International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 7, No.4 (July-2018)

http://dx.doi.org/10.12785/ijcds/070405

IEEE 802.11 DCF Improvement: Waiting DIFS while Waiting Back-off

Latifa Souad Mahi-Rekik and Malika Bourenane[1]

[1]*University of Oran1 Ahmed Ben Bella, Oran, Algeria*

Received 28 Jan.2018, Revised 10 Mar. 2018, Accepted 11 Jun. 2018, Published 1 July 2018

Abstract: With the emergence of time-sensitive applications such as games and telephony, the introduction of the Quality of Service (QoS) in general and the improvement of transmission delays in particular have become a must in wireless networks. The distributed coordination function (DCF) being the fundamental access method and the basis of the wireless LANs IEEE 802.11 MAC protocol, a great number of works have been done to improve it. In DCF, after sensing an idle channel, stations have to wait before the transmission of each frame a length of time called DIFS (DCF Inter Frame Space) followed by another called back-off. If the medium becomes busy during the back-off process, the back-off timer is paused and resumed when the medium is sensed free for a DIFS again. This time loss becomes considerable when the number of interruptions of back-off process grows. This paper proposes a contribution to improve DCF by combining IFS and back-off time. The simulation results show that the approach benefits are proportional to the contention level of the network and to the number of hops in multi-hop network topologies.

Keywords: IEEE 802.11, CSMA/CA, DCF, IFS, DIFS, Back-off algorithm, contention window

1. INTRODUCTION

The fundamental access method of the IEEE 802.11 MAC protocol is a distributed coordination function (DCF) also called CSMA/CA for Carrier Sense Multiple Access with Collision Avoidance. The distributed CSMA/CA algorithm requires a gap of a minimum specified duration (called Inter Frame Spacing-IFS) between the different frames transmitted in order to establish a priority system between frames. Control frames such as acknowledgment (ACK) or clear to send (CTS) for example are given priority higher than that of data frames by waiting smaller IFS. The sending station checks if the channel has remained free during this time before it can transmit its frame. If the channel is busy, the station must delay its transmission by choosing a random number called back-off in an interval called contention window; this will determine an additional waiting time in order to solve partially channel access conflicts.

The objective of this work is to minimize wait times which lead to under usage of the channel by combining IFS and back-off times. In our approach, the DIFS time is eliminated whenever the back-off time is greater than or equal to DIFS (DCF IFS for Data frames); thus, the station is not obliged to wait for the channel to be free for DIFS time since its back-off time already includes it. However, DIFS wait is kept in case of back-offs that are smaller than DIFS. The approach is simulated using NS2 simulator and tested on different contention level scenarios of ad hoc single-hop and multi-hop 802.11 networks. Compared to the standard DCF, better performance is noticed in terms of throughput and end-to-end delay.

The rest of this paper is organized as follows: section II gives a brief description of DCF mechanism focusing on its inter-frame spacing and back-off mechanisms. Section III presents some researches on DCF and back-off algorithms improvement. Details and explanations of our contribution are given in section IV. Section V is devoted to implementation on NS2 simulator. Section VI presents simulation results and their interpretation for different topologies and scenarios. Section VII concludes the paper.

2. IEEE 802.11 MAC SUBLAYER

MAC sublayer of stations operating in an IEEE 802.11 LAN proposes three coordination functions which control access to the wireless medium: (1) DCF, the standard basis, (2) Hybrid Coordination Function (HCF) present only in QoS stations and (3) Point Coordination Function (PCF) optional, used for contention-free services. The IEEE 802.11 MAC architecture is described in Fig.1 as providing the PCF and HCF through the services of DCF [1].

The DCF coordinates the access to a shared medium (more exactly channel) by multiple stations. DCF is a CSMA/CA access mechanism. Like Ethernet, the station first checks that the channel is clear before transmitting. DCF defines two access mechanisms for packet transmission: (1) the basic one called the two way handshaking technique where the sender transmits data and the receiver responds with an ACK; and (2) the optional one called RTS/CTS technique or the four handshaking technique where the sender first transmits a short Request to send frame (RTS) and waits for a Clear-

E-mail:mahi.latifa@univ-oran.dz, mb_regina@yahoo.fr

To-Send frame (CTS) from the destination before transmitting data and receiving ACK.

This last technique is used to prevent hidden node problems but causes additional transmission delays. The rest of the paper focuses only on the basic technique.

DCF is based on a two-type time delay principle : (1) the inter frame spacing (IFS) to establish a priority system between frames of different natures,and (2) the random back-off timing to establish a priority system between stations which want an access to the channel simultaneously.

A. Inter frame spacing

The time interval between frames called the IFS plays an important role in coordinating access to the transmission medium. DCF uses five different inter frame spaces (see tables I and II). Varying inter frame spaces creates different priority levels for different types of traffic. The logic behind this is simple: high-priority traffic doesn't have to wait that long once the medium becomes idle. Therefore, if there is any high-priority traffic waiting, it grabs the network before low-priority frames have a chance to try [1, 2].

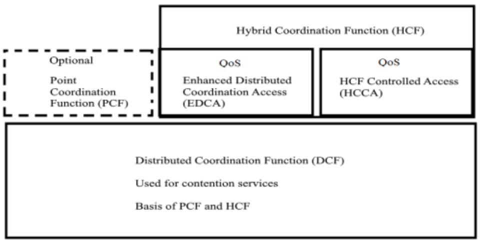


Figure 1 IEEE 802.11 MAC architecture

TABLE I DIFFERENT IFS AND THEIR USE

IFS	Name	Used by
SIFS	Short Inter frame Space	CTS, ACK, Fragments of a frame
PIFS	PCF Inter frame space	Point coordinated traffic
DIFS	DCF inter frame space	Data and management frames after a correctly received frame
AIFS	Arbitration inter frame space	QOS stations
EIFS	Extended Inter frame space	After an incorrectly received frame

TABLE II SOME IFS VALUES

		PHY Value Examples (in μs)		
Timings	**Value**	**FHSS**	**DSSS**	**OFDM**
Slot time	PHY dependent [1]	50	20	9
SIFS	PHY dependent [2]	28	10	16
PIFS	SIFS+ SlotTime	78	30	25
DIFS	SIFS+2*SlotTime	128	50	34

1. A Slot Time=Minimum time required for the PHY to determine the state of the channel +time to turnaround from receive to transmit mode +air propagation time +MAC processing delay.

2. SIFS=Time required to pass channel information between PHY sub-layers and between PLCP and MAC +time to turnaround from receive to transmit mode +MAC processing delay.

B. Backoff timing

The IEEE 802.11 Standard defines the Binary Exponential Back-off (BEB) algorithm to be performed in the following cases: (1) when the station listens to the medium before the first transmission of a packet and the medium is busy. (2) After each retransmission. (3) After a successful transmission. Whenever a data frame is to be sent, the station senses the medium; if it is free for at least a DCF inter-frame space (DIFS) period of time, the back-off mechanism is not used and the frame is directly transmitted. Otherwise, if the medium is busy, a back-off time B (measured in time slots which depend on the characteristic of physical layer) is chosen randomly in the interval [0, CW], where CW is called the contention window. After the medium has been detected idle for at least a DIFS, the back off timer is decremented by one for each time slot the medium remains idle [3].

If the medium becomes busy during the back off process, the back off timer is paused, and is resumed when the medium is sensed free for a DIFS again. When the back-off timer reaches zero, the frame is transmitted. Fig. 2 below illustrates the back-off process of two stations wanting to reach the channel simultaneously.

On the first transmission attempt, CW is set to a minimum value CWmin and at the next times (at the event of a collision), CW is doubled until it reaches a maximum value CWmax i.e. CW=min (2*CW, CWmax) . A new back-off time is then chosen and the back-off procedure starts over. After a successful transmission, the contention window is reset to CWmax.

We can design the Back-off algorithm as presented in Algorithm I below.

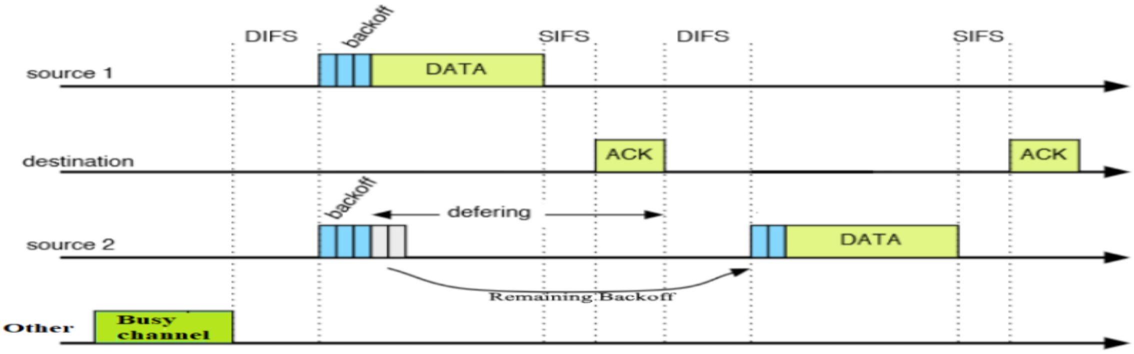


Figure 2 Back-off mechanism [4]

Algorithm I Back-off Algorithm in DCF transmission cycle

```
If (first transmission) and (channel free for DIFS) go to 4
     CW=CWmin
     For each packet transmission or retransmission
1.   Wait for channel to be free for a DIFS
     // DIFS wait before back-off start
2.   N← rand(0,CW)
     // Back-off time initialization (in time slots)
3.   Do
         o  Wait a slot_time
         o  If (channel has been free during the whole
            slot_time)
                ▪  N←N-1
         o  Else
            // Channel busy, Back-off interrupted (frozen)
                ▪  Wait for channel to be free for a
                   DIFS
                   // DIFS wait before back-off resume

     While (N>0)
4.   Transmit.
5.   If (successful transmission) CW←CWmin;
              else CW←min(2*CW, CWmax)
```

3. Related work

DCF being the fundamental access method and the basis of the IEEE 802.11 MAC protocol, many works have shown great interest to improve it. The reminder of this section classifies the contributions into two main categories:

A. Introduction of a certain quality of service :

DCF is unable to provide the required performance for voice and video applications, because it is mainly developed for best effort services. Basically, service differentiation at the MAC level is achieved by two main methods: priority and fair scheduling. The former binds channel access to different traffic classes by prioritized contention parameters; the latter partitions the channel bandwidth fairly by regulating wait times of traffic classes in proportion according to given weights. The tunable parameters for both approaches are CW size, back-off algorithm, and inter-frame space[5]. Some specific service differentiation mechanisms are:

1) *Enhanced DCF (EDCF):*

Part of IEEE 802.11e, EDCF prioritizes traffic by including Arbitrary IFS (AIFS) and minimum and maximum back-off window sizes; in EDCF, traffics keep using the same DCF access mechanism but have different probabilities of winning the channel. The IEEE802.11e amendment was approved in order to provide QoSsupport to WLANs. It defined the Hybrid Coordination Function (HCF) as an enhanced medium access mechanism which includes two access mechanisms that are: Enhanced Distributed Coordination Access (EDCA) and HCF Controlled Channel Access (HCCA). Although this amendment introduces the service differentiation scheme, it was not able to guarantee QoS for applications having strict QoS requirements [6].

2) *Distributed Fair Scheduling (DFS) and Optimal DCF (O-DCF):*

The main idea of DFS [7] is to differentiate the back-off interval (BI) based on the packet length and traffic class. In DFS, the station with smaller BI transmits first. O-DCF[8]controls link access aggressiveness by both CW size and transmission length based on the state of the queue. In O-DCF, links with a greater queue length are prioritized by decreasing their CW size and/or by increasing their transmission length.

B. DCF enhancement by changing backoff algorithm:

BEB is the key component of the DCF mechanism; however, it suffers from certain problems including significant delay degradation in case of saturated networks. Several proposals of back-off schemes have been made in order to solve the problem of exponential increase of the contention window after each failed transmission (generating useless access delays) and thus, providing better delay performance; we can cite for example:

1) *MILD (Multiplicative Increase Linear Decrease):*

MILD is a back-off algorithm where the multiplicative factor is 1.5 (instead of 2 in BEB); in MILD, the back-off upper bound (CW) is set as follows:

CW=min(1,5*CW,CWmax) after each failed transmission [9,10].

2) PBA (Padovan Backoff Algorithm):

In PBA which is based on the Padovan sequence, CW takes, after each failed transmission, the next Padovan term (CW=min(P(r),CWmax)), where r is the retry count and P(r) is the Padovan term [11].

In their study of back-off design for IEEE 802.11 DCF , Xinghua and Lin categorize back-off schemes into two groups (aggressive back-off with $\lim_{i\to\infty}(CW_{i+1}/CW_i)>1$and mild back-off with $\mathrm{Lim}_{i\to\infty}(CW_{i+1}/CW_i)=1$). They show that aggressive back-off schemes such as BEB suffer from delay degradation when the network size is large.

Our contribution attempts to minimize 802.11 DCF channel access delay by combining inter frame space time and back-off time. In what follows, our approach is named DIB_DCF for DIFS In Back-off DCF and the original IEEE 802.11 DCF ORG_DCF.

4. PROPOSED APPROACH

As explained in section 2.B (back-off timing), a station wanting to transmit must defer its transmission for an additional time equal at least to DIFS (wait until the channel remains free for at least DIFS) before beginning its back-off process.Once the back-off process begins, it is decremented by one slot as long as the channel remains free; if it becomes busy again, the back-off process is frozen and resumes after a period of idle channel equal at least to DIFS and the operation is repeated until the back-off reaches zero.

Suppose a back-off process which has been interrupted n times before it reaches zero (Fig. 4 a), theoretically, the minimum time waited by the station (wait while channel is free) in this case is:

minwait_ORG=DIFS+BO1+DIFS+BO2+DIFS+…+DIFS +BOn

$$\text{minwait\_ORG}= n*\text{DIFS} + \textstyle\sum_{i=1}^{n} BO_i \quad =>$$

$$\text{minwait\_ORG}=n*\text{DIFS}+\text{BO} \qquad (1)$$

Where BOi is the back-off portion elapsed before the interruption i.

And $\sum_{i=1}^{n} BO_i = BO$ (Back-off time).

Our approach proposes to:

- Eliminate the DIFS wait before the first packet transmission and before each packet retransmission (back-off start) whenever, the initial back-off time (BO) is greater than or equal to DIFS.
- Eliminate the DIFS wait before each back-off resume whenever the remaining back-off time(RB) is greater than or equal to DIFS.

The minimum wait becomes in this case:

If (RB>DIFS) $\text{min\_wait\_DIB}=\sum_{i=1}^{n} BO_i$

Else $\text{min\_wait\_DIB}=(\sum_{i=1}^{n} BO_i)+\text{DIFS}$

In other terms :

If (RB>DIFS)min_wait_DIB= BO time

else min_wait_DIB=BO time +DIFS (2)

From (2) and (1), we deduce that the gain is at least equal to (n-1)*DIFS.

Fig. 3 illustrates the original DCF and the new DCF mechanisms

Fig. 4 b illustrates the new DCF transmission cycle.

Algorithm 2 gives the new steps for DCF transmission cycle.

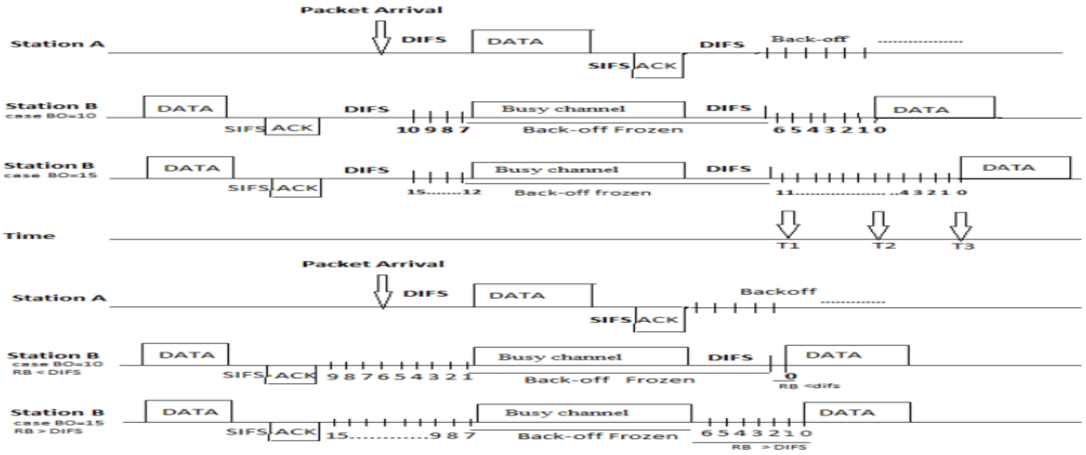


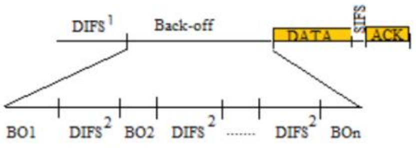
Figure 3 Illustration of ORG_DCF and DIB_DCF

The upper part of Fig. 3 shows the 802.11 DCF process in an example of two stations A and B transmitting in a wireless channel; the lower part shows our approach with the same example. Between both parts, we can see a time axis where the beginning times of DATA transmissions are reported.We can notice that, in both cases of station B (Remaining Back-off (RB)< DIFS and RB > DIFS), the station reaches the channel in a shorter time; it transmits data at time T1 in case of RB< DIFS and at T1+one slot time in case of RB > DIFS while with the original DCF, station B transmits respectively at T2 and T3.

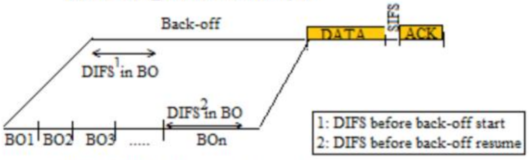
Algorithm II DIB-DCF cycle Algorithm

```
    For each packet transmission or retransmission
1.  N← rand(0,CW)
    // Back-off time initialization (in time slots)
2.  Rtime← N*slot_time
    //Back-off time in µs)
3.  If (Rtime<DIFS)
        o  Wait for channel to be free for a DIFS
           // DIFS wait before Back-off start only in case of
           BO<DIFS
4.  Do
        o  Wait a slot_time
        o  If (channel has been free during the whole
           slot_time)
               ▪  N←N-1
        o  Else
           // Channel busy, Back-off interrupted (frozen)
               ▪  If (Rtime<DIFS) Wait for channel to be
                  free for a DIFS
                  // DIFS wait before back-off resume
                  only if RB<DIFS

    While (N>0)
5.  Transmit
```



(a) The ORG_DCF transmission cycle



(b) The DIB_DCF transmission cycle (when BO >DIFS)

Figure 4 Transmission cycles in ORG and DIB DCF

5. SIMULATION OF THE PROPOSED APPROACH

In this section, NS2 simulator is described and the main changes made in its code to implement our proposal are presented. Simulation parameters are also given.

NS2 simulator is a popular discrete event network simulator developed under several previous research grants and activities; it remains in active use and will continue to be maintained [15].

A. Basic principles of NS2

NS2 is a C++ executable program which we call with a TCL configuration file as a parameter; Fig.5 describes the basic architecture of NS2. A TCL configuration file (called TCLsimulation script) is written in the interpreted Tool Command Language-TCL to describe the network to simulate (number of stations called nodes in NS2, type of links between nodes -duplex link, half duplex,…-, applications attached to nodes -constant bit rate, variable bit rate…- , type of physical layer, MAC layer, time at which transmissions begin, etc.). One simulation script describes one network scenario.

The C++ code (object oriented) contains the modelization of different components and protocols implied in the OSI layers of wired and wireless networks (propagation model, MAC layer 802.11, application layer, etc.) in addition to special components like timers or random number generators.

From the initial scenario, NS2, a discrete event simulator, creates a list of events with their execution times (the execution of an event consists in executing its associated actions). The simulation process then consists in executing the events in ascending order of execution times. In order to delay events (delaying events consists in scheduling them for a specified time), NS2 simulator uses the special components called timers. Class back-off timer is used to implement back-off timing and class defer timer to implement inter frame spacing delay. For a more complete and detailed presentation of NS2 see [13].

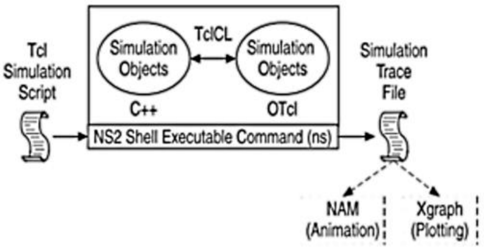


Figure 5 Basic architecture of NS2 Simulator [13].

B. *Changes in NS2 code*

The main changes were made in the following NS2 C++ functions (in file NS2.35/mac/mac-timer.cc):

1) *BackoffTimer::start (int cw,int Idle,double difs)*

```
{…………
Double slot = mac_phymib_.getSlotTime();
rtime=(Random::random()%cw)*slot;
difs_wait=difs;
If (rtime>= difs) difs_wait=0;          // Added
…………
s.schedule(this, &intr, rtime+difs_wait);
}
```

2) *BackoffTimer::resume (double difs)*

```
{………….
Difs_wait = difs;
If  (rtime>= difs) difs_wait=0;          // Added
          ………
s.schedule(this, &intr, rtime+difs_wait); }
```

C. *Simulation parameters*

Table III.gives the main simulation parameters which remain the same for all simulated scenarios.

CWmin has been adapted to the size of the network; both values 31 and 63 have been used.

TABLE III SIMULATION PARAMETERS

Parameter	Value
MAC protocol	802.11 b (DSSS PHY)
CWmin CWmax	31, 63 1023
Basic rate, DataRate, CBR Rate	1Mb/s, 2Mb/s, 1600Kb/s
Packet Size	512 Byte
Traffic (Flows)	UDP/CBR
Simulation Time, recording period	80s, 2s
Mobility	none
RTSThreshold	3000 bytes (RTS/CTS mechanism disabled)
CSThreshold, RXThreshold (carrier sense and communication ranges),	550m, 250m

6. SIMULATION RESULTS AND EVALUATION

In order to highlight the benefits of our approach, we have simulated it on different scenarios and network topologies.

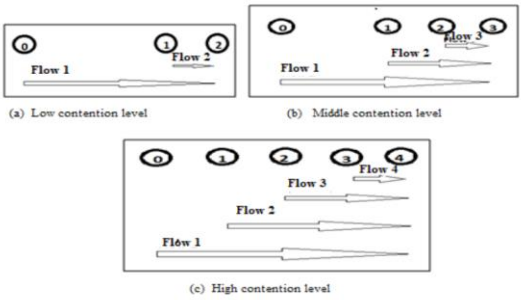
A. *Single-hop topologies*

1) *Scenarios*

Figure 6 presents three single-hop scenarios. The first one (Fig. 5a) contains 3 nodes and 2 flows (a flow from node 0 to node 2 and another one from node 1 to node 2). In order to increase the contention level of the network, new nodes and new flows are added to obtain scenario 2 and 3 (Fig. 5b, Fig.5c). All nodes are within the communication range of each other; they use the same channel and start at the same time.

We consider flow 1 as the main flow and the others as secondary flows. The latter play the role of disruptive flows increasing the interruption probability of the back-offs of flow1 frames.

With these scenarios, we aim at showing that the gain in delay is closely related to the number of back-off interruptions.



(a) Low contention level (b) Middle contention level

(c) High contention level

Figure 6 Network scenarios

2) *Results, interpretation and evaluation*

As expected by the theoretical calculations, the nodes that encounter the most interruptions in their back-offs find their delays the most diminished (more DIFS removed). The nodes that encounter the most interrupts are those that are the furthest away from the receiver (Flow1 for example). The nodes close to the receiver have a higher probability of transmission since their packets arrive in a shorter time at destination. Thus, the delay of flow 1scenario 3 is decreased by 7,24%. The results also show that the greater the number of nodes between the transmitter and the receiver increases, the more the gain of flow1 is important. Finally, the mean end-to-end delay of all scenarios has improved and the average throughput is the same in all cases; however, ORG_DCF remains more efficient when there is no or few contention (flow 3 scenario 2, flows 3 and 4 scenario 3). Table 4 and fig.7 show the simulation results for the single-hop topologies.

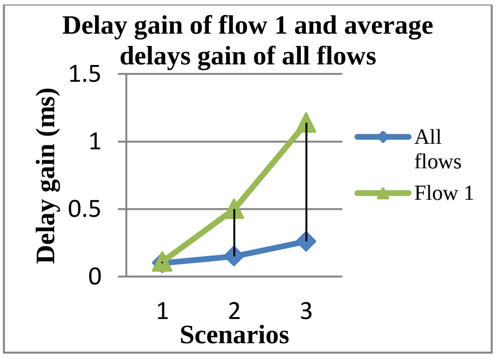


Figure 7 Delay gain flow1 and all flows

TABLE IV SIMULATION RESULTS FOR SINGLE HOP TOPOLOGIES[1]

	Scenario 1				**Scenario 2**				**Scenario 3**			
	Delay		***Throughput***		***Delay***		***Throughput***		***Delay***		***Throughput***	
	ORG	DIB	ORG	DIB	ORG	DIB	ORG	DIB	ORG	DIB	ORG	DIB
Flow 1	6,96	6,85	0,64	0,65	10,97	10,47	0,41	0,42	15,73	14,59	0,29	0,30
Flow2	5,83	5,76	0,76	0,76	9,28	9,19	0,48	0,48	15,08	14,82	0,30	0,31
Flow 3					8,80	8,96	0,50	0,50	12,45	12,58	0,36	0,35
Flow 4									10,58	10,80	0,42	0,41
mean	6,40	6,30	0,70	0,70	9,69	9,54	0,46	0,46	13,46	13,20	0,34	0,34
Gain flow1	-0,11		0,01		-0,5		0,01		-1,14		0,01	
Gain rate flow1	-1,58%		1,56%		-4,55%		2,43%		-7,24%		3,44%	

1. Delay (ms), throughput (Mb/s)

B. Multi-hop topologies

1) Scenarios

Simple two, three and four hops network scenarios have been simulated and compared to a simple (one flow) single hop scenario.

In a multi-hop topology, the destination is out of the sender communication range; the transmitted frame is thus forwarded from node to node till it reaches the destination.

The goal behind these scenarios is to demonstrate that our approach benefits may increase with the number of hops since a same frame will initiate and perform the DCF procedure several times (for each hop).

The four hop network scenario presented in Fig. 8 is based on a string topology with no hidden node problem (all nodes can sense each other). This topology has been inspired from [14].

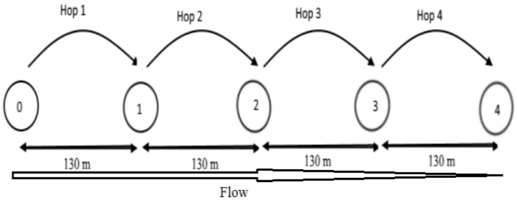


Figure 8 Multi-hop topology

2) Results, interpretation and evaluation

Figure 9 shows the results obtained for the four scenarios (one, two, three and four hops). We notice that the delay gain is closely related, this time, to the number of hops.

A very interesting observation is that the difference between the gains relative to two successive scenarios revolves around a DIFS.

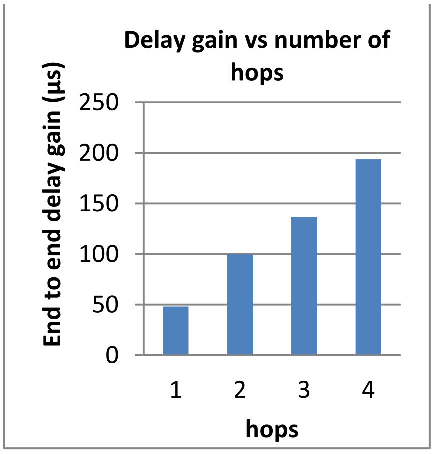


Figure 9 Delay gain versus number of hops

More accurately, it should revolve around p*DIFS; p being the probability for a station to select a back-off number greater than 2 (back-off time greater than DIFS) in the contention window [0,31].

P=29/32 p*DIFS=29/32*50=45,31µs.

This corresponds to the only DIFS removed for each packet transmission (DIFS before back-off start) when back-off time is greater than DIFS.

When a packet is forwarded n times in a multi-hop topology, on average, n*(p*DIFS)are removed from its end to end delay.

There are no DIFS removed before back-off resumes since in these simple scenarios, disruptive flows are absent.

C. Multi-hop scenarios with disruptive flows

Single-hop scenarios with different contention levels have allowed us to see the effects of our approach on flows whose backoffs are likely to encounter repeated interruptions (DIFS before backoffs resumes removed when remaining backoff>DIFS).

Simple multi-hop scenarios allowed us to confirm that the delay gain is also proportional to the number of hops. For each hop, the DIFS before backoff start is removed (in case of backoff > DIFS).

Multi-hop scenarios with disruptive flows should combine the benefits of the last two ones.

1) *Scenarios*

Two hops with three disruptive flows (scenario 1), two with six disruptive flows (scenario 2) and four with three disruptive flows (scenario 3) have been simulated.

Other scenarios have been simulated (four hops with twelve disruptive flows and twenty one nodes for example). However, in these scenarios, flows had very significant access delays (with ORG_DCF and DIB_DCF). It is the well-known IEEE 802.11 ad-hoc unfairness issue where some flows seize completely the channel when others are starved. We thought that the results with such scenarios would be unreliable.

2) *Results, interpretation and evaluation*

Table V presents the simulation results for the three scenarios mentioned above.

The results are very encouraging, up to 38,38% reduction of delay and 81,08% throughput increase in scenario 2.

TABLE V SIMULATION RESULTS FOR MULTI-HOP TOPOLOGIES[1]

	Scenario 1		**Scenario 2**		**Scenario 3**	
	Delay	***Throughput***	***Delay***	***Throughput***	***Delay***	***Throughput***
ORG_DCF	14,81	0,31	115,22	0,037	33,4	0,19
DIB_DCF	13,84	0,32	70,99	0,067	20,98	0,2
Gain	-0,97	0,01	-44,23	0,03	-12,42	0,01
Gain rate	-6,54%	3,22%	-38,38%	81,08%	-37,18%	5,26%

1. Delay (ms), throughput (Mb/s)

7. CONCLUSION AND FUTURE WORK

In this paper, a new approach is proposed to improve 802.11 DCF mechanism; DCF is based on delay principles: Back-off and IFS delays. By noticing that DIFS is, in some cases superfluous, authors have removed it, thus reducing the access time. To study the effects of this new approach, authors tested it on multiple scenarios with increasing contention levels using NS2.35 simulator.The results confirmed theoretical calculations which predicted a diminution of access time equal at least to (n-1) * DIFS (n being the number of interruptions in the back-off process).

By increasing the number of nodes between a transmitter and its destination, we increase the probability that the back-off process of the flow is interrupted by the neighboring flows (flows in the middle); simulation results show that the gain in delay obtained for a specified flow increases proportionally to the number of nodes placed inside the flow.

Thanks to this work, we have identified one of the in depth causes of the well-known IEEE 802.11 ad-hoc unfairness issue. The stations having a large bakoff number are doubly penalized. The waiting time of the backoff is great and these stations are more likely to be interrupted in their backing-off by the stations having smaller backoffs. The more interruptions there are, the more DIFS the stations have to wait.

In a multi-hop topology, for a same frame, DCF procedure is performed several times (each time the frame is relayed by the intermediate nodes). The delay gain is then proportional to the number of hops.

Results obtained with multi hop topologies are very promising. Despite these results, the approach needs to be explored further in order to better understand the new behaviour of the system as a whole and not only for specific flows.

Our contribution focus on IFS and back-off; these two basic mechanisms remain present in many of the amendments that came after IEEE 802.11 b. It would be interesting to simulate it on these ones.

REFERENCES

[1] IEEE Std 802.11, “Part 11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Computer Society, 2007.

[2] M. Gast, “802.11 wireless networks: The definitive guide,” O'Reilly, 2002.

[3] M. Sapna, “Modeling and Analysis of IEEE 802.11 DCF MAC,” Procedia Computer Science, vol. 57, pp. 473-482, 2015.

[4] D. Dhoutaut, [Free translation]“Study of the IEEE 802.11 standard in ad hoc networks, from simulation to experimentation” “Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc, de la simulation à l’expérimentation”PhD Thesis, INSA Lyon, pp. 31, 2003.

[5] Z. Hua, L. Ming, C. Imrich, P. Balakrishnan, “A survey of quality of service in IEEE 802.11 networks,”, IEEE Wireless Communications, vol. 11, n° 4, pp. 6-14, 2004.

[6] E. Charfi, L. Chaari, L. Kamoun,“PHY/MAC enhancements and QoS mechanisms for very high throughput WLANs: A survey,”EEE Communications Surveys & Tutorials, vol. 15, n° 4, pp. 1714-1735, 2013.

[7] V. Nitin, D. Anurag, G. Seema, B. Paramvir,“Distributed fair scheduling in a wireless LAN,”IEEE Transactions on Mobile Computing, vol. 4, n° 6, pp. 616-629, 2005.

[8] L. Jinsung, L. Hojin, Y. Yung, C. Song, K. Edward, C. Mung, “Making 802.11 DCF near-optimal: Design, implementation, and evaluation,”IEEE/ACM Transactions on Networking, vol.24, 2016.

[9] B. Vaduvur, D. Alan, S. Scott, Z. Lixia, “MACAW: a media access protocol for wireless LAN's,”ACM SIGCOMM Computer Communication Review, vol. 24, n° 4, pp 212-225, 1994.

[10] B. Swati, K. WU, "An Approach for Improving Performance of Back off Algorithm,"International J. Computer Applications, vol. 46, n° 5, pp. 45-49, 2012.

[11] D. Moad, S. Djahel, F. Nait-Abdesselam, "Padovan sequence based Backoff Algorithm for improved wireless medium access in MANETs,"Global Information Infrastructure and Networking Symposium (GIIS), 2014.

[12] S. Xinghua, D. Lin, "Backoff design for IEEE 802.11 DCF networks: Fundamental tradeoff and design criterion," IEEE/ACM Transactions on Networking (TON), vol. 23, n° 1, pp 300-316, 2015.

[13] T. Issariyakul, E. Hossain, "Introduction to Network Simulator NS2,"Springer Science+Business Media, LLC 2012.

[14] N. Ping Chung, L. Soung Chang, "Throughput analysis of IEEE 802.11 multi-hop adhoc network," IEEE/ACM Transactions on Networking, vol. 15, pp 309-322, 2007.

[15] "The difference between NS2 and NS3," http://www.nsnam.org/support/faq/ns2-ns3, visited 23/02/2018.

Latifa S. Rekik-Mahi obtained her engineering and magister degrees at the University of Oran, Algeria, and She has taught at the computer science department of this same university since 1998. Her first research interests were devoted to the interactive environments of human learning. Currently, she is preparing her "Doctorat" in the area of wireless networks. Her research interests involve scheduling algorithms, routing algorithms and MAC protocols.

Malika Bourenane obtained her "doctorat", magister and engineering degrees at the University of Oran, Algeria, and She has taught at the computer science department of this same university. Her research interests are linked to computer networking, routing, wireless computing, QoS, TCP. She supervises a research team; she contributed through her publications in many scientific international magazines and conference.