Priority Collision Resolution - Distributed Coordination Function for Distributed Wireless Networks

Xiaohui Ye*, Xin Liu#, S. J. Ben Yoo*, and Zhi Ding*
* Department of Electrical and Computer Engineering, University of California, Davis
# Department of Computer Science, University of California, Davis
Davis, California, 95616 USA
{xye, xinliu, sbyoo, zding}@ucdavis.edu

Abstract— **In distributed wireless access networks, the short-term unfairness of IEEE 802.11 Distributed Coordination Function (DCF) has been revealed by many works. In this paper, a modified DCF, based on the principle of Priority Collision Resolution (PCR), is proposed to improve the short-term fairness in distributed access wireless networks. Our PCR-DCF achieves short-term fairness improvement without estimating behaviors of other users and the system contention level, such as the number of active users. Only the capability to identify collision is required. Theoretical analysis is carried out to validate the performance of PCR-DCF based on a slotted system model. Both simulations and analyses show that PCR-DCF has a much smaller packet transmission delay jitter than IEEE 802.11 DCF, with little degradation on the average packet transmission delay. Moreover, the analysis and simulation results indicate that PCR-DCF also experiences a much lower packet drop rate than IEEE 802.11 DCF in almost all load ranges with reasonable retransmission limit.**

Keywords-short-term fairness; priority collision resolution; distributed; wireless network

I. INTRODUCTION

IEEE 802.11 [1] Distributed Coordination Function is the most prevalent mode today in popular Wireless Local Area Networks (WLANs). Since the sender is unable to detect collision while transmitting, CSMA [2] and Binary Exponential Backoff (BEB) are adopted to enable stations to access the channel and resolve collision in a distributed manner. Before transmission, the station has to sense the channel state. The transmission starts only after the channel has been idle for DIFS (DCF Interframe Space) and the station cuts down a random number of idle slots. For each transmission attempt, the random number referred to as the backoff timer is selected uniformly from [0, CW-1], where CW represents its current Contention Window. CW is reset to CW_{min} after each successful transmission or after discarding a packet due to reaching its maximum retransmission limit. CW is doubled after each collision until reaching CW_{max}.

Generally speaking, IEEE 802.11 DCF is a fair sharing protocol, since each station adopts the same behavior and has an equal chance to access the channel. In long-term, each user with the same physical layer capability (e.g. frame error rate) tends to have the similar throughput. But existing works have shown that 802.11 DCF suffers from the short-term fairness problem [3-5]. The problem is caused by many factors [6]. If we only consider the scenario without hidden stations, in other words, if all stations can hear from each other, then the short-term unfairness is mainly due to the characteristics of BEB used in IEEE 802.11 DCF [5]. In short-term, some stations cannot access the channel because of successive collisions, and some other stations will always occupy the channel because their CWs are luckily kept small after successful transmissions. Apparently, the inherent mechanism embedded in BEB favors successful stations, which causes the unfairness problem.

Short-term fairness can be improved if each station monitors the channel and adjusts its behavior. In [5], it is proposed that all stations adopt optimized CW_{min} without using BEB. However, tuning CW_{min} to the optimal value requires the knowledge about the number of active users in the system and the waste of channel [5].

In [6] Adaptive Transmission Control (ATC) again requires the sender to first estimate the number of active stations within the contention range and the actual bandwidth acquired by the station. Based on the knowledge from estimation and the knowledge conveyed by the receiver, the sender enters one of three modes: aggressive, restrictive, and normal, thereby tuning the CW accordingly. Upon entering aggressive mode while the backoff timer is non-zero, the backoff timer will be regenerated based on the new CW.

Information exchange among stations can also be used to improve short-term fairness. In [7], Randomly Ranked Mini Slots (RRMS) and RRMS-Busy Tone (RRMS-BT) employ a pseudo random sequence for all stations. Stations exchange their seeds with one other, and thus in each slot each station can determine ranks of all stations which are generated from the sequence and seeds. In each slot, the station with the highest rank in its interference range will transmit.

Compared with the above approaches, our new Priority Collision Resolution-DCF (PCR-DCF) adopts a different approach where collision is resolved with high priority. The proposed PCR-DCF has the following advantages. It does not

---

This material is based upon work supported in part by the National Science Foundation under Grant No. 0520126.

need to estimate the number of active users in the system. Also stations do not need to exchange information with others to estimate their behavior. Furthermore, short-term fairness is significantly improved in PCR-DCF without optimizing or tuning CW_{min}. The only requirement for implementation is that stations need to identify collisions, which is not required by IEEE 802.11 DCF. The complexity of PCR-DCF will mainly arise from the task of differentiating collisions from other transmissions. Like IEEE 802.11 DCF, PCR-DCF is fully distributed and scalable, and the algorithm complexity does not increase while the number of user increases.

The remainder of this paper is organized as follows. In section 2, we first describe the basics of PCR-DCF. Next, in section 3, we present an analytical model to analyze performance of PCR-DCF. We provide simulation results and comparisons with the legacy IEEE 802.11 DCF in section 4 and conclusions in section 5.

II. PRIORITY COLLISION RESOLUTION - DCF

Since BEB used in IEEE 802.11 DCF is the main source of short-term unfairness, it will no longer be used in our scheme. To replace BEB, a new collision resolution scheme needs to be designed. By observing that the number of participants of most collisions is generally small, if we can resolve the collision first, the packet transmission delay of the stations with collisions will be greatly reduced. Although transmission of other stations without collision is delayed, the impairment will be little since most collisions will be resolved quickly.

The key mechanism of PCR-DCF can be described as follows. When a station senses a collision and its backoff timer is non-zero, it increases its backoff timer by a pre-defined deferment parameter "DEFER"; when the station transmits a packet and does not receive the corresponding ACK, it randomly selects a backoff timer from the range [0, DEFER] to schedule a retransmission. After each successful transmission, the station will randomly choose a backoff timer value from the range [0, CW_{min}]. Backoff timer reduction follows the same way as the legacy IEEE 802.11 DCF, and transmission starts if the backoff timer reaches zero.

By deferring stations without collision, a period of DEFER slots is reserved to resolve the current collision. In the period of DEFER slots, the collision probability for potential senders will be reduced since fewer stations are able to transmit during this particular period. The stations with collision have a much higher probability of making a successful retransmission in a short period. Overall, the mechanism of priority collision resolution enables the station to spend much shorter time to wait for retransmission, thus helping reduce packet transmission delay variation. On the other hand, although consecutive collisions may still be possible in PCR-DCF, their occurrence is much less likely. The associated side effect is that the packet drop rate is much smaller compared to IEEE 802.11 DCF. The detailed analysis is shown in the next section.

As discussed in section I, PCR-DCF still relies on CSMA scheme. To implement PCR-DCF, the station needs to be able to detect all ACKs sent from others. Although due to noise and interference, the station may not decode all ACKs, if we do not consider the hidden station problem, ACKs transmitted at the fixed rate will be easily recognized by all stations since they have fixed length and conformed format.

Like IEEE 802.11 DCF, three parameters are used in PCR-DCF: CW_{min}, DEFER, and the retransmission limit (RETRAN). Selecting proper values for DERER and RETRAN will further help improve the performance. If we choose a small value for DEFER, in the DEFER slots, the previously collided stations still have a high collision probability; if we choose a large value for DEFER, the packet transmission delay increases and the system throughput degrades.

Fig. 1 shows an example of PCR-DCF. There are four stations in the system: S1, S2, S3 and S4. CW_{min} and DEFER are set to 8 and 7 respectively. Each line shows the backoff timer value of each station. When S2 and S3 collide, the backoff timer of S1 and S4 is increased by 7, while S2 and S3 generate new backoff timer randomly from the range [0, 7].

III. PCR-DCF PERFORMANCE ANALYSES

To analyze PCR-DCF performance, we develop a slotted system model to characterize its operation. A basic slot is defined as time interval between two consecutive backoff timer changing events, either decreasing or increasing. The slot length can be very short, if it is an idle slot, or very long, if it is a transmission slot. Basically, there are two types of transmission slots in the system, collision slots and success slots. A collision slot consists of packet transmission and the subsequent DIFS; a success slot consists of the packet transmission, the ensuing SIFS (Short Interframe Space), the ACK transmission, and the DIFS. For ease of analysis, we define the extended collision slot as a group of successive slots, which starts with a collision slot, followed by idle slots, success slots, and possibly extended collision slots. The number of basic slots contained in one extended slot equals 2+DEFER. Fig. 2 shows an example to illustrate the slotted system model and the concept of the extended collision slot. In this example, we set DEFER to 7.

If we zoom in the first extended slot in the top layer, we can find that the extended slot starts with a collision caused by three stations transmitting simultaneously, and the collision is followed by one success slot, 6 idle slots, and one extended collision slot since two stations collide again. Further zooming in the extended collision slot at the second layer, we will see that it starts with a collision, and followed by 2 success slots and 6 idle slots. By introducing the concept of extended collision slot, the collision and the subsequent collision resolution can be considered as a whole event.

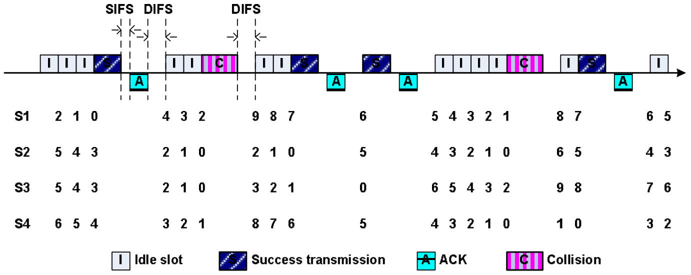

Figure 1. An example of PCR-DCF operation

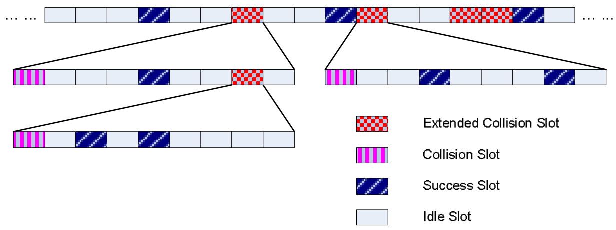

Figure 2. An example of slotted PCR-DCF

For further convenience, the *extended slot* refers to a top layer slot, either a top layer idle slot, or a top layer success slot or a top layer extended collision slot. After defining the extended slot, the system can be treated as the basic scheme of IEEE 802.11 DCF without exponential backoff stages, where the station directly discards the packet without executing BEB after a collision. Thus the former analysis derived for IEEE 802.11 DCF basic scheme can be applied to PCR-DCF with the backoff stage parameter set to 0 [8]. The difference is that the collision slot in 802.11 DCF analysis is replaced by the *extended collision* slot in our system.

A. *Collision probability analysis*

Assume the system operates in the overloaded range. Each station continuously has packets to send. Also assume each station operates independently. Following the analysis in [8] and considering backoff stage parameter to be 0, the probability that a station transmits in a randomly chosen slot is

$$\tau = 1/(CW_{\min} + 1), \quad CW_{\min} = 2^m . \tag{1}$$

In a system that has N active stations, for a randomly chosen slot, the probability for the slot to be an idle slot is

$$P(idle) = C_N^0 (1-\tau)^N , \tag{2}$$

where C_N^k is defined as $N!/(N-k)!/k!$. The probability that in the slot there is a successful transmission is

$$P(suc) = C_N^1 (1-\tau)^{N-1} \cdot \tau . \tag{3}$$

And the probability that in the slot m $(2 \le m \le N)$ stations collide together is

$$P(m-col) = C_N^m (1-\tau)^{N-m} \cdot \tau^m . \tag{4}$$

Table I shows the parameters used for simulation and analysis in this section. The value of DEFER can be chosen from the set $\{2^j - 1 \mid 1 \le j \le 10\}$ and the value of RETRAN should be no less than zero. Table II shows the activity (excluding idle) probability distribution of the system. The results are obtained from both simulation and analysis.

TABLE I. Simulation and analysis parameters

	802.11 DCF	PCR-CDF
CW_{min}	32	
CW_{max}	1024	-
DEFER	-	15
RETRAN	5	3
Data rate	2 Mbps	
Payload length	500 Bytes	
PHY header	192 usec	
MAC header	34 Bytes	
ACK	248 usec	
DIFS	50 usec	
SIFS	10 usec	
Slot Time	20 usec	
Active user number	10	

TABLE II. Activity probability distribution of the system (exclude idle events) when using PCR-DCF

	Simulation	Analysis
Success	0.7427	0.74274
2-station-collision	0.21565	0.21563
3-station-collision	0.037108	0.037098
4-station-collision	0.0041984	0.037098
5-station-collision	0.00032261	0.00032427
6-station-collision	1.7133e-005	1.7434e-005
7-station-collision	5.3127e-007	6.4273e-007

If we have a particular station in mind, when the observed station has a non-zero backoff timer, the system then can be viewed as a subsystem that contains *N-1* stations. Directly from (2)-(4), the conditional probability distribution can be obtained. When the observed station transmits, starting from (2)-(4) we will have the conditional probability that the observed station makes a successful transmission equal to

$$P(suc \mid transmit) = C_{N-1}^0 (1-\tau)^{N-1} . \tag{5}$$

And the conditional probability that m $(2 \le m \le N)$ stations collide together is

$$P(m-col \mid transmit) = C_{N-1}^{m-1} (1-\tau)^{N-m} \cdot \tau^{m-1} . \tag{6}$$

Later we will use the non-conditional probability distribution (2)-(4) to derive the packet drop rate and the system throughput. The conditional probability distributions will be used to derive the throughput and the packet transmission delay of the observed user.

B. *Collision resolution phase analysis*

When several stations collide, the system enters the collision resolution phase. For example, in Fig. 2, the first extended collision slot starts with a 3-station-collision. In the first collision resolution stage, one station makes a successful transmission. The other two stations collide again in the first collision resolution stage and enter the second collision resolution stage. For the second extended collision slot, a 2-station-collision is resolved in the first collision resolution stage. Similar to IEEE 802.11 DCF, a maximum collision resolution stage, or called the retransmission limit, is defined in PCR-DCF. The packet is eventually dropped if the collision remains unresolved at the end of the maximum collision resolution stage.

Following the definition of the extended collision slot, each collision and its corresponding collision resolution phase can be treated as an extended collision slot. Given a collision starting with m-station-collision, the deferring window size DEFER, and the maximum collision resolution stage K, the conditional dropping probability can be derived. For example, when m=2 and K=2, the conditional drop probability will be that the two stations collide again at each collision resolution stage, which is

$$P_{drop}(m=2, K=2) = 1/(DEFER+1)^2 .$$

Combining the activity probability distribution obtained from equation (1)-(3), the packet drop probability of the system is

$$P_{pkt-drop} = \sum_{m=2}^{N} P_{drop}(m,K) \cdot P(m-col) . \tag{7}$$

For an environment with parameters shown in Table 1, the packet drop probability of PCR-DCF calculated from analysis is $1.3498\text{x}10^{-4}$, while the result from simulation is $1.3438\text{x}10^{-4}$, matching our analysis. From the simulation, the packet drop rate of IEEE 802.11 DCF is $6.5604\text{x}10^{-4}$, which is more than four times the packet drop rate of PCR-DCF. Although in PCR-DCF fewer retransmissions are tried in comparison to IEEE 802.11 DCF, stations experiencing collision will temporarily have a less “crowded” environment to resolve the collision immediately after it happens.

C. System throughput and user throughput

For one extended slot, from (2)-(4), we can obtain the extended slot length distribution. The Cumulative Distribution Function (CDF) of the slot length from both analysis and simulation are shown in Fig. 3. From the slot length distribution, we are able to calculate the average slot length ($\overline{T}_{slot}$). The distribution of the number of bits transmitted in one extended slot can be obtained from activity probability distribution (2)-(4) and collision resolution analysis. Then we have the average bits ($\overline{B}$) sent in one extended slot. The average throughput of the system can be derived from

$$\overline{Th}_{system} = \frac{\overline{B}}{\overline{T}_{slot}} . \tag{8}$$

For an environment with parameters of PCR-DCF shown in Table 1, from the analysis, the average slot length is 1955.9 µs and the average bits sent during one slot is 2423.9 bits, thus the average system throughput is 1.2393 Mbps. By comparison, the average system throughput collected from simulation is 1.2402 Mbps, which is very close to the analytical results derived.

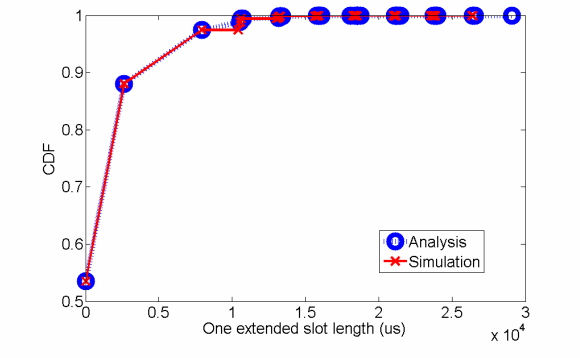

Figure 3. The CDF of the extended slot length using PCR-DCF

Next we will derive the average packet transmission delay and the average user throughput. The packet transmission delay here consists of two parts: the time spent on the first transmission and possible retransmissions, and the time spent on decreasing backoff timer before each transmission. From the conditional activity probability distribution, we can obtain the conditional extended slot length distribution under the condition that the backoff timer of the observed station is non-zero and when the observed station starts transmission. Notice that after each successful transmission, the station will randomly select a value from the range [0, CW_{min}] as the backoff timer for the next transmission. Thus, the average value for a newly generated backoff timer is $(CW_{min}-1)/2$. We can approximate the average packet transmission delay for one user whether the packet is finally sent out or dropped from

$$\overline{T}_{delay} = \frac{CW_{min}-1}{2} \cdot \overline{T}_{cond-ntr} + \overline{T}_{cond-tr} , \tag{9}$$

where $\overline{T}_{cond-ntr}$ represents the average extended slot length when the backoff timer of the observed station is non-zero, and $\overline{T}_{cond-tr}$ represents the average extended slot length when the observed station transmits. Thus, the average throughput of one particular user is obtained from

$$\overline{Th}_{user} = \frac{Pkt}{\overline{T}_{delay}}(1-P_{pkt-drop}) , \tag{10}$$

where Pkt is the average payload bits contained in a packet. In this work, this value is fixed to 4000.

The analytical results show that the average packet transmission delay is 32273 µs for the observed user and the throughput of the observed user is 0.12392 Mbps. From simulation, the average packet transmission delay is 32248 µs and the throughput for the observed user is 0.12416 Mbps. In summary, the analysis is fully corroborated by our simulation results. Notice that in PCR-DCF the throughput of the observed user multiplied by the number of active users is almost the system throughput; thus achieving throughput fairness at least in long-term.

D. *Packet transmission delay analysis*

The packet transmission delay is the total time spent on transmitting a particular packet. In PCR-DCF, the delay can be expressed as follows

$$T_{\text{delay}} = \sum_{i=0}^{bt} x_i + x_t , \tag{11}$$

where b_t is a random variable with uniform distribution on [0, $CW_{\min}$], x_i is a random variable representing the time length of the extended slot in which the observed station does not transmit, and x_t represents the random time length of the extended slot in which the observed station starts transmission and is possibly followed by collisions resolution.

We assume each x_i to be i.i.d to obtain approximate analytical results. We found the average packet transmission delay to be 32198 µs. The result is consistent with what we have derived in Part C. Fig. 4 shows the cumulative distribution function of the packet transmission delay for IEEE 802.11 DCF (simulation), PCR-DCF (simulation), and PCR-DCF (analysis). We observe little difference between PCR-DCF (simulation) and PCR-DCF (analysis). The reason is that each x_i is not truly i.i.d and the slot length distribution of the latter one depends on the slot length distribution of its previous slots.

IV. PERFORMANCE EVALUATIONS

In this section, we compare performance of IEEE 802.11 DCF and PCR-DCF. All results shown in this section are collected from simulation. By varying the number of users, DEFER, and RETRAN, while keeping all other parameters in Table 1 unchanged, we obtained Figures 5-9. In these figures, D denotes DEFER and R denotes RETRAN. Fig 5 shows that the average packet transmission delay is almost the same for both DCF schemes. However, the standard deviation of the delay for PCR-DCF is much smaller as shown in Fig. 6. In fact, the difference grows as the number of users increases. Delay jitter or variation is an indicator of short-term fairness. We achieve significant short-term fairness improvement with only small degradation in other performance measures.

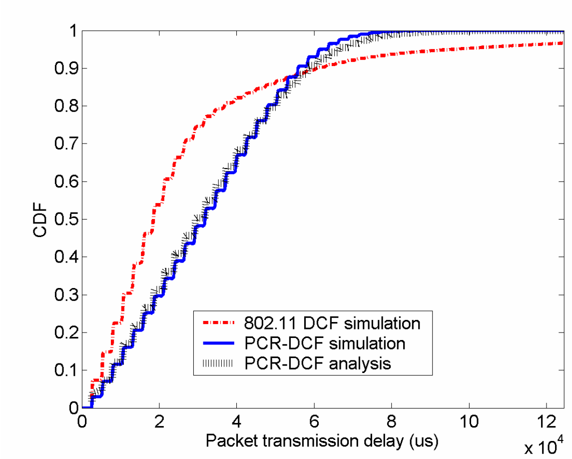

Figure 4. Packet transmission delay CDF comparison.

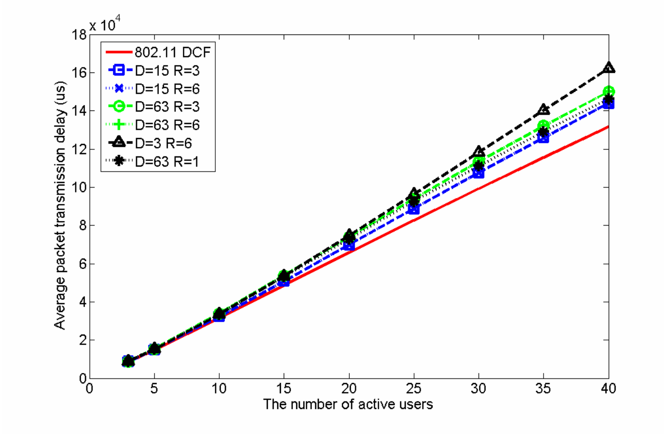

Figure 5. Average packet transmission delay versus the number of users

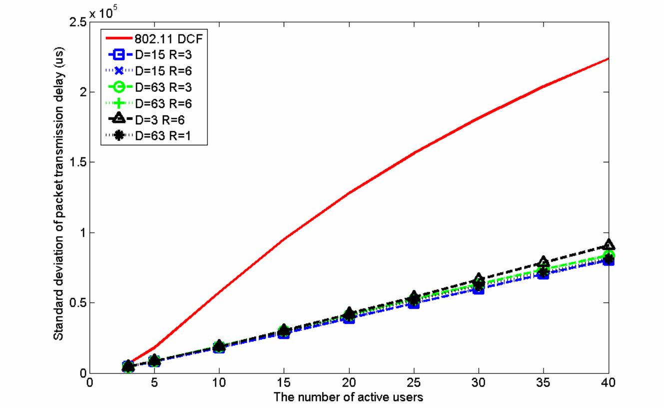

Figure 6. Standard deviation of packet transmission delay versus the number of users

Fig. 7 evaluates the fairness among users. We use Jain's fairness index [9], defined as

$$F(X) = \frac{\left(\sum x_i\right)^2}{n \cdot \sum x_i^2} , \tag{12}$$

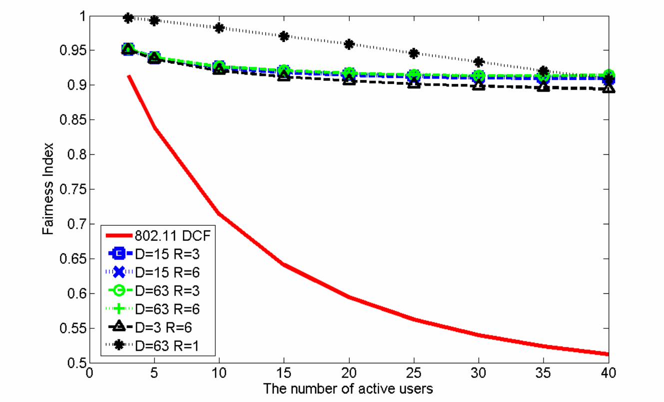

Figure 7. Fairness Index versus the number of users

in which n is the number of competing users and x_i represents the number of successful transmissions made by the i-th user as we observe activities of all users in a short interval. The length of observation window is measured by the number of transmission slots, described in Section III. Here the ratio of the observing window size to the number of competing users is fixed to be 5. The results of Fig. 7 confirm again that the PCR-DCF achieves significant short-term fairness improvement compared with IEEE 802.11 DCF.

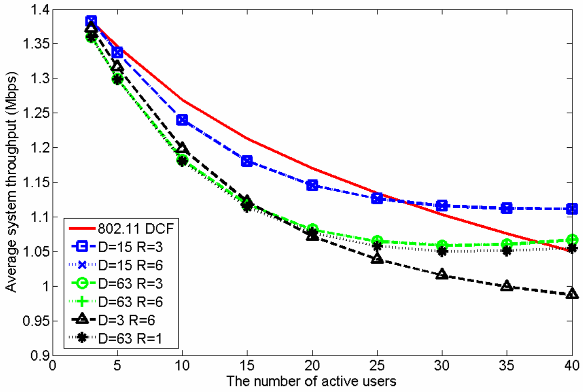

Figure 8. Average system throughput versus the number of users

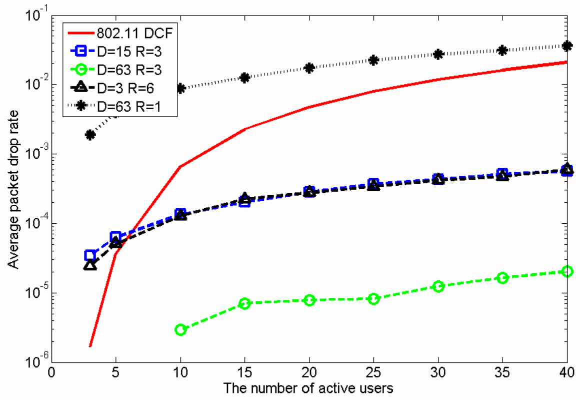

Figure 9. Average packet drop rate versus the number of users

As the average packet transmission delay is similar for the two DCF schemes, it is no wonder that the average system throughput for PCR-DCF and IEEE 802.11 DCF also remains similar as shown in Fig. 8. This further confirms that the short-term fairness is improved with little impairment on other system performance measures. Interestingly, with a large number of users, the system throughput of IEEE 802.11 DCF at some parametric settings is smaller than that of PCR-DCF while the average packet delay of IEEE 802.11 DCF is smaller than that of PCR-DCF. The reason is that the packet drop rate of IEEE 802.11 DCF increases significantly with the increasing number of users, but for PCR-DCF, if we do not choose very small RETRAN, the packet drop rate increases very slowly, as observed in Fig. 9. Furthermore, the packet drop rate for the settings RETRAN=6 and DEFER=15 or 63 is too small (less than 10^{-6}) to be observed in our simulation. It is clear that varying DEFER and RETRAN does not influence performance of short-term fairness. But choosing a large DEFER or a small DEFER degrades the system throughput. This result confirms the discussion in section II. With a small RETRAN, PCR-DCF has a larger packet drop rate. Increasing RETRAN has little effect on system throughput, since collisions become very rare after several retransmissions. In general, compared with IEEE 802.11 DCF, the short-term fairness and packet drop rate are clearly improved in PCR-DCF with little effect on the average packet transmission delay.

V. CONCLUSIONS AND FUTURE WORK

In this work, we propose a novel distributed coordination function known as the Priority Collision Resolution-DCF. Introducing a mechanism to first resolve collisions, the short-term fairness is significantly improved over IEEE 802.11 DCF. The packet drop rate in PCR-DCF grows much slower than that of IEEE 802.11 DCF with increasing number of active users. The system throughput with a large number of users further benefits from the improvement of the packet drop rate. We also developed a slotted system model and carried out theoretical analysis to prove the performance improvement. In future works, we plan to investigate PCR-DCF in scenarios with imperfect wireless channel, hidden terminals, and some stations that cannot always correctly recognize collisions.

REFERENCES

[1] http://standards.ieee.org/getieee802/802.11.html.

[2] L. Kleinrock and F.A. Tobagi, Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access and their Throughput-Delay Characteristics. IEEE Transactions on Communications, Dec 1975. 23(13): p. 1417-1433.

[3] C.E. Koksal, H. Kassab, and H. Balakrishnan., An analysis of short-term fairness in wireless media access protocols, in Proceedings of ACM SIGMETRICS, 2000, June 2000.

[4] Barrett, C.L., et al. Analyzing the short-term fairness of IEEE 802.11 in wireless multi-hop radio networks, in Proceedings of 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002, Oct. 2002.

[5] K. Medepalli and F.A. Tobagi., On Optimization of CSMA/CA based Wireless LANs: Part I - Impact of Exponential Backoff, in Proceedings of IEEE International Conference on Communications, 2006, June 2006.

[6] Z. Li, S. Nandi, and A.K. Gupta., Achieving MAC fairness in wireless ad-hoc networks using adaptive transmission control, in Proceedings of Ninth International Symposium on Computers and Communications, 2004, 28 June-1 July 2004.

[7] J. Eshet and L. Ben, Randomly Ranked Mini Slots for Fair and Efficient Medium Access Control in Ad Hoc Networks, IEEE Transactions on Mobile Computing: Accepted for future publication, 2007.

[8] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications, Mar 2000. 18(3): p. 535-547.

[9] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. New York: Wiley-Interscience, Apr. 1991.