Constrained Priority Countdown Freezing - A Collision Memory Avoidance Algorithm

Ivan Kedzo
University Department of Professional Studies
University of Split
Split, Croatia
e-mail: ikedzo@oss.unist.hr

Julije Ozegovic, Vesna Pekic
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
University of Split
Split, Croatia
e-mail: julije.ozegovic@fesb.hr
e-mail: vesna.pekic@fesb.hr

***Abstract*— Collision memory in IEEE 802.11 Distributed Coordination Function (DCF) has been detected. The collision memory can increase the physical collision rate and this effect is inherent to any DCF type of countdown. In this paper, we introduce a collision memory avoidance algorithm, called Constrained Priority Countdown Freezing (CPCF). The CPCF can completely or partially remove collision memory depending on how many priority freezing steps are allowed. Since DCF's well known countdown decreases the contention overhead, but increases collision memory effect, the solution is to find the compromise between the two, in order to achieve good performance in both low and high load network conditions. The CPCF achieves this by limiting the countdown process, and thus reducing the collision memory, while still producing significant countdown effect.**

Keywords – collision memory; constrained freezing; backoff freezing; DCF countdown; wireless MAC.

I. INTRODUCTION

The basic channel access method for wireless networks with distributed access is IEEE 802.11 Distributed Coordination Function (DCF) [1]. DCF has one distinguished feature called backoff freezing mechanism, which allows count down of priority through multiple Contention Resolution Periods (CRP), where priority is chosen from certain Contention Window (*CW*). Freezing the countdown process after loosing the medium contention is the key component of DCF, which insures shorter contention overhead. This reduction of contention overhead is called countdown effect [2][6].

However, in our previous work [3], we have seen that countdown freezing mechanism has impact on collision rate. In [3], it can be observed that DCF-like countdown protocol called *Binary Priority Countdown – DCF Countdown* (BPC-DC), which is essentially binary countdown version of DCF protocol, exhibits increased collision rate when compared to *Binary Priority Countdown – Decrement After LCI only* (BPC-DAL) protocol, which chooses new priorities before each CRP. In [3], it is concluded that the reason for increased collision rate is collision memory.

Collision memory occurs due to freezing of the priority countdown, which, as a result, preserves priority collisions. If the freezing of the countdown is not constrained, like in DCF, priority collision always leads to physical collision. This can increase physical collision rate and this effect is called collision memory effect.

In this paper, we propose collision memory avoidance algorithm called Constrained Priority Countdown Freezing (CPCF). CPCF protocol puts constrains on freezing mechanism of DCF in order to achieve short contention overhead and low collision rate.

The rest of the paper is organized as follows. In Section II, the related work is presented. Section III explains in details collision memory, while in Section IV we introduce a new collision memory avoidance algorithm called CPCF. Section V verifies proposed protocol using the ns-2.33 simulator and comments on simulations results. In Section VI, we consider future work and conclude.

II. RELATED WORK

Medium Access Control (MAC) protocols which use Priority Number (*PN*) to resolve the medium contention are called priority contention protocols. These protocols schedule competing Stations (STA) regarding their priorities, allowing higher priority competitors to access the medium earlier. Priority number *PN* is chosen from the set of allowed values called Priority Space (*PS*). If a priority contention protocol employs priority countdown, then lower *PN* should indicate higher priority (e.g., IEEE 802.11 DCF protocol).

DCF uses priority number *PN* as the number of consecutive time slots in which STAs have to wait before starting transmission to the medium. DCF requires a STA to calculate Backoff Counter (*BC*), which is essentially a priority number *PN*, after each transmission. *BC* is chosen randomly from the priority space *PS* limited with the *CW*. After the channel is sensed to be idle for a Distributed Inter Frame Space (DIFS) interval, a STA decrements *BC* when the medium is idle in the current time slot, and *BC* is frozen when another STA is transmitting. When *BC* is decremented to zero, STA accesses the medium.

 ISBN: 978-1-61208-203-5

The 802.11 DCF function has been excessively studied. This included different analysis and enhancements in order to explain or fix DCF's drawbacks. A new protocol, called Enhanced DCF (EDCF), was introduced, supporting Quality of Service (QoS), and it became a new standard [1]. Also, various enhancements were proposed to increase throughput or influence fairness or delay [7][8].

The throughput increase is done mainly through *CW* [7] adaptations in order to reduce collisions or contention overhead, while different Inter Frame Space (IFS) values are used to achieve fairness and low delay for different types of traffic [1].

III. COLLISION MEMORY

DCF countdown allows unconstrained priority freezing after losing the medium contention. The priority chosen by the STA that has lost the medium contention is counted down through multiple CRPs, until it reaches the highest priority and either wins the medium access or enters the collision. When two or more STAs choose the same priority in CRP, priority collision occurs. If the priority collision occurs between the highest priorities chosen, the physical collision occurs in the observed CRP. Obviously, in a single contention there can be multiple priority collisions without any physical collision.

Collision memory can be defined as the ability to preserve priority collisions from previous CRPs, which have occurred during the countdown freezing process. Collision memory can increase the physical collision rate and this effect is called collision memory effect. The collision memory effect occurs due to countdown freezing mechanism. After losing the medium contention, STAs decrement their priority numbers *PN* with the winning priority from the current CRP and freeze decremented *PN* values to be used in the next CRP. All STAs that have experienced a priority collision, will remain in priority collision in the next CRP, and can potentially cause physical collisions.

Collision memory can be preserved through one or more CRPs, depending on the protocol's "memory size". For instance, DCF has infinite collision memory since it can theoretically freeze countdown indefinitely. Remembering priority collisions from previous CRPs is a major drawback when combined with countdown mechanism, which increases STA's priority after each CRP. The DCF's unconstrained priority countdown with freezing insures that all priority collisions eventually become collisions with the highest priority and therefore cause physical collisions.

This can be avoided if we constrain the DCF's backoff procedure. In [3], it is shown that protocol without countdown freezing mechanism, can achieve lower collision rate. However, such protocol also shows increased contention overhead due to lack of the DCF's countdown effect. Therefore, deep investigation of DCF's countdown mechanism with freezing is important, if both, low collision rate and short contention overhead, are desired.

IV. COLLISION MEMORY AVOIDANCE ALGORITHM

The reason causing collision memory effect, which can decrease overall throughput, is DCF's unconstrained priority countdown freezing mechanism. Let's consider the following formula:

$$P_{CM}(m) = \sum_{i=1}^{m} p_i \qquad (1)$$

$$m = 1, \ldots, \infty$$

where $P_{CM}(m)$ is the probability that a STA has experienced the priority collision due to collision memory, after it has frozen its priority *m* times. Since collisions are remembered due to freezing of priority, $P_{CM}(m)$ is equal to the sum of *m* probabilities denoted with p_i, where each p_i represents the probability that a STA has experienced the priority collision in *i*-th step of priority freezing. From this formula it is clear that the bigger *m* we have, the higher probability $P_{CM}(m)$ becomes. Obviously, the $P_{CM}(m)$ probability has direct influence on physical collisions. STAs that are in countdown can win the medium contention after several consecutive CRPs, and have higher chance of experiencing the physical collision due to high $P_{CM}(m)$ probability.

Therefore, in order to reduce collision memory effect, the priority freezing can be constrained in a way that we can control how many times can priority be frozen and decremented before choosing a new priority. We call this collision memory avoidance algorithm, a Constrained Priority Countdown Freezing (CPCF).

CPCF STA has a counter called freezing counter and a countdown freezing limit *k*. Parameter *k* defines the maximum number of times we can freeze and countdown priority, before choosing a new priority value. The parameter *k* is used to limit the actual number of countdowns *m* from (1):

$$m = 1, \ldots, k \qquad (2)$$

In (2), *m* is constrained, and the maximum number of CRPs in which a STA can countdown its priority is equal to *k*. Besides partially constraining countdown freezing, CPCF can also completely remove it by not allowing STAs to freeze their priorities. This is done by setting the freezing limit *k* to zero (*k*=0), which forces STAs to choose new random priorities in each CRP. In this case the $P_{CM}(m)$ is equal to zero, since STAs have no memory of priority collisions that occurred in the past.

The algorithm works as follows. In the beginning, all STAs choose their priorities randomly from certain *CW*, and reset their freezing counters to freezing limit *k*. CPCF STAs that have lost the medium contention will decrement freezing counter by one, and decrement their priority with the winning priority from the current CRP, just like in

 ISBN: 978-1-61208-203-5

DCF. When freezing counter becomes zero, a STA must choose a new priority number and reset freezing counter to freezing limit *k*. This way, STAs that have lost the medium contention can countdown their priority at most through *k* consecutive CRPs.

A STA that has won the medium also chooses the new priority number, and resets the freezing counter to *k*. After collision, a STA doubles its *CW*. Initial value is set to CW_{min}, and can be increased until it reaches the CW_{max}. This mechanism is identical to the DCF's Binary Exponential Backoff (BEB) mechanism [4]. Obviously, DCF countdown mechanism is CPCF mechanism with $k=\infty$.

V. SIMULATIONS

The performance of the proposed CPCF protocol is verified using the network simulator ns2, version 2.33. The simulator is upgraded with the CPCF module based on ns2 mac-802_11Ext module [5]. The main performance measure is the network throughput achieved, whereas collision probability graphs are presented for reference. For comparison with CPCF, a basic 802.11 DCF MAC protocol was used. Simulations include verification of CPCF's collision memory avoidance algorithm using different values of *k*.

In the network scenario used, a simple wireless ad-hoc network where all *n* STAs can hear each other is simulated. STAs positions are fixed and chosen at the beginning of the simulation, with STAs randomly choosing coordinates from the predefined area. Each STA with address *a* has one ftp flow (bulk packet transfer) directed towards the STA with address (*a*+1)/modulo *n*. Flows are started gradually, from the beginning of the simulation, every 0.1s. Ftp flow is carried over tcp (tcp receiver window is 20 packets wide). Two sets of scenarios are simulated, with tcp packet sizes set to 250 bytes in one, and 2000 bytes in the other set. No MAC segmentation is used and capture effect is turned off. The number of active STAs is increased gradually from 2 to 20, simulating the most frequent numbers of STAs in actual ad-hoc networks. Simulations are repeated with different *k* and CW_{min} parameters. In addition to CPCF parameter *k*, other Physical Layer (PHY) and MAC parameters used are inherited from ns2 802_11Ext class. Table 1 shows fixed parameters used in all simulations.

PHY bandwidth of 6 Mbps is chosen to emphasize the influence of packet lengths used in simulations (250 and 2000 bytes). Low PHY bandwidth produces greater ratio between packet transmission time and overhead than high PHY transfer rates. This way, overall throughput is less affected by overhead, and more by transferring of large collided frames. Therefore, when packets are large (2000 bytes) and low PHY bandwidth is used, collision rate has more influence on throughput than would have for faster PHY. Short packets (250 bytes) represent the real-time traffic and have smaller aforementioned ratio, so keeping the contention resolution period short becomes more important, even with low PHY bandwidth.

In simulations, both DCF and CPCF use three different CW_{min} values (15, 31 and 63). For each CW_{min} value, four different freezing limits *k* are used (0, 1, 2 and 6).

Figures 1, 2 and 3 show the throughput results when both protocols use CW_{min}=15, CW_{min}=31 and CW_{min}=63, respectively.

TABLE I. FIXED CPCF PARAMETERS

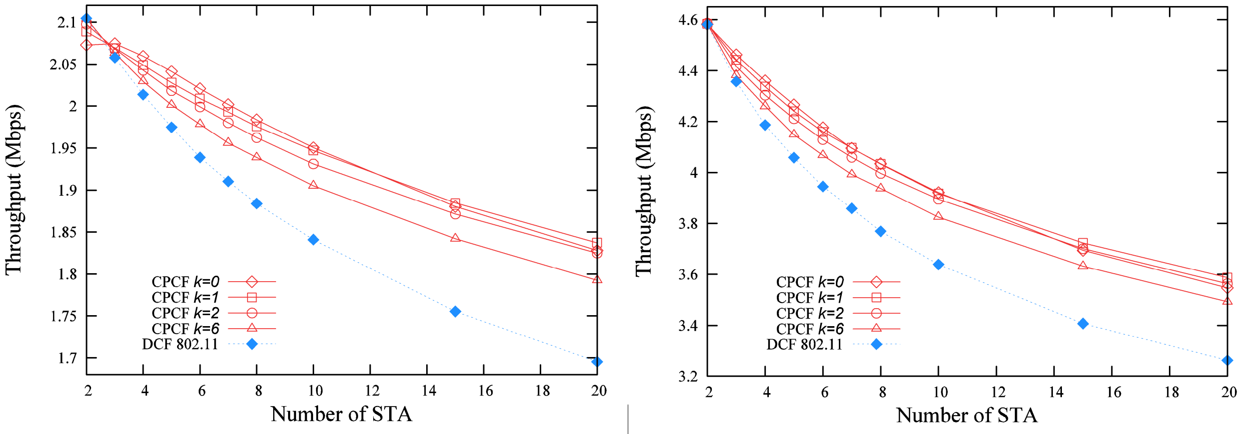
Parameter	Value
SIFS	16 µs
Slot Time	9 µs
ShortRetryLimit	7
RTSThreshold	3000 bytes
PHY bandwidth	6 Mbps

DCF countdown achieves good results when the number of STAs is low (up to 4) due to countdown effect. The benefit of DCF's countdown effect is especially visible in Figure 2a and 3a where CW_{min} is 31 and 63, respectively. When CW_{min} insures low collision rate, reducing the contention overhead becomes extremely important, especially when packets are short (collision loses are less expensive in terms of throughput), and thus DCF achieves better results than CPCF. When the number of STAs becomes large, DCF shows poor performance. This can be explained with collision memory, which increases overall collision rate when the number of STAs increases.

CPCF protocol shows good performance, in both low and high load conditions, depending on the freezing limit *k* used. For large *k* value (*k*=6), CPCF has the ability to countdown longer and can significantly reduce the contention overhead. This is very important when the number of collisions is low (STA count is low). However, long countdown can increase collisions due to collision memory and this is the reason why *k*=0, *k*=1 show better performance when the number of STAs gets large.

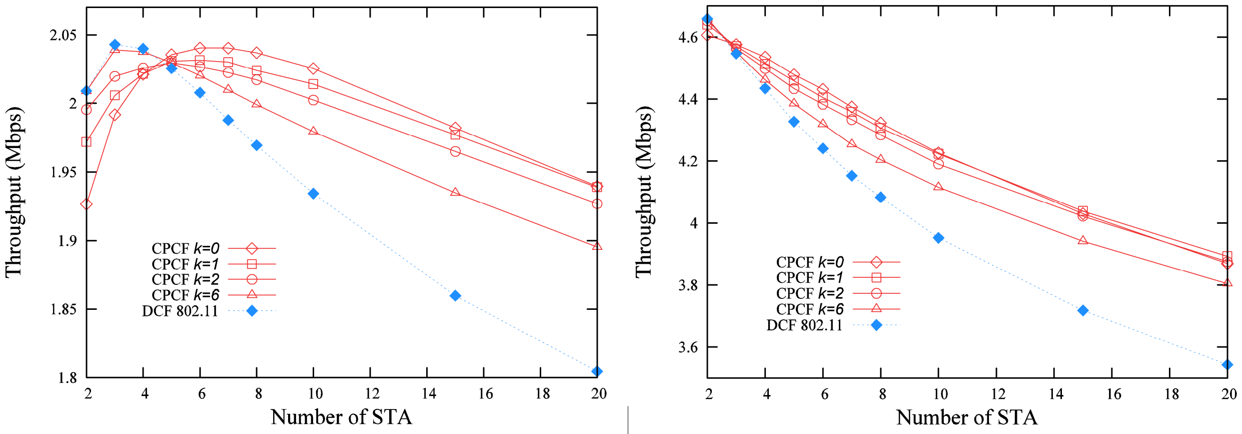
The really interesting effect occurs when the number of STAs becomes very large (above 15). Since the smallest parameter *k*=0 insures no collision memory effect, it was expected that it would produce the lowest collision rate. However, this was not true. When the number of STAs gets large, *k*=0 and *k*=1 graphs show small but definite difference in collision rate in favor of *k*=1 (Figures 4, 5 and 6). There is obviously another mechanism affecting collisions besides collision memory.

One possible explanation can be found in disturbed distribution of *PN* choices because of constrained countdown freezing. This introduces complex relationship between contention overhead and collision probability. This is visible in the Figure 3a and 3b when the number of STAs is 20. CPCF *k*=0 exhibits better throughput results for short packets than for long packets, when compared with *k*=1. Surprisingly, *k*=0 achieves good throughput results due to shorter contention overhead and not due to lower collision rate as expected. This is confirmed in Figure 6, where it is visible that the *k*=1 shows lower collision rate than *k*=0. Therefore, constrained countdown freezing mechanism should be further investigated in the future.



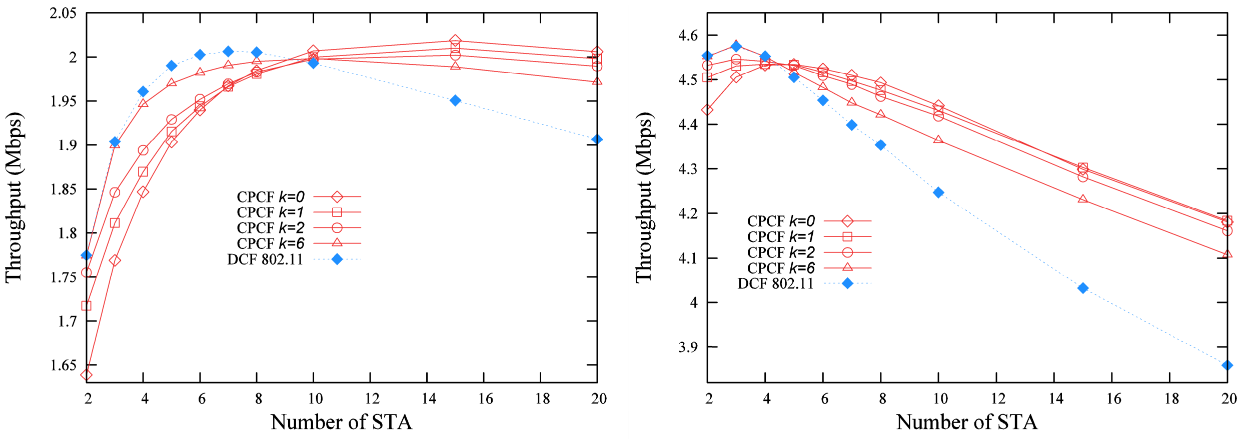
a) b)

Figure 1. Throughput CW_{min} =15: (a) tcp packet size 250 bytes (b) tcp packet size 2000 bytes



a) b)

Figure 2. Throughput CW_{min} =31: (a) tcp packet size 250 bytes (b) tcp packet size 2000 bytes



a) b)

Figure 3. Throughput CW_{min} =63: (a) tcp packet size 250 bytes (b) tcp packet size 2000 bytes

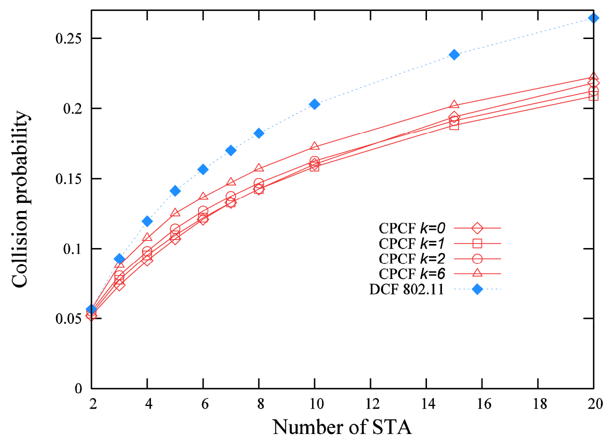


Figure 4. Collision probability CW_{min} =15

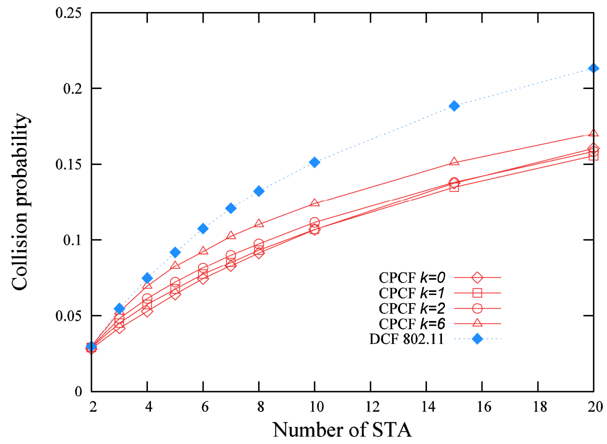


Figure 5. Collision probability CW_{min} =31

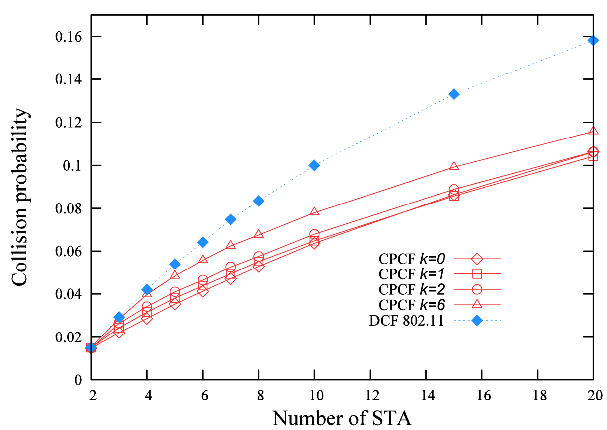


Figure 6. Collision probability CW_{min} =63

VI. CONCLUSION AND FUTURE WORK

In this paper, we have elaborated collision memory and have proposed a new protocol that can reduce collision memory effect. The backoff freezing mechanism of DCF protocol causes collision memory effect by preserving once developed priority collisions, which can increase the physical collision rate. In order to solve the problem, a new collision memory avoidance protocol called Constrained Priority Countdown Freezing (CPCF), is introduced. The CPCF constrains the priority freezing with freezing limit *k*. Freezing limit defines the maximum number of contention resolution periods in which a STA is allowed to decrement and freeze its priority. Simulations have shown that by constraining the countdown freezing mechanism, better throughput results are achieved compared to DCF protocol. However, the simulations have also shown that countdown effect and collision memory effect are not the only effects occurring in CPCF type of countdown. Surprisingly, the most constrained version of CPCF protocol (when *k*=0), which has no collision memory, can have higher collision rate and lower contention overhead compared to less constrained versions (*k*=1, *k*=2). An investigation of this effect should be subject of the future work.

REFERENCES

[1] IEEE Std. 802.11, "IEEE Std. 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 2007.

[2] S. Choi, K. Park, and C. Kim, "On the performance characteristics of WLANs: revisited," ACM SIGMETRICS Performance Evaluation Review, ACM, 2005, pp. 97-108.

[3] I. Kedzo, J. Ozegovic, and A. Kristic, "BPC – a binary priority countdown protocol," *Ad Hoc Networks*, submitted for publication, unpublished.

[4] IEEE 802.11 WG, "Wireless LAN Medium Access Control (MAC) and Physical-Layer (PHY) Specifications," *IEEE std*, 1999.

[5] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of ieee 802.11 modeling and simulation in ns-2," Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, ACM, 2007, pp. 159-168.

[6] M. A. Youssef, A. Vasan and R.E. Miller, Specification and Analysis of the DCF and PCF Protocols in the 802.11 Standard Using Systems of Communicating Machines, Proceedings of the 10 th IEEE International Conference on Network Protocols (ICNP'02) 1092-1648, 2002.

[7] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: enhanced service differentiation for IEEE 802.11 802.11 wireless ad-hoc networks," *Proc. IEEE WCNC'03*, pp. 1373-1378.

[8] N.S. Nandiraju, H. Gossain, D. Cavalcanti, K.R. Chowdhury, and D.P. Agrawal, "Achieving Fairness in Wireless LANs by Enhanced 802.11 DCF," the Proc. of IEEE Wireless and Mobile Computing, Networking and Communications (WiMob 2006), 2005.

 ISBN: 978-1-61208-203-5