Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2022, vol. 27, no. 2, pp. 190–197
https://doi.org/10.2478/acss-2022-0020
https://content.sciendo.com

Network Condition-Aware Enhanced Distributed Channel Access for IEEE 802.11e Wireless Ad-Hoc Networks

Soulimane Mammar[1*], Hafid Haffaf[2], Kamel Rahmouni[3]
[1–3]*Research Laboratory in Industrial Computing and Networks (RiiR), University of Oran 1, Ahmed Ben Bella, Oran, Algeria*

***Abstract* – The increasing use of multimedia applications in wireless ad-hoc networks makes the support of quality of service (QoS) an overriding necessity. In this article, we present a new extension of the IEEE 802.11e EDCA scheme called NCA-EDCA, which uses the lifetime and the number of retransmissions attempts of a packet to assess the aggressiveness of the environment in order to adjust the channel access parameters to work in the best possible way and improve service quality accordingly. This new extension aims to (1) improve the performance of real-time applications; (2) increase the overall throughput by reducing the collision rate and (3) achieve an acceptable level of fairness. The simulation results show that our extension significantly improves EDCA for better QoS support of multimedia applications. More specifically, it increases throughput of the different flows by no negligible factors and significantly reduces the collision rate while maintaining a high degree of fairness between flows of equal priority.**

***Keywords* – 802.11 MAC, ad-hoc wireless networks, EDCA, IEEE 802.11e, QoS, multimedia traffic.**

I. Introduction

Interest in mobile ad-hoc network (MANET) is constantly growing due to its diverse applications. Examples range from standard Internet services such as web access to real-time services with stringent latency/throughput requirements such as video and Voice over IP (VoIP). The advent of the Internet of Things (IoT) as well as its reliance on ad-hoc communications in most cases has only increased this interest as this technology is expected to transform many industries with diverse applications, such as smart grids, telemedicine, vehicle telematics, surveillance, and manufacturing systems [1], [2]. Given the high demands and diverse requirements of these applications, there is a need to support Quality of Service (QoS). The 802.11 working group developed the 802.11e [3] protocol to provide such QoS support at the MAC (Medium Access Control) layer. Numerous studies have evaluated the current standard [4]–[5] and highlighted the importance of the improvements that 802.11e offers.

The standard defines two different access mechanisms: Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA). We focus on the former here because it is the only mechanism available for ad-hoc communication.

Although 802.11e improves current access mechanisms, there are still several issues that need to be addressed to enable better QoS support. In this study, we focus on three points: *network condition/load, packet-level service differentiation,* and *fairness*. Our main goal is to design a scheme that achieves the best performance for multimedia applications under high load conditions without neglecting best-effort/background traffic. Low-priority traffic cannot always be sacrificed and in some cases (SIP, H323, MGCP protocols) is essential for the operation of multimedia.

The rest of this paper is organised as follows. Section II gives a quick overview of the legacy 802.11 standard and the 802.11e amendment. Section III provides an overview of the existing work and discusses the main features of our scheme. Section IV introduces the new Network Condition-Aware access mechanism called NCA-EDCA. In Section V, we present simulation results, including throughput and some other metrics, and compare them to those of the IEEE 802.11e EDCA protocol. Section VI concludes this study.

II. The 802.11 Standard

A. Legacy 802.11

The 802.11 basic protocol defined in [3] is called Distributed Coordination Function (DCF). In DCF, a node starts its transmission when the medium is idle longer than the Distributed Interframe Space (DIFS). If the medium is busy, the node defers its transmission until a DIFS is detected, and then generates a random backoff period before transmission. The backoff timer decreases when the channel is idle, freezes when it is busy and continues if the channel is idle again for more than one DIFS. A node can initiate a transmission when its backoff timer reaches zero. This random time is determined by each station as a multiple of a slot time (determined by the PHY) and selected uniformly from the range [0,*CW*]. CW is the Contention Window and is initially set to $CW_{\min}$.

* Corresponding author's e-mail: soulimane.mammar@enp-oran.dz

©2022 Soulimane Mammar, Hafid Haffaf, Kamel Rahmouni.
This is an open access article licensed under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0).

A station is informed of the success of its transmission by an acknowledgment (ACK) sent by the receiving station after a Short Interframe Space (SIFS). After each transmission error, retransmission is scheduled with a double-sized *CW* (without exceeding CW_{max}). After each successful transmission, the sending station resets *CW* to its minimum value CW_{min} and initiates another random backoff.

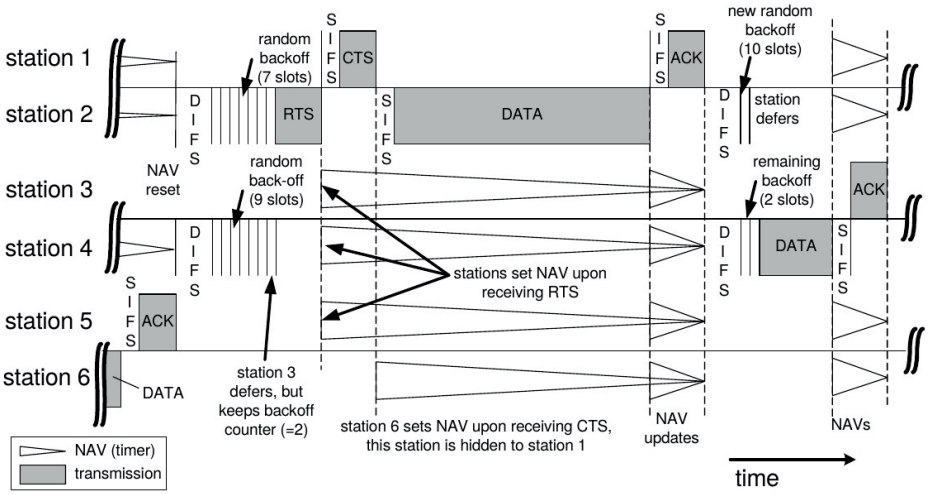


Fig. 1. 802.11 DCF frame exchange timing and NAV settings. Station 6 cannot see the RTS frame of the transmitting station 2, but receives the CTS frame from station 1. Although station 6 is hidden from station 2, it refrains from accessing the channel.

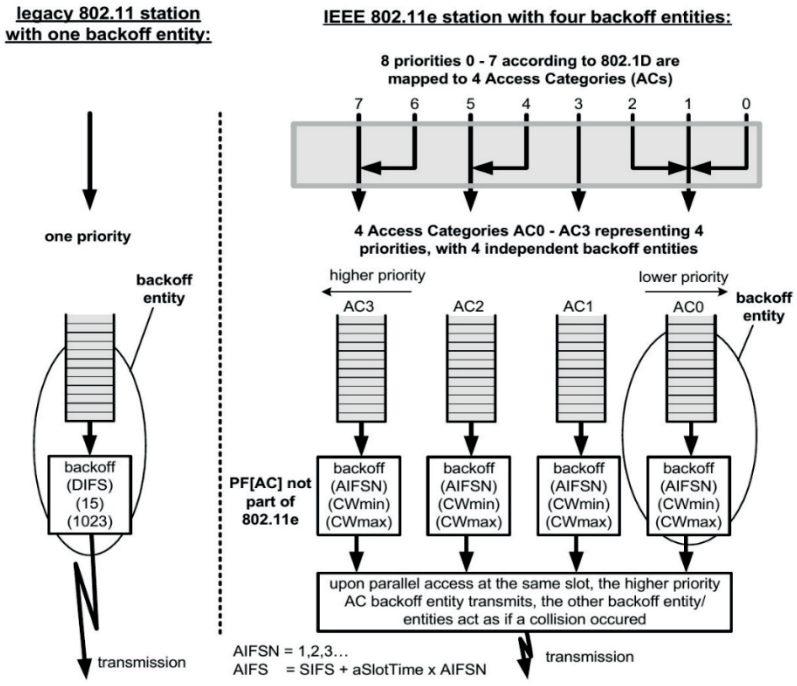


Fig. 2. Legacy 802.11 station and 802.11e station with four ACs within one station.

A station can employ RTS/CTS mode to minimize collision time and reduce the impact of hidden stations [3]. Before transmitting an MPDU, a station transmits an RTS, which is followed by the receiving station CTS. The stations within the range of the sender and the masked stations within the range of the receiver adjust their Network Allocation Vector (NAV) with the duration announced in the RTS/CTS frames. Fig. 1 shows a typical frame exchange in RTS/CTS access mode.

The only way to provide QoS for delay-sensitive applications in the old 802.11 is through a station called Point Coordinator (PC). However, in MANET environments such a central station does not exist.

B. QoS Mechanisms of the 802.11e Amendment

The 802.11e MAC is an extension of the existing 802.11 MAC for better QoS support. It defines a new coordination function called Hybrid Coordination Function (HCF) that combines two modes of media access: a distributed scheme, which is an improved version of the old DCF, called Enhanced Distributed Channel Access (EDCA), and a centralized scheme called HCF Controlled Channel Access (HCCA). In the following, we only focus on EDCA, as this is the only QoS mechanism compatible with ad-hoc communications.

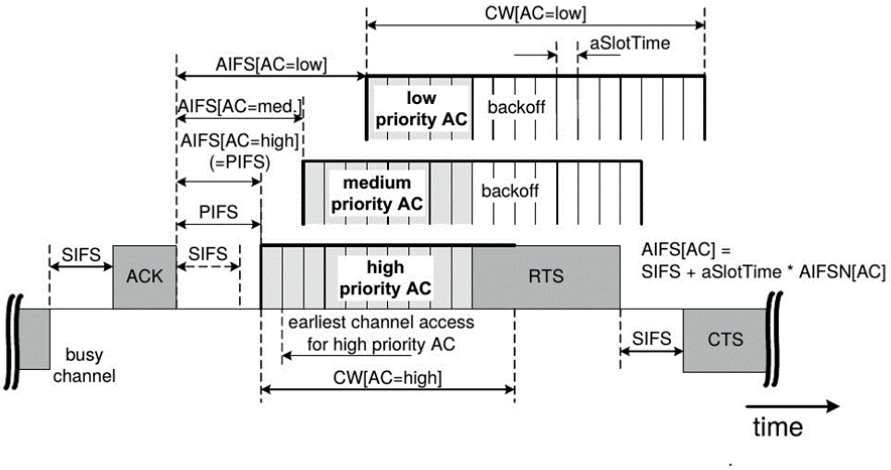


Fig. 3. Multiple backoff entities compete in parallel for channel access with different priorities. The earliest possible channel access time after a busy channel is DIFS.

QoS support in EDCA is achieved via Access Categories (AC). Each AC has its transmission queue and a set of channel access parameters. MSDUs are provided by parallel backoff entities within a single 802.11e station, prioritized by AC-specific contention settings.

An 802.11e station must implement four different backoff entities with four AC priorities 0…3. See Fig. 2 for an illustration of backoff entities. Each frame arriving at the MAC layer with a user priority is assigned an access category (AC).

Each backoff entity within a station independently competes for medium access, using different parameter values. It starts counting down the backoff counter after determining that the media has been idle for a duration defined by the Arbitration Interframe Space (AIFS[AC]) instead of DIFS. The AIFS[AC] is determined by:

$$\mathrm{AIFS[AC]} = \mathrm{SIFS} + \mathrm{AIFSN[AC]} \cdot \mathrm{SlotTime}, \quad \mathrm{AIFSN[AC]} \geq 2.$$

AIFSN[AC] is the number of additional idle slots to wait before a backoff entity starts its contention process, taking integer values in the interval [2,7]. The smaller AIFSN[AC] is, the higher the medium access priority.

The value of the backoff counter is randomly drawn from the range [1,CW+1], where CW is the Contention Window. The contention window initially assumes the minimum value $CW_{\min}$[AC] and can enlarge in the event of transmission errors without exceeding the maximum value $CW_{\max}$[AC]. See Fig. 3 for an illustration of AIFS[AC] and $CW_{\min}$[AC].

A virtual collision can occur when the counters of two or more backoff entities in the same station reach zero at the same time. In this case, the backoff entity with the highest priority gets access to the media, while the others act as if a real collision occurred.

The TXOPLimit[AC] is another parameter that is set for each AC. This is an interval of time during which a station is allowed to transmit data frames separated by SIFS. The larger TXOPLimit[AC], the larger the capacity share for this AC.

III. Related Work

In this section, we present some of the limitations of supporting QoS in 802.11e networks and related work to overcome these limitations.

One of the limitations of the 802.11e standard is that it does not take into account the current state of the network. Interference, propagation loss, and multipath propagation directly affect packet loss, causing retransmissions and consequently increasing latency and degrading throughput. In a worst-case scenario, the authors in [6] report a priority reversal, where higher-priority nodes perform less well than lower-priority nodes. The authors in [6] propose a mechanism, ChaPLeT, to maintain service differentiation in the presence of channel variations. Three parameters, namely *fragmentation threshold*, *persistence factor,* and *defer-countdown* are used to enable packet-level prioritization. Using a runtime adjustment policy, ChaPLeT dynamically configures the above parameters based on the traffic class as well as the current channel conditions. ChaPLeT shows up to 2× improvement in average throughput in different network scenarios and avoids priority inversion situations.

Another mechanism to cope with time-varying network conditions is adaptive EDCF (AEDCF) [7]. This scheme dynamically modifies the contention window value after each

successful transmission or collision to improve the performance of the EDCA. Adaptive slow *CW* reduction has been proposed to avoid collisions. This can be seen as a trade-off between wasting backoff time and risking a collision followed by the transmission of the entire packet. The simulation results show a 25 % improvement in overall goodput compared to EDCA versus a performance loss of low-priority background flows at high load.

In [8], the authors propose a DACW algorithm (Dynamic Adaptation of the Contention Window), which is an improvement on AEDCA and includes an adaptation of $CW_{\min}[i]$. This improvement reduces $CW[i]$ of high priority classes when competition is high and increases $CW[i]$ of low priority classes and vice versa. This tuning takes into account the network average collision rate.

The quality of service can also be negatively influenced by the number of competing nodes so that the required minimum performance can no longer be achieved despite prioritization. The simulations presented in [9] show that video streams suffer from a 33 % reduction in goodput when the number of nodes is increased from 4 to 16, despite EDCA prioritization.

The authors of [9] propose the Adaptive Fair EDCF scheme based on two points: (1) a fast backoff mechanism that accelerates from an adaptive threshold calculated for each AC according to the network state and (2) a uniform increase of the contention window in deferring periods. This scheme increases the total throughput by up to 34 % under high load conditions and achieves a high level of fairness. Simulation results demonstrate the superiority of Adaptive Fair EDCF over EDCF and the other improved schemes.

Although the differencing mechanisms used by EDCA provide better QoS for real-time traffic (and minimal service for low-priority traffic), overall performance degrades rapidly when the network reaches medium sizes [10]. Additionally, this mechanism could result in starving low-priority traffic under heavily loaded conditions, which is not always acceptable, especially for protocols like SIP, H323, MGCP..., etc. Since each TC is implemented as an independent backoff entity, the collision rate increases dramatically when the network load is high, impacting throughput and delay and, hence, time-bound traffic. This leads us to develop a new scheme called Network Condition-Aware EDCA (NCA-EDCA) that uses the packet lifetime and the number of retransmissions attempts to measure the aggressiveness of the environment and uses this measure to optimally adjust the channel access parameters. This new scheme aims to (1) improve the performance of real-time applications; (2) reduce the collision rate and thus increase the overall throughput and (3) be fair towards best-effort/background traffic.

IV. NCA-EDCA Scheme

While EDCA provides service differentiation, its performance under moderate and heavy loads is quite unsatisfactory. In fact, at these load levels, voice and video packets compete so fiercely for channel access that they more likely to collide since they behave similarly due to virtually similar channel access parameters. This increases the number of wasted idle slots due to backoffs and therefore leads to poor network packet delivery ratio (PDR) and wasted bandwidth as these delayed packets are no longer useful.

For a heavily loaded channel, the throughput of an access class *i* is:

$$Thr[i] = \frac{E(P_{\text{size}}[i])}{E(T_x[i])},$$

where $E(P_{\text{size}}[i])$ and $E(T_x[i])$ are the expected (average) packet length and transmission time, respectively. $E(T_x[i])$ can be expressed as follows:

$$E(T_x[i]) = E(T_{Fx}) + E(T_{Sx}),$$

where $E(T_{Fx})$ denotes the average time lost due to unsuccessful transmissions and $E(T_{Sx})$ is the expected time of successful transmission. $E(T_{Fx})$ and $E(T_{Sx})$ may be written as:

$$\begin{aligned} E(T_{Fx}) = \; & E(collisons) \times E(IdleSlots[i]) \times SlotTime \\ & + AIFS[i] + SlotTime \\ & + E(PacketTxTime[i]) \end{aligned}$$

$$\begin{aligned} E(T_{Sx}) = \; & E(IdleSlots[i]) \times SlotTime \\ & + AIFS[i] + SlotTime \\ & + E(PacketTxTime[i]) \\ & + SIFS + ACKTxTime \end{aligned}$$

The above equations show that maximum throughput is achieved when there are no collisions and no idle slots, i.e., $E(collisions) = 0$ and $E(IdleSlots[i]) = 0$. The maximum throughput of queue *i* is then:

$$Thr[i] = \frac{E(P_{\text{size}}[i])}{E(\text{Min}T_{Sx})},$$

where $E(\text{Min}T_{Sx})$ is the expected minimum time spent in a successful transmission. $E(\text{Min}T_{Sx})$ is defined as follows:

$$\begin{aligned} E(\text{Min}T_{Sx}) = \; & AIFS[i] + SlotTime \\ & + E(PacketTxTime[i]) \\ & + SIFS + ACKTxTime \end{aligned}$$

This ideal situation can only be achieved if the MAC function supports a perfect scheduling algorithm that selects only one packet for transmission across all queues and all nodes. It is impossible to implement such a function in a distributed manner (without going through a coordination point). To get close to this perfect behaviour, we propose a scheme that uses the time spent by a packet in the queue and the number of retransmissions attempts as key channel access parameters. It seems reasonable to assume that the time a packet spends in the queue, as well as the number of retransmission attempts, are good indicators of the aggressiveness of the environment and can therefore be used to parameterize the access function.

Our scheme called Network Condition-Aware EDCA (NCA-EDCA) uses a backoff timer that does not decrease linearly like in the standard, but with a speed that depends on the aggressiveness of the environment. This not only allows us to distinguish between packets from different ACs but also

between packets within the same AC. This new backoff mechanism significantly reduces the collision rate and consequently increases the overall media throughput.

In addition to the backoff mechanism, our scheme also sets the size of the contention window CW[AC] according to the aggressiveness factor AF in a way that reduces the probability of a collision.

TABLE I. EDCA PARAMETERS

AC	AIFSN	$CW_{\min}$	$CW_{\max}$	TXOP limit	Packet size	Packet interval	Flow rate
AC-BK	7	15	1023	0 ms	1500 bytes	12.5 ms	120 Kbytes/s
AC-VI	2	7	15	3.008 ms	1280 bytes	10 ms	128 Kbytes/s
AC-VO	2	3	7	1.504 ms	160 bytes	20 ms	8 Kbytes/s

C. Scheme Algorithm

The following is a description of our NCA-EDCA algorithm.

1. *Initialization*: Each time a new backoff is generated

$$R \leftarrow 0.$$

2. *Backoff timer decrease*: When a queue i detects that the medium is idle for $AIFS[i]$, it starts decrementing its backoff timer for each timeslot like this:

   a. Calculate $AF[i]$

   b. $R \leftarrow R + (1 - AF[i])$

   c. if $R \geq 1$

$$BT_{\text{new}}[i] = \begin{cases} BT_{\text{old}}[i] - 1 & \text{if } BT_{\text{old}}[i] \geq 1 \\ 0 & \text{otherwise} \end{cases}$$
$$R \leftarrow R - 1$$

   where $AF[i]$ is a factor that measures the backoff entity estimate of the *aggressiveness* of the environment. This factor depends on the *priority* of the queue, as well as the *age* of the packet to be transmitted and the number of retransmissions that have occurred so far. This factor takes values from 0 to 1, where 1 denotes a very harsh environment where it is practically impossible to send packets. This factor is the key element used by the various backoff entities to mitigate the aggressiveness of the environment by adopting a slow backoff timer reduction strategy.
   The aggressiveness factor for queue i is calculated as follows:

$$AF[i] = \min(\max(AMD[i], RMR[i]), 1)$$

   $AMD[i]$ and $RMR[i]$ are the ratios:

$$AMD[i] = \left\lfloor \frac{\text{Age}}{\text{MaxDelay}[i]} \right\rfloor$$

$$RMR[i] = \left\lfloor \frac{\#\text{Retransmissions}}{\text{MaxRetransmissions}[i]} \right\rfloor,$$

   where $\text{MaxDelay}[i]$ is the maximum acceptable delay for queue i, $\#\text{Retransmissions}$ – the current number of retransmissions and $\text{MaxRetransmissions}[i]$ – the maximum number of retransmissions allowed for queue i.

3. *Collision handling*: If a queue experiences packet loss, possibly due to a collision, it needs to recalculate AF and adjust its contention window and backoff timer as follows:

$$CW[i] = \min(\lfloor CWIF[i] \times (CW\ [i] + 1) - 1 \rfloor, CW\max[i])$$

$$BT\ [i] = \text{uniform}(1, CW\ [i])\ .SlotTime,$$

   where $CWIF[i]$ is the Contention Window Increasing Factor, which depends on the aggressiveness factor of the queue *i*

$$CWIF[i] = 2 + AF[i].$$

4. *Successful Packet Transmission*: Upon successful transmission, a queue resets its AF to 0 and reduces its $CW[i]$ to $CW\min[i]$.

5. *Deferring*: To protect high-priority flows and improve fairness, a queue in our scheme recalculates its AF at the end of each deferring period.

V. SIMULATION RESULTS

The performance of NCA-EDCA was evaluated by the NS-3 network simulator [11]–[12] using different network topologies and channel loads. Each topology consists of client nodes sending three different flows (audio, video, and data) to a common server, as shown in Fig. 4.

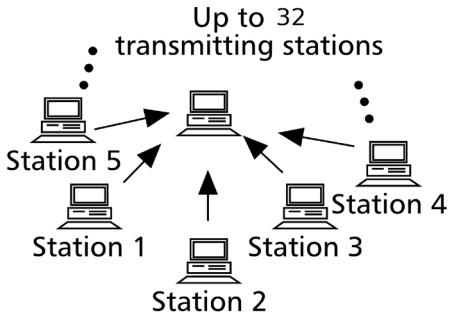


Fig. 4. Simulation topology.

The PHY data rate is set to 54 Mb/s, ACs and flow parameters (according to the IEEE 802.11e standard) are shown in Table I.

The simulations were conducted for 15 s and the results averaged over 5 runs with different flow start times.

Fig. 5 shows that NCA-EDCA significantly improves total goodput in high-stress situations (more than 600 % total goodput gain). In addition, Fig. 6 and Fig. 7 show that our scheme protects the goodput of voice and video flows in all network states, while with EDCA the goodput degrades rapidly under medium and high load conditions.

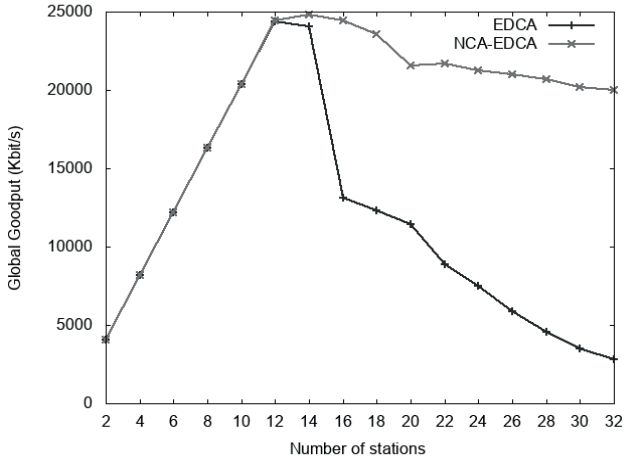


Fig. 5. EDCA and NCA-EDCA total goodput.

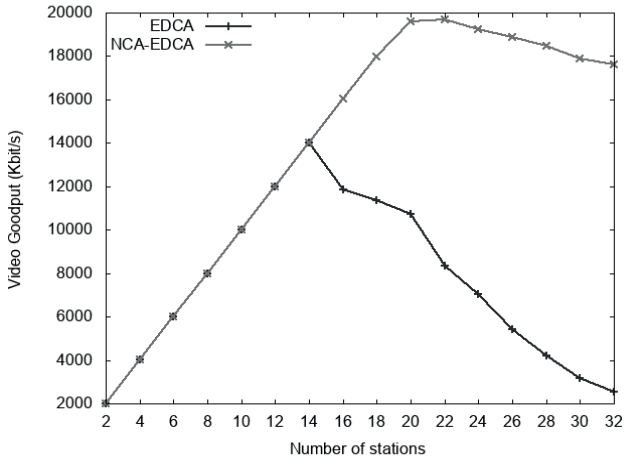


Fig. 6. EDCA and NCA-EDCA video goodput.

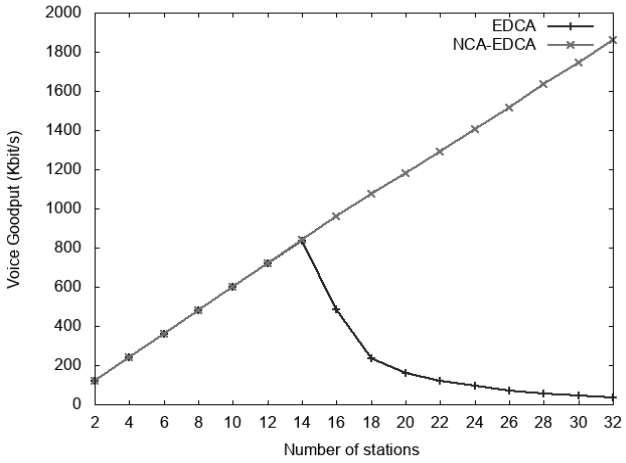


Fig. 7. EDCA and NCA-EDCA voice goodput.

Furthermore, we can see from Fig. 8 that the data flow goodput is also improved by up to 800 % in medium load conditions (around 16 nodes) and more than 100 % in high load conditions (more than 22 nodes).

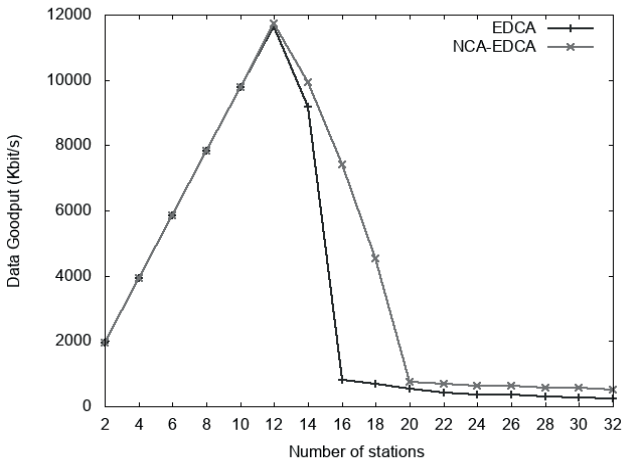


Fig. 8. EDCA and NCA-EDCA data goodput.

Figures 9 and 10 show the cumulative distribution function for packet delays in NCA-EDCA and EDCA, respectively, when the channel is 80 % loaded. In this high load case, we can see that although with NCA-EDCA 8 % of voice traffic experiences a slight delay compared to EDCA, 100 % of it falls under the voice quality service level agreement (SLA) [13] (≤150–300 ms). In addition, the cumulative fractions show that 96 % of video traffic is within the video quality SLA for NCA-EDCA, while EDCA meets the video quality SLA for only 31 %.

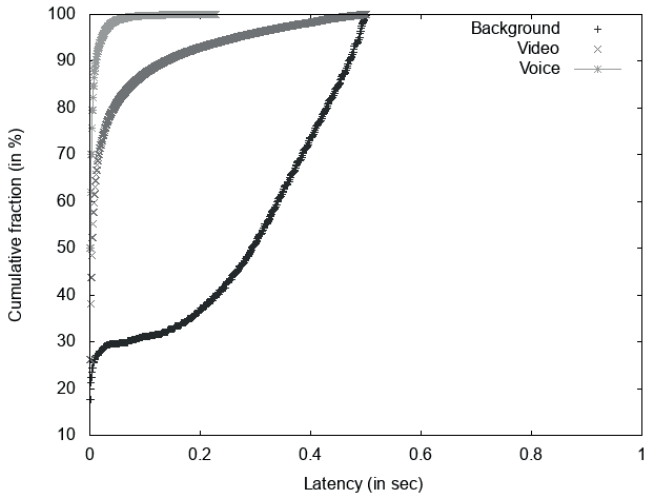


Fig. 9. CDF for NCA-EDCA.

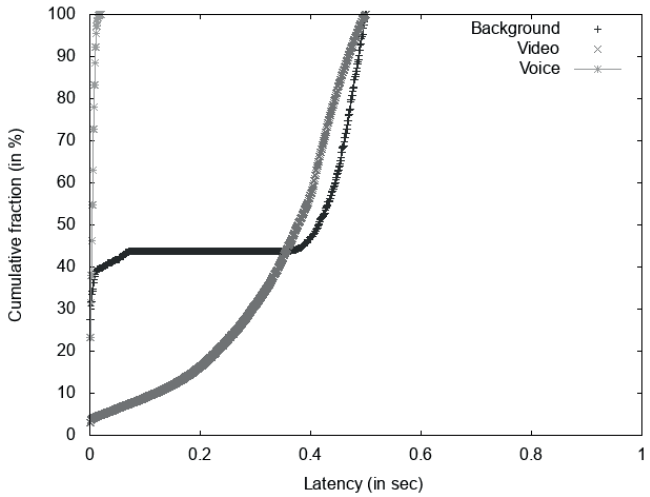


Fig. 10. CDF for EDCA.

As shown in Fig. 11, our scheme significantly reduces the number of collisions in both medium (80 % less) and high (65 % less) load states. This collision rate reduction is the key behind all the goodput and delay improvements.

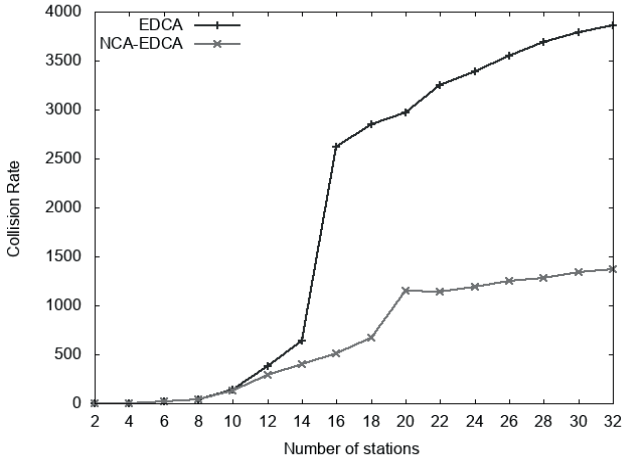


Fig. 11. Collision rate.

To assess the fairness of our scheme between flows of equal priority, we use the *fairness index* defined in [14] by:

$$FI = \frac{(\sum_{i=1}^{n} T_i)^2}{n \times \sum_{i=1}^{n} T_i^2},$$

where n is the number of flows with equal priority and T_i is the throughput of flow i. FI takes values from 0 to 1, with 1 representing the highest degree of fairness (if all T_i are equal). As in Fig. 12 and Fig. 13, our scheme shows a high level of fairness for voice and video flows compared to EDCA, especially at high loads. This is because the $CW[i]$ contention window size of the voice and video queues across all stations quickly reaches maximum values, although they grow differently. These queues almost always transmit in the same contention window. NCA-EDCA fairness is also better for background flows at low, moderate, and very high loads. In these situations, all queues transmit in roughly the same contention window (close to either $CW_{\min}[i]$ or $CW_{\max}[i]$). If the load is over moderate and not too high, the fairness of NCA-EDCA for background traffic will be affected mainly due to queues operating in different contention windows.

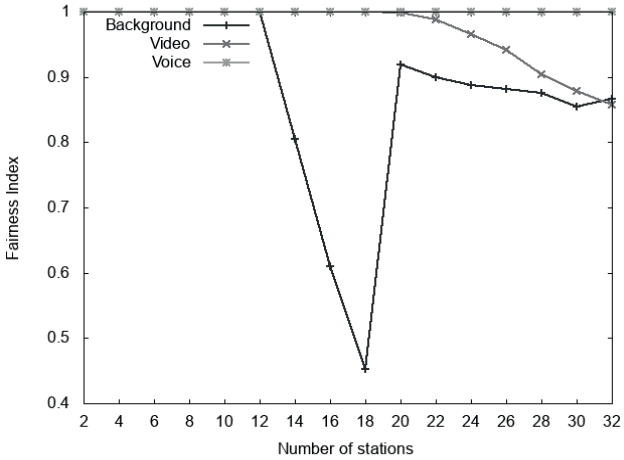


Fig. 12. Fairness of NCA-EDCA.

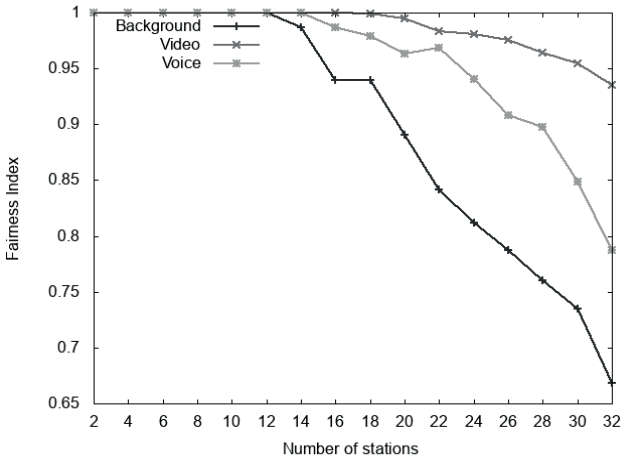


Fig. 13. Fairness of EDCA.

VI. Conclusion

In this study, we have introduced NCA-EDCA, a new QoS improved scheme for IEEE 802.11 ad-hoc wireless networks. Our contribution extends the basic EDCA scheme by using the time a packet spends in the queue as a key parameter to estimate the aggressiveness of the environment and using this information to dynamically parameterize the channel access function. Our scheme uses a differential slow decrease of the backoff timer whenever the environment becomes aggressive to reduce the collision rate and thereby increase the throughput of the various flows. It also protects the transmission of multimedia streams while maintaining fairness between those with the same priority by adjusting the contention window according to the aggressiveness of the environment. The simulation results show that NCA-EDCA offers higher overall throughput (more than 600 % gain) and very good multimedia flow performance at all channel loads. In addition, it offers a higher degree of fairness between voice and video flows.

References

[1] Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys and Tutorials, vol. 17, no. 4, pp. 2347–2376, Jun. 2015. https://doi.org/10.1109/COMST.2015.2444095

[2] M. Condoluci, G. Araniti, T. Mahmoodi, and M. Dohler, "Enabling the IoT machine age with 5G: Machine-type multicast services for innovative real-time applications," IEEE Access, vol. 4, pp. 5555–5569, May 2016. https://doi.org/10.1109/ACCESS.2016.2573678

[3] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2012, Mar. 2012. https://doi.org/10.1109/IEEESTD.2012.6178212

[4] S. Choi, J. D. Prado, N. Sai Shankar, and S. Mangold, "IEEE 802.11e contention-based channel access (EDCF) performance evaluation," in IEEE International Conference on Communications, vol. 2, 2003, pp. 1151–1156.

[5] J. Hui and M. Devetsikiotis, "A unified model for the performance analysis of IEEE 802.11e EDCA," IEEE Transactions on Communications, vol. 53, no. 9, pp. 1498–1510, Sep. 2005. https://doi.org/10.1109/TCOMM.2005.855013

[6] Sarr, C. Chaudet, G. Chelius, and I. G. Lassous, "Bandwidth estimation for IEEE 802.11-based ad hoc networks," IEEE Transactions on Mobile Computing, vol. 7, no. 10, pp. 1228–1241, Oct. 2008. https://doi.org/10.1109/TMC.2008.41

[7] Lindgren, A. Almquist, and O. Schelén, "Quality of service schemes for IEEE 802.11 wireless LANs – An evaluation," Mobile Networks and Applications, vol. 8, no. 3, pp. 223–235, Nov. 2003. https://doi.org/10.1023/A:1023389530496

[8] P. H. Gordon, D. Perdana, and R. Munadi, "Performance evaluation of VANET with EDCA access scheme using Gauss-Markov mobility model," International Journal of Simulation: Systems, Science and Technology, vol. 18, no. 2, pp. 1–7, Jun. 2017. https://doi.org/10.5013/IJSSST.a.18.02.01

[9] N. Ramos, D. Panigrahi, and S. Dey, "ChaPLeT: Channel-dependent packet level tuning for service differentiation in IEEE 802.11e," in Proc. International Symposium on Wireless Personal Multimedia Communications, Sep. 2003, pp. 86–90. https://www.researchgate.net/publication/2879980_ChaPLeT_Channel-dependent_Packet_Level_Tuning_for_Service_Differentiation_in_IEEE_80211e

[10] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks," in IEEE Wireless Communications and Networking Conference, WCNC, vol. 2, New Orleans, Louisiana, USA, Mar. 2003, pp. 1373–1378. https://doi.org/10.1109/WCNC.2003.1200574

[11] H. Park, S. Pack, and C. H. Kang, "Dynamic adaptation of contention window for consumer devices in WiMedia home networks," IEEE Transactions on Consumer Electronics, vol. 57, no. 1, Feb. 2011, pp. 28–34. https://doi.org/10.1109/TCE.2011.5735477

[12] M. Malli, Q. Ni, T. Turletti, and C. Barakat, "Adaptive fair channel allocation for QoS enhancement in IEEE 802.11 wireless LANs," in IEEE International Conference on Communications, vol. 6, Paris, France, Jun. 2004, pp. 3470–3475. https://doi.org/10.1109/ICC.2004.1313189

[13] "NS-3 network simulator," 2010. [Online]. Available: https://www.nsnam.org/

[14] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in Modeling and Tools for Network Simulation, K. Wehrle, M. Güneş, J. Gross, Eds. Springer, Berlin, Heidelberg., 2010, pp. 15–34. https://doi.org/10.1007/978-3-642-12331-3_2

[15] T. R. Henderson, M. Lacage, and G. F. Riley, "Network simulations with the ns-3 simulator," in SIGCOMM'08, Seattle, Washington, USA, Aug. 2008, p. 527. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://conferences.sigcomm.org/sigcomm/2008/papers/p527-hendersonA.pdf

[16] Lewis and S. Pickavance, Selecting MPLS VPN Services: A Guide to Using and Defining MPLS VPN Services. Cisco Press, 2006.

[17] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: an explanation," ATM Forum/99-0045, vol. 45, Feb. 1999. https://www.cse.wustl.edu/~jain/atmf/atm99-0045.htm



Soulimane Mammar is an Assistant Professor at the Polytechnic School of Oran – ENPO, Oran, Algeria. He received his engineering degree (1995) and Master degree (2002) in Computer Science from the University of Science and Technology of Oran. Since 2000, he has been involved in system and network administration, cloud infrastructure and security. He is now pursuing the PhD degree at the University of Oran 1, Oran, Algeria. His current research interests are in the area of wireless/mobile networks with emphasis on QoS guarantee, wireless LAN, next-generation mobile networks, data link layer protocols, routing protocols and security.
Address: Department of Electrical Engineering, Polytechnic School of Oran – ENPO, BP 1523 EL M'NAOUER ES-Senia Oran – Algeria.
ORCID iD: https://orcid.org/0000-0002-4407-0858
E-mail: soulimane.mammar@enp-oran.dz



Hafid Haffaf is a Professor at the University of Oran Es-Senia (Algeria). He actually heads the RIIR Laboratory at Computer science department –Oran University His research focuses on a variety of topics, including automatic control and diagnosis, optimization reconfiguration using matroid theory, system of system approaches, and their applications in bond graph and monitoring. He has many projects in collaboration with European laboratories, including Polytech Lille (where he worked in intelligent transport systems infrastructures) and LIAUPau (France) in the domain of Wireless Sensors
Address: Department of Computer Science, University of Oran1 – BP 1524 EL M'NAOUER ES-Senia Oran – Algeria.
ORCID iD: https://orcid.org/0000-0002-8249-4598
E-mail: haffaf.hafid@univ-oran1.dz

Mustapha Kamel Rahmouni received the PhD degree in operational research from Southampton University, UK, in 1987. He is a Professor at the University of Oran Es-Senia (Algeria). His areas of interest in research include formal specifications, information system specification approaches, and database reverse engineering.
Address: Department of Computer Science, University of Oran1 – BP 1524 EL M'NAOUER ES-Senia Oran – Algeria.
E-mail: archon.lightning@gmail.com