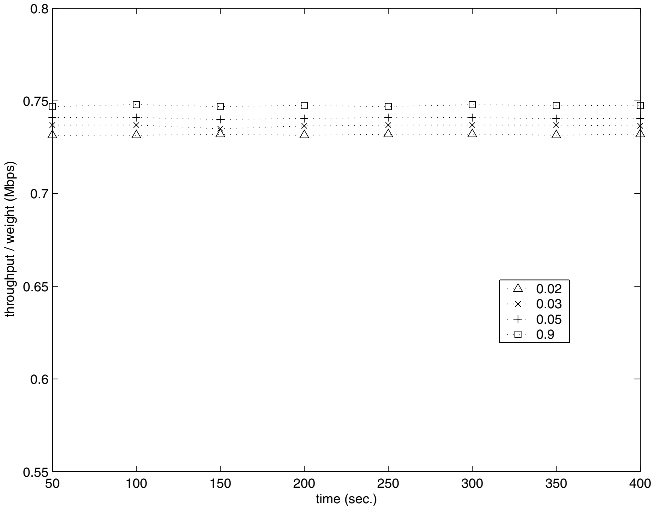

Fig. 7. Throughput/weight for DERR.

V. CONCLUSIONS

In this paper, a fair scheduling algorithm called DERR is proposed for IEEE 802.11 WLAN ad hoc networks. In the proposed DERR, IEEE 802.11 DCF is modified and a mapping of allowance into IFS is performed. The allowance is dynamically adjusted according to both waiting time and excessive amount of last data transmission. Through simulations, we have compared the performance between DDRR and

DERR, and the results show that our DERR not only improves both throughput and delay but also has much better fairness performance than DDRR.

REFERENCES

[1] S. S. Kanhere, H. Sethu, and A. B. Parekh, "Fair and efficient packet scheduling using elastic round robin," *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 324–336, March 2002.

[2] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proc. ACM MobiCom '00*, pp. 167–178, Aug. 2000.

[3] A. Banchs and X. Perez, "Distributed weighted fair queueing in 802.11 Wireless LAN," in *Proc. IEEE ICC '02*, pp. 3121–3127, May 2002.

[4] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy, "Starvation prevention and quality of service in wireless LANs," in *Proc. IEEE Wireless Personal Multimedia Communications '02*, pp. 1078–1082, Oct. 2002.

[5] R. Jain, G. Babic, B. Nagendra, and C. Lam, "Fairness, call establishment latency and other performance metrics," Tech. Rep. ATM_Forum/96-1173, ATM Forum Document, Aug. 1996.

[6] C. F. Lee, *Fair and Efficient Scheduling Mechanisms for IEEE 802.11 Wireless LANs*, Master Thesis, Chapter 2, Dept. of Comp. & Inform. Eng., Natl. Taiwan Univ. of Sci. and Tech., Taipei, Taiwan, July 2004.