A Fair Scheduling Mechanism for the IEEE 802.11e Wireless LAN

Huei-Wen Ferng, Chung-Fan Lee, Han-Yu Liau, and Cheng-Ching Peng

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

A fair scheduling mechanism called elastic enhanced distributed coordination function (EEDCF) based on the elastic round robin (ERR) is proposed in this paper to improve the enhanced distributed coordination function (EDCF) of the IEEE 802.11e wireless LAN. EEDCF dynamically adjusts the backoff interval according to different priorities and the collision status of different queues using a flexible adjustment procedure to enhance the fairness and transmission efficiency. Observing performance of EEDCF and EDCF in terms of throughput, delay, dropping probability, and fairness etc., we demonstrate that EEDCF outperforms EDCF.

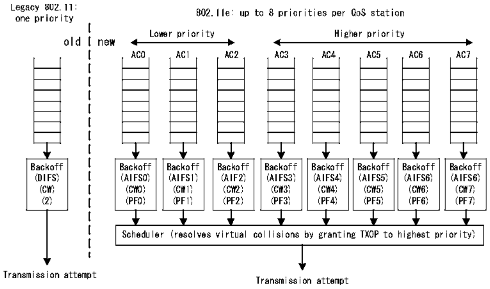


Fig. 1. Legacy DCF vs. EDCF.

1: INTRODUCTION

IEEE 802.11 has become one of popular contemporary wireless standards. However, there are some drawbacks pointed by [2], [6] because of its medium access control (MAC), i.e., distributed coordination function (DCF) and point coordination function (PCF); for example, the service differentiation and transmission priority used for delay and delay jitter guarantee in current multimedia networks are not provided by IEEE 802.11. Thus, the throughput gets down and the delay rises when the traffic load gets high for DCF. As for PCF, it is not widely applied because of unpredictability in transmission time. Therefore, IEEE 802.11 task group E undertook the modification of DCF to support multimedia transmission in IEEE 802.11 wireless LANs. The enhanced DCF is then called EDCF which strengthens some functions of DCF with multiple queues of different priorities to take care of different quality of service (QoS) requirements (see Fig. 1). In the literature, many papers have focused on the issue of how to achieve QoS requirements in wireless multimedia networks. For example, [1] and [8] have studied how to fulfill different QoS requirements in multimedia networks. In [10] and [15], the authors applied the queue-based mechanism to reduce collisions. In this paper, we also design a scheduling mechanism considering QoS requirements.

About the status of scheduling, many scheduling algorithms have been proposed and studied in the literature for both wired and wireless networks, e.g., [3], [5], [7], [12], [13], [14], and [16]. As for fair scheduling for wireless LANs, related papers include distributed fair scheduling (DFS) [18], distributed weighted fair queuing (DWFQ) [4], distributed deficit round robin (DDRR) [17], and distributed elastic round robin (DERR) [9]. DFS achieves fairness through adjusting the backoff interval, while DWFQ adjusts the contention window to provide fairness. Unlike DFS and DWFQ, DDRR and DERR use a mapping function of the inter frame space (IFS) to avoid backoff and achieve fairness with better performance than DFS and DWFQ with the aid of fine discrimination of IFS intervals. However, algorithms in [4], [9], [17], [18] are not applicable for IEEE 802.11e since they do not (properly) consider the QoS issue. Towards designing a fair scheduling algorithm for IEEE 802.11e, we propose a new fair scheduling mechanism called EEDCF in this paper. EEDCF employs some ideas of ERR [11] (so does DERR) so that it is able to dynamically adjust system parameters according to different priority settings to achieve fair scheduling. Moreover, it improves system performance by taking the network status into consideration to properly make each backoff interval disperse so that possible collisions can be further avoided as compared to EDCF which always sets the contention window (CW) to the minimum value regardless of the network status after a successful transmission. In short, EEDCF not only offers better fairness but also improves transmission efficiency as compared to EDCF.

The organization of the remaining paper is arranged as follows. In Section II, the detailed design of EEDCF is described. Then, numerical examples and discussions are given in Section III. Finally, Section IV concludes the paper.

2: DESCRIPTION OF EEDCF

The EEDCF mechanism is composed of two parts. The first part concerns the backoff interval adjustment after a successful transmission. The adjustment is based on the *allowance, excess,* and the total amount of data sent. The second part is the backoff interval adjustment once a collision occurs. These two parts are described as follows.

2.1: Backoff Interval Adjustment after a Successful Transmission

Due to various QoS requirements (see Table I), we have to calculate individually the backoff interval for each queue with a different priority after a successful transmission. Similar to DERR [9] which is a collision-free scheme, *allowance,* which is the amount allowed to transmit data for a queue, is elastic

TABLE I
PRIORITY TO ACCESS CATEGORY MAPPING

Priority	Access Category (AC)	Designation (Information)
1	0	Best Effort
2	0	Best Effort
0	0	Best Effort
3	1	Video Probe
4	2	Video
5	2	Video
6	3	Video
7	3	Video

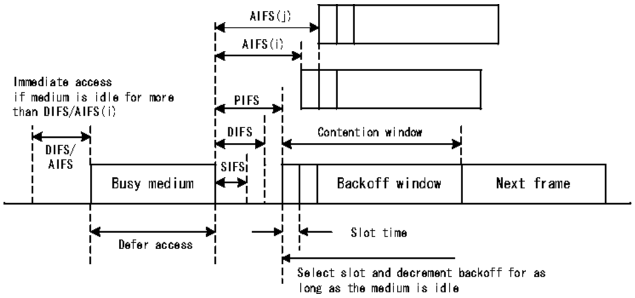


Fig. 2. The IFS relationship of EDCF.

TABLE II
TRAFFIC TYPES AND THE CORRESPONDING CHARACTERISTICS

Type	Inter-arrival Time (Avg. in sec.)	Frame Size (bytes)	Data Rate (Mbps)
Voice	Constant (0.02)	92	0.0368
Video	Constant (0.001)	1462	1.4
Data	Exponential (0.012)	1500	1.0

and adjustable. Data in a queue is allowed to be continuously transmitted and the amount transmitted can be a bit more than the *allowance* after getting the media. The extra amount of data more than the allowance is then called *excess.* Let $E_i(t')$, $A_i(t')$, and $F_i(t')$ represent the *excess, allowance,* and the total amount of data sent, respectively at time t' for queue *i.* Then, they have the following relation:

$$E_i(t') = F_i(t') - A_i(t'). \qquad (1)$$

After a frame is successfully transmitted, the next frame in the queue can be successively transmitted. The process continues until the total length of transmitted frames is more than the *allowance.* In other words, the acquired services of a queue can exceed its origin allotment. Moreover, the *allowance* is not constant and its value can be determined according to two intervals of frame transmission. Let each queue have an individual value of T_i (its reciprocal is related to *weight* [9] (Note that we need to set higher weights according to the order: voice, video, and data in EEDCF) and the ratio of $E_i(t')$ and T_i, i.e., $E_i(t') / T_i$, directly proportion to the *desired throughput.* At time t ($t > t'$), the *allowance* is calculated as follows:

$$A_i(t) = \frac{E_i(t')}{T_i}(t - t') - E_i(t'). \qquad (2)$$

In (2), allowance is increased proportionally to the desired throughput with deduction of the excess amount. Note that a larger waiting time causes a larger *allowance.* Hence, no starvation exists for EEDCF. Moreover, deducting *excess* from the accumulated *allowance* achieves better resource sharing.

We now relate the *allowance* to the backoff interval using the following relations. First, we calculate the backoff interval mainly based on allowance by

$$BI_i(t) = \max_i CW[i] - \varphi A_i(t), \qquad (3)$$

where φ is a constant used to make the interval $BI_i(t)$ fall within the specification of EDCF (also see Fig. 2). The above relation says that a larger allowance may result in a shorter backoff interval. To further consider the collision situation, let us define *collision rate* k_i for queue *i*, which is the ratio of the number of collided packets sent by queue *i* and the number of packets sent by queue *i.* Periodically calculating k_i and using $BI_i(t)$ obtained by (3), the desired backoff interval is got as follows:

$$BI_i(t) \Leftarrow \max(0.2, 1 - k_i) BI_i(t) \qquad (4)$$

in which the term max(0.2, $1-k_i$) tries to make the backoff interval a bit shorter (but not very short, e.g., 0) when the collision rate gets high. By doing so, collisions can be alleviated quickly.

2.2: Backoff Interval Adjustment after a Collision

The backoff interval is set according to (5) when a collision occurs.

$$BI_{new}[j] = BI_{old}[j] rand(PF[j-1], PF[j]), \quad (5)$$

where index *j* represents the priority class and *P F* standards for the persistence factor. Using a random value between the current priority class and the former priority class is to disperse the *P F* value to avoid consecutive collisions for the same priority class.

Thus, the proposed EEDCF takes advantages of ERR to avoid starvation for the low priority queue with consideration of system status. EEDCF will reduce collision and improve throughput and delay etc. Moreover, better fairness is achieved.

3: NUMERICAL EXAMPLES AND DISCUSSIONS

3.1: Simulation Arrangements

The simulation is done under an ad-hoc mode with transmission rate of 11 Mbps. Three different data types, including voice, video, and data are considered (we assume video and voice are generated according to constant bit rate (CBR) traffic). In Table II, traffic types with corresponding characteristics are listed. Since video and voice are delay-sensitive but data is loss-sensitive, buffer sizes for video and voice are fixed at 20 kbits and 1 Mbits, respectively. For data, an infinite buffer is assumed. In Table III, general parameters for various types of traffic, including, priority and AC in the following simulation are given if no re-definition is claimed. Without loss of generality, each station only has one traffic type. In EDCF, no concept of weight is defined. But in EEDCF, we use the relation $T_i = 10^{-4}/w_i$ (sec.) to define weight w_i for data stream i (or queue i). For the comparison purpose, EDCF also uses the same definition of weight.

TABLE III
GENERAL TRAFFIC PARAMETERS FOR SIMULATIONS

Type	Priority	AC	PF	AIFS	CW_{min}	CW_{max}
Voice	7	3	2	PIFS	7	15
Video	5	2	4	PIFS	15	31
Data	0	0	6	DIFS	31	1023

3.2: Performance Metrics

To gauge the performance of different mechanisms, the following metrics are used: 1) *throughput*, 2) *delay*, 3) *dropping probability*, 4) *utilization* (which is defined as $\frac{TotalTxTime - CollisionTime - IdleTime}{TotalTxTime} \times 100\%$, 5) *collision rate*, and 6) *standard deviation for rations of throughput and weight* which can reflect fairness using the reciprocal of the standard deviation.

3.3: Simulation Results

First, 4 voice stations, 2 video stations, and 4 data stations are assumed. Shown in Fig. 3 is the throughput vs. time. Compared to EDCF, 8% improvement for video and voice and about 6% improvement for data stations are obtained when EEDCF is employed. As for delay vs. time shown in Fig. 4, about 7% and 8% lower than EDCF are gained by EEDCF when transmitting voice and video, respectively. Since data employs an infinite buffer, the delay may go very high and is omitted in this figure.

Next, let us observe the impact when the traffic load gets high. Initially, we set 2 voice stations, 1 video stations, and 2 data stations (a total of 5 stations). Afterwards, the number of stations is proportionally increased. Hence, we observe the results ranging from 5 stations to 40 stations. As the number of stations increases, delay shown in Fig. 5 also increases for both EDCF and EEDCF. However, EEDCF increases more smoothly than EDCF when the number of stations increases. Fixing the number of stations at 40, EEDCF gets about 10% improvement. As for the packet dropping probability shown in Fig. 6, EEDCF gets 11% lower than EDCF when the number of stations is 40. In Fig. 7, utilizations for both EEDCF and EDCF increase when the number of stations increases. When the number of stations reaches 15, saturation phenomenon is observed because of collisions. When the number of stations further increases, the utilization can be maintained for EEDCF, while it decreases for EDCF. In Fig. 8, collision rates are shown. Of course, more stations cause a higher collision rate. Again, the increasing trend for EEDCF is smoother than that of EDCF. When the number of stations is 40, 7% improvement is obtained by EEDCF. Considering four different classes of traffic shown in Table IV, Figs. 9(a) – (b) exhibit ratios of different classes for EEDCF and EDCF, respectively. Using the reciprocal of the standard deviation to stand for the degree of fairness [9], we show that about 62% improvement is gained by EEDCF as compared to EDCF. This demonstrates that EECDCF exhibits excellent fairness as compared to EDCF.

4: CONCLUSIONS

A fair scheduling mechanism for IEEE 802.11e, i.e., EEDCF, is proposed and studied in this paper. EEDCF adopts advantages of ERR to enhance fairness and considers different QoS requirements as well as network status, i.e., collision rate, to improve system performance. Through simulations, we show that the improvement in performance and improvement in fairness for EEDCF are about 6%–11% and more than 60%, respectively. Obviously, the above results illustrate that EEDCF outperforms EDCF in terms of transmission efficiency and fairness.

TABLE IV
PARAMETERS FOR OBTAINING RATIONS OF THROUGHPUT AND WEIGHTFOR EEDCF AND EDCF

Weight	Priority	AIFS	CW_{min}	CW_{max}	Rate	Frame Size
0.9	7	PIFS	2^4-1	2^3-1	8 kbps	160 Bytes
0.05	4	PIFS	2^4-1	2^5-1	128 kbps	1280 Bytes
0.03	1	DIFS	2^5-1	$2^{10}-1$	120 kbps	1500 Bytes
0.02	0	DIFS	2^5-1	$2^{10}-1$	96 kbps	1500 Bytes

ACKNOWLEDGEMENT

The first author thanks the partial support in finance by the National Science Council, Taiwan under Contract NSC 95-2221-E-011-029.

REFERENCES

[1] I. Aad and C. Castelluccia, “Differentiation mechanisms for IEEE 802.11,” in *Proc. IEEE INFOCOM ’01,* pp. 209–218, April 2001.

[2] A. Almquist, A. Lindgren, and O. Schelen, “Evaluation of quality of service schemes for IEEE 802.11 Wireless LANs,” in *Proc. LCN ’01,* pp. 348–351, 2001.

[3] O. Altinatas, Y. Atsumi, and T. Yoshida, “A packet scheduling discipline for supporting real-time applications,” in *Proc. IEEE PACRIM ’97,* pp. 895–899, Aug. 1997.

[4] A. Banchs and X. Perez, “Distributed weighted fair queuing in 802.11 wireless LAN,” in *Proc. IEEE ICC ’02,* pp. 3121–3127, May, 2002.

[5] J. C. R. Bennett and H. Zhang, “WF2Q: Worst-case fair weighted fair queueing,” in *Proc. IEEE INFOCOM ’96,* pp. 120–128, Jun. 1996.

[6] K. C. Chen, “Medium access control of wireless LANs for mobile computing,” *IEEE Network,* vol. 8, no. 5, pp. 50–63, September 1994.

[7] A. Demers, S. Keshav, and S. Shenkar, “Analysis and simulation of a fair queueing algorithm,” in *Proc. ACM SIGCOMM ’89,* pp. 1–12, Sep. 1989.

[8] J. Deng and R. S. Chang, “A priority scheme for IEEE 802.11 DCF access method,” *IEICE Trans. on Communications,* vol. 82-B, no. 1, pp. 96–102, January 1999.

[9] H. W. Ferng, C. F. Lee, J. J. Huang, and G. M. Chiu, “Designing a fair scheduling mechanism for IEEE 802.11 wireless LANs,” *IEEE Commun. Lett.,* vol. 9, no. 4, pp. 301-303, April 2005.

[10] IEEE WG, “802.11e draft 3.1,” May 2002.

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

[12] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, “Weighted round robin cell multiplexing in a general-purpose ATM switch chip,” *IEEE JSAC*, vol.9, no. 8, pp. 1265–1279, Oct. 1991.

[13] T. G. Kwon, S. H. Lee, and J. K. Rho, “Scheduling algorithm for real time burst traffic using dynamic Weighted round robin,” in *Proc. IEEE ISCAS ’98,* pp. 506–509, Jun. 1998.

[14] K. Liu, D. Petr, V. Forst, H. Zhu, C. Braun, and W. Edward, “Design and nalysis of a band-width management of framework of ATM-based boadband ISDN,” *IEEE Comm. Magazine*, pp. 138–145, May 1997.

[15] S. Mangold, “IEEE 802.11e - Coexistence of overlapping basic service sets,” in *Proc. of the Mobile Venue ’02,* pp. 131–135, May 2002.

[16] N. Matsufuru and R. Aibara, “Efficient fair queueing for ATM networks using uniform round robin,” in *Proc. IEEE INFOCOM ’99,* pp. 389–397, Mar. 1999.

[17] 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.

[18] 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.

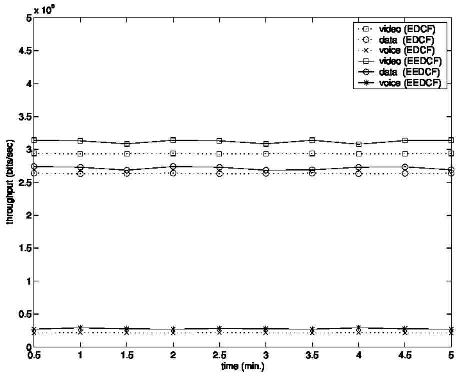


Fig. 3. Throughput vs. time between EEDCF and EDCF.

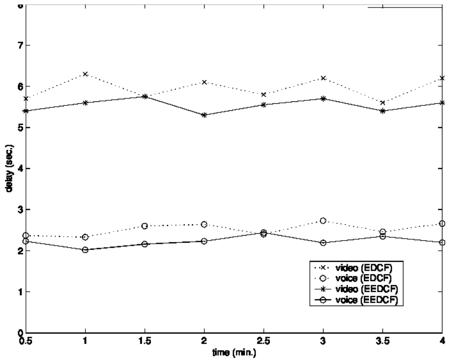


Fig. 4. Delay vs. time between EEDCF and EDCF.

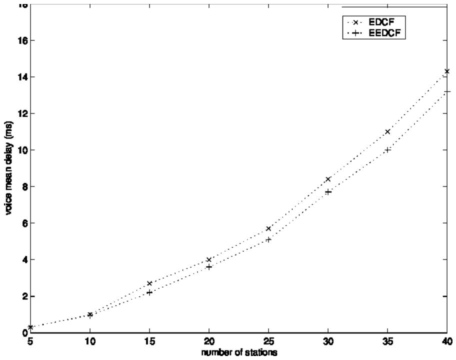


Fig. 5. Delay comparison between EEDCF and EDCF.

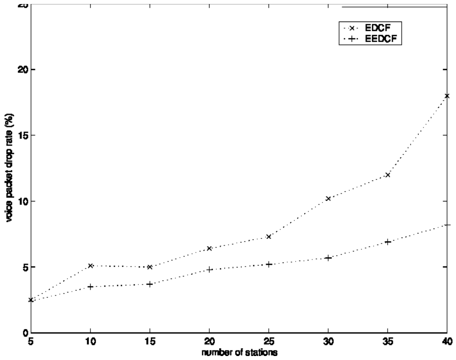


Fig. 6. Dropping probability comparison between EEDCF and EDCF.

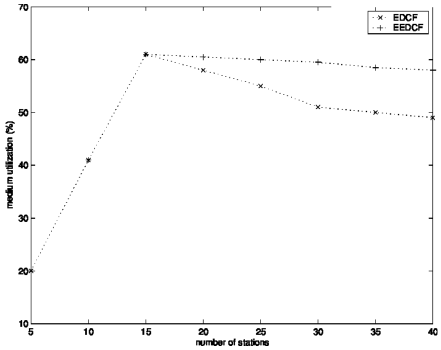


Fig. 7. Medium utilization comparison between EEDCF and EDCF.

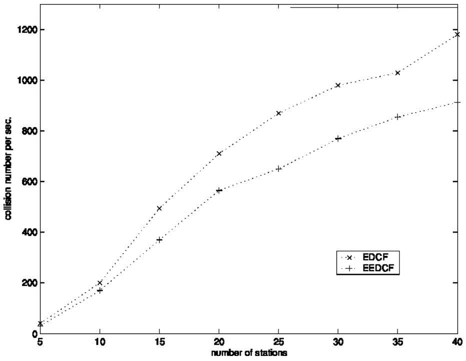
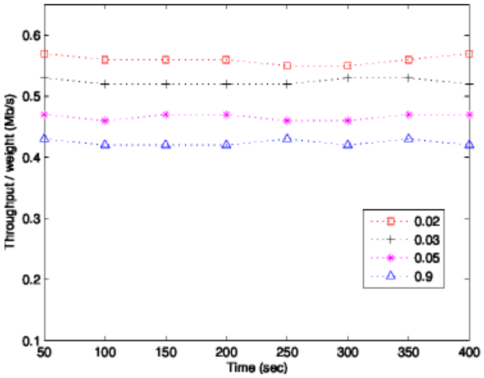
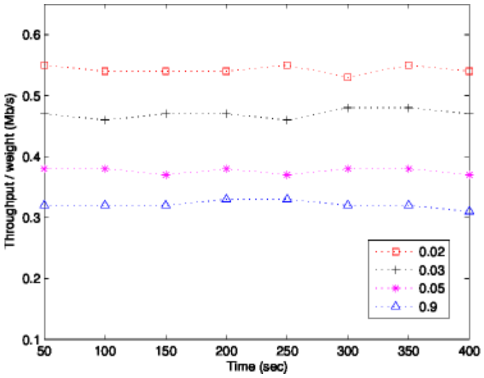


Fig. 8. Collision rate comparison between EEDCF and EDCF.



(a) EEDCF.



(b) EDCF.

Fig. 9. Ratios of throughput and weight for EEDCF and EDCF under four different weights.