HAL
open science

Adaptive Negotiation for Block Acknowledgment Session Management

Kaoutar Abdelalim, Getachew Redieteab, Sandrine Roblot, Karine Amis

To cite this version:

Kaoutar Abdelalim, Getachew Redieteab, Sandrine Roblot, Karine Amis. Adaptive Negotiation for Block Acknowledgment Session Management. VTC2019-Spring : IEEE 89th Vehicular Technology Conference, Apr 2019, Kuala Lumpur, Malaysia. pp.1-5, 10.1109/VTCSpring.2019.8746503 . hal-02286135

HAL Id: hal-02286135

https://hal.science/hal-02286135v1

Submitted on 13 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Negotiation for Block Acknowledgment Session Management

Kaoutar Abdelalim, Getachew Redieteab, Sandrine Destouet Roblot
Orange Labs
Cesson-Sevigné, France
{kaoutar.abdelalim, getachew.redieteab, sandrine.roblot}@orange.com

Karine Amis
IMT Atlantique
Brest, France
karine.amis@imt-atlantique.fr

***Abstract*—The expansion of wireless applications in dense environments raises many technical issues. The 802 standards need to adapt and enhance the network quality by developing new technologies. The block acknowledgment (BA) mechanism was introduced in the IEEE 802.11e standard to improve medium access control (MAC) efficiency. It requires the exchange of many control frames to establish a session with each user, which turns into an issue for networks in dense environments as it causes increased overhead and latency. This paper deals with the optimization of the BA session management procedure. We propose a modified block acknowledgment session control mechanism which reduces the overhead and latency compared to the original one.**

***Index Terms*—IEEE 802.11ax, IEEE 802.11e, QoS, Block Acknowledgment, WLANs, Network Simulator 3, ns-3**

I. Introduction

Wireless technologies have experienced tremendous evolution when it comes to usage and applications. Each IEEE 802.11 [1] amendment addressed new points with regard to previous generations, in an effort to develop and revise some aspects manifesting drawbacks. In particular, the increase in the number of users calls for more performance (high efficiency, increased throughput, more energy...), especially in wide environments. As increasing the data rate is among the significant targets, the standards made valuable progress on several techniques that ensure higher data rates across the IEEE 802.11 generations. For instance, the maximum data rate is improved from 54 Mbps in 802.11a [2] to 6.93 Gbps in 802.11ac [3]. This increase was obtained through some modifications (e.g. multiplexing techniques and more spatial streams) but also through the introduction or the enhancement of several techniques namely orthogonal frequency division multiplexing (OFDM), single user multiple input multiple output (SU-MIMO), multiple user multiple input multiple output (MU-MIMO), to mention a few. This contributes a lot in the later systems quality improvement, even in data rates. As for the connectivity and the capacity of the network, they remain very substantial points as well, especially for the last standard IEEE 802.11ax [10]. The IEEE 802.11 committee launched the Task Group (TG) TGax to work on improving the network, in the case of a dense environment, to offer higher performance and insure an optimal throughput for each user. The 802.11ax amendment is based on the IEEE 802.11ac-2013 [3] [4] [5]. It thus proposes quite a few techniques such as including uplink MU-MIMO (UL MU-MIMO) and orthogonal frequency division multiple access (OFDMA) techniques, among other improvements [7] [8] [9], to reduce network impairments. The block acknowledgment (BA) mechanism has been maintained by the 802.11ax amendment. It increases the communication efficiency compared to simple acknowledgment. This work joins some of 802.11ax objectives and aims to improve user experience within a crowded network. For that, this paper focuses on the BA mechanism in particular, by proposing an optimized BA session negotiation that contributes to enhancing throughput. This is depicted using the file transfer protocol (FTP) over user datagram protocol (UDP) application since it is one of standard network protocols used to transfer computer files between a client and server on a computer network. For our simulation the network simulator 3 (ns-3) [11] is exploited. The paper is organized as follows. Section II presents the original BA mechanism. Section III describes the proposed adaptive negotiation of BA sessions. Section IV is dedicated to simulation results and the proposed mechanism is compared to the original one. Conclusions and perspectives are presented in Section V.

II. Problem statement

A. Block acknowledgment mechanism

The BA has been introduced with the 802.11e [12] amendment for the purpose of improving quality of service (QoS). It enables to notify the successful reception of the preceding block of data frames while the simple acknowledgment mechanism notifies frame per frame. Such a scheme thus improves the transmission efficiency and enhances throughput [13]. In this paper we measure the transmission efficiency by the correct reception of transmitted data. The simple acknowledgment mechanism enables an individual acceptance on a per-frame basis. One drawback is the overhead increase and thus the latency of transmissions, which is a real issue for a high number of users. With the BA mechanism, instead of acknowledging each frame one by one, a summary of the reception status of a block of frames is sent through a single frame called BA frame. The number of sent frames within a block is called block size and is mainly limited by the storage capacity of the receiving station. There are two types of BAs: immediate BA and delayed BA. The later high throughput (HT) 802.11n amendment [5] [14] [15] introduced HT-immediate BA and

978-1-7281-1217-6/19/$31.00 ©2019 IEEE

HT-delayed BA, in order to improve MAC efficiency and exploit aggregation and higher data rates benefits. In very high throughput (VHT) 802.11ac [3], VHT stations could support HT-immediate BA and HT-delayed BA (since VHT stations are also HT stations). The original, i.e. HT, mechanisms have been maintained in all amendments following 802.11n to support interoperability with legacy stations. The immediate BA was initially intended for applications with low-latency constraints while delayed BA rather applied to applications without strong latency requirements. This article particularly focuses on the more widespread immediate BA. As presented in Fig. 1 the BA mechanism relies on three main steps.

1) Session setup: The originator first sends an add BA request (ADDBA Request) frame to establish the BA session which includes a traffic identifier (TID) signaling the access category of the session under-negotiation to the recipient. The ADDBA Request indicates also the BA policy and the block size recommended by the originator (it's up to the recipient to set the buffer size that shall be used). If the recipient does not operate with this BA policy, it can refuse and the BA session won't be established. Furthermore, the two parts concur to use aggregated MAC service data units (A-MSDUs) or not, through an A-MSDU supported subfield in the ADDBA Request and the ADDBA Response. If the ADDBA Request is correctly received, the recipient answers with an acknowledgment (ACK) frame and sends in turn an ADDBA Response. The originator answers with an ACK to confirm ADDBA Response reception. The acceptance/refusal of the session is declared within the ADDBA Response. The originator and the recipient continue sending ADDBA Requests or ADDBA Responses until the four way handshake is successfully completed. If the inactivity timeout (also signaled during the exchange) is reached, the session fails.

2) Data exchange: The second step begins once the BA session is accepted/set up. The originator sends a block of QoS Data frames separated by short inter frame space (SIFS) or as part of an aggregate MAC packet data unit (A-MPDU). The BA frame can simultaneously acknowledge up to 64 consecutive MAC protocol data units (MPDUs) which are not fragments. The recipient has to prepare a BA according to the scoreboard saved before (a bitmap of 64 bits which forms an array of indices starting from the starting sequence number). Then it reassembles any complete MPDUs from buffered preceding MPDUs and transfers them up to its higher layer. If an MPDU was indicated as not received in the block acknowledgment map, the originator should retransmit the corresponding QoS Data frame until its lifetime inactivity limit is reached. If at least one QoS Data was successfully transmitted, the acknowledgment would be effective.

3) Tearing down the BA session: Once there is no more data to send, the last step begins when the originator sends a delete BA (DELBA) to end the session. If it is successfully received, an ACK is sent by the recipient and the BA session is closed.

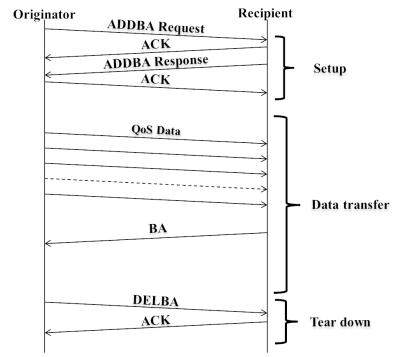

Fig. 1. Immediate block acknowledgment

B. Block acknowledgment session management limitations

As it can be noticed, the BA session management mechanism includes several control frames. As for the first step, the exchange of ADDBA Request and ADDBA Response is mandatory to establish the BA session. Likewise for tearing down the session, the recipient needs to send a DELBA frame and waits for an ACK to cancel the session. This is obviously a limiting factor for high number of users since a BA session needs to be established for each user (and per access category) and has to rely on all these control frames. This induces more overhead and lowers the MAC efficiency of the network. Moreover, the BA buffer size cannot be changed immediately but needs an additional ADDBA Request/Response exchange to tune its value. This definitely increases latency within the session. For these reasons we develop a new proposition as a modified BA session control mechanism that enables a negotiation of block size of data and relies on a more optimized method.

III. Proposed solution

A. Operational principle

Our proposition is an optimization of the BA session control mechanism. The idea is to reduce the overhead by introducing two bits tag instead of sending several session control frames. In what follows, these two bits will be named the "control bits" and labeled b_1 and b_0 for request bits and $b_1^{'}$ and $b_0^{'}$ for response bits. The AN-BA relies also on three main steps as follows.

1) Session setup: To set the session up, the ADDBA Request/Response exchange is replaced by a QoS Data frame tagged with these control bits (i.e. b_1 and b_0). The originator suggests the recipient establishing a session which can be accepted or refused through the control bits included in the ACK (i.e. $b_1^{'}$ and $b_0^{'}$).

2) Data exchange: If the recipient accepts the session, the originator begins sending a bulk of QoS Data frames tagged with the control bits. The different states of the bits along with the corresponding meaning are listed in Table I. Depending on the control bit states, our solution includes an implicit negotiation involving the originator and the recipient. This negotiation relies on adapting the number of sent frames from

TABLE I
CONTROL BIT STATES

	Originator			Recipient		
Session steps	b_1	b_0	**Case**	$b_1^{'}$	$b_0^{'}$	**Case**
Session setup	0	0	Nothing to do	0	0	Nothing to do/ Timeout
	0	1	Reserved	0	1	Refuse the request
	1	0	Initiate without A-MSDU	1	0	Accept the request
	1	1	Initiate with A-MSDU	1	1	Refuse for incompatibility
Established session	0	0	Nothing to change	0	0	Reduction request
	0	1	Halve the buffer size	0	1	Refuse buffer increase / Timeout if reduction
	1	0	Double the buffer size	1	0	Accept the modification
	1	1	Close the session	1	1	Tear down request

the originator to the recipient. The mechanism allows the originator to call for increasing or reducing the number of QoS Data MPDUs using different control bit states. The recipient can accept or refuse the requested modification. The procedure is illustrated in Fig. 2. The originator sends a QoS Data frame including control bits to invite the recipient to establish the session. Using a specific state (i.e. b_1b_0 = 10 or b_1b_0 = 11) the originator announces to the recipient which configuration will be used for the data transmission: whether it is on non-aggregated frame or aggregated frames. The recipient should accept or refuse the request depending on its capacity of handling A-MSDUs. The recipient receives the QoS Data frame tagged with control bits and responds by an also tagged ACK, informing its acceptance ($b_1^{'}b_0^{'}$ = 10) or refusal ($b_1^{'}b_0^{'}$ = 01 or $b_1^{'}b_0^{'}$ = 11). When the recipient accepts to establish the session, the data transfer session begins. The originator sends a block of QoS Data frames with control bits indicating the type of request. The originator can for example ask to double the number of transmitted frames (b_1b_0 = 10). At recipient side, the increase request could be accepted or refused. If it is accepted, a BA is sent to acknowledge the reception of the previous frames with the control bits announcing the acceptance ($b_1^{'}b_0^{'}$ = 10). A similar procedure applies for the refusal ($b_1^{'}b_0^{'}$ = 01 or $b_1^{'}b_0^{'}$ = 11). Since the recipient buffer size is limited, it can call for a reduction ($b_1^{'}b_0^{'}$= 00) so that the originator would halve the number of transmitted QoS Data frames. If the recipient accepts the increase, the originator sends a block of QoS Data frames up to the new size. If not, it continues with the same size. The control bits are tuned depending on the state of the transmission.

3) Tearing down the BA session: The recipient can ask to stop the session by sending suited control bits in the last BA ($b_1^{'}b_0^{'}$=11). The originator takes the last tagged BA and the closing session information within the next transmission.

B. Advantages of proposed solution

As stated earlier, the classical BA session management mechanism includes several control frames for establishing a session. This induces more latency in crowded environment and consumes bandwidth. For instance to restore the buffer size value, both parts (originator and recipient) must exchange ADDBA Request/Response one more time. The proposed method improves the efficiency of networks by enabling dynamic negotiation of BA session parameters, namely buffer size, through two control bits instead of dedicated control frames. Such a mechanism permits to tune the channel resources according to the corresponding traffic.

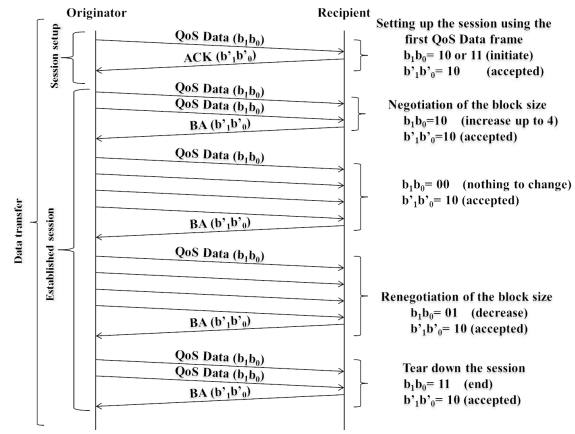

Fig. 2. Example of the adaptive negotiation for BA session management

We shall note that also for the sake of enhancing network efficiency, V. Pitchaiah proposed in [16] an improved BA setup mechanism. Despite the fact that it improves the session establishment efficiency, it has a disadvantage of reserving unjustified resources. This could lead to a failed negotiation and hence increase the transmission delay. On the contrary, our method proposes to negotiate the buffer size based on the available resources and to re-adapt it during the session, which is a convenient advantage with regard to network performances.

IV. SIMULATION

To compare the performance of the proposed adaptive negotiation of BA session (AN-BA) with the classical mechanism one (BA), we use the network simulator ns-3. Several modifications have been made within ns-3 modules to cover the functioning of our method. Precisely, a new class has been added representing the control bits tag with many modifications within MacLow, EdcaTxop, and BlockAckManager modules. Instead of exchanging ADDBA Request/Response to establish a session, or a DELBA to delete it, the management is made with the aid of the new control bits tag class. Furthermore, we introduce an updating aspect where the buffer size can change depending on originator and recipient requirements to bring out the impact of the AN-BA compared to BA mechanism. Every T_{update} of the duration transmission, an updating of the buffer size can occur. The evaluation is made on the base of different networks inspired from IEEE 802.11ax scenarios [17] under IEEE 802.11ac settings.

A. Simulation scenarios

For the performance analysis, we consider 5 and 10 stations within an apartment to model a medium density case. The stations are connected to a single access point (AP). For a high density environment, we choose 2 architectures. The first one is a building with 3 floors and 3 apartments per floor. Each apartment includes 4 stations. The second one is a ground

floor including 5 apartments with 5 stations in each one. In every apartment all stations are connected to a single AP. For simplicity reasons, all nodes have a random and fixed position during the whole simulation duration. All stations are operating with a same physical layer (PHY) data rate under the very high throughput (VHT) modulation scheme channel (MCS) 7 withing 20MHz bandwidth and using 2 spatial streams. Each scenario exploits six maximum buffer size values (2, 4, 8, 16, 32, and 64) to cover multiple cases of the buffer size. As mentioned before, the method will be analyzed by using uplink FTP over UDP file transfers so as not to be impacted by the limitation in AP traffic queues. Network arrivals are modeled as Poisson process with rate λ. The λ parameter specifies the intensity of packet arrivals. To analyze the behavior of both procedures we fix λ to 0.8 which is a quite high arrival rate and a convenient case to be studied. The simulation duration is fixed to 10 seconds with an updating of the buffer size every 100 milliseconds. Transmitted FTP packet size is 512 kilo Bytes. We consider the average throughput, the FTP packets outage ratio, and the average delay within FTP sessions as comparison metrics. The depicted metrics have been obtained by averaging over 50 independent runs.

B. Simulation results

The following figures give an overview of the throughput behavior, the maximum ratio of non-transmitted FTP packets, and the average delay within a session. As mentioned earlier, we distinguish the behavior of the two mechanisms under medium and high density networks.

1) Non dense environment: Fig. 3, Fig. 4 and Fig. 5 depict the behavior of the two methods in a non dense environment. We can easily notice that the gap between the two mechanisms is growing by doubling the number of users. For the case of 5 stations, BA and AN-BA represent approximately the same behavior for packet outage rate and average delay. However, the maximum throughput within the AN-BA procedure is slightly above that of the basic mechanism (39.8 Mbps vs 38.03 Mbps). Meanwhile the difference is more significant for 10 users especially when it comes to throughput (26.74 Mbps vs 24.8 Mbps) and average delay (148 ms vs 206 ms).

2) Dense environment: The high density case is shown in Fig. 6, Fig. 7 and Fig. 8. For this case, the AN-BA method presents considerable improvement compared to the BA mechanism. We notice from Fig. 8 that the average throughput over an AN-BA session is well above the average throughput for BA mechanism whatever the architecture tested (25 or 36 stations). FTP packet outage is more frequent within the BA mechanism than the AN-BA method. We can also notice from Fig. 6, that for 36 stations, there is 41% of lost packets in the BA mechanism compared to 34% for the AN-BA method for a buffer size of 64. This explains the crossover points in case of 36 stations starting from 32 buffer size value (Fig. 7). While the packets are not transmitted, transmitting stations exploit the opportunity to send all the remaining FTP packets which consequently increases the throughput to 1.52 Mbps at

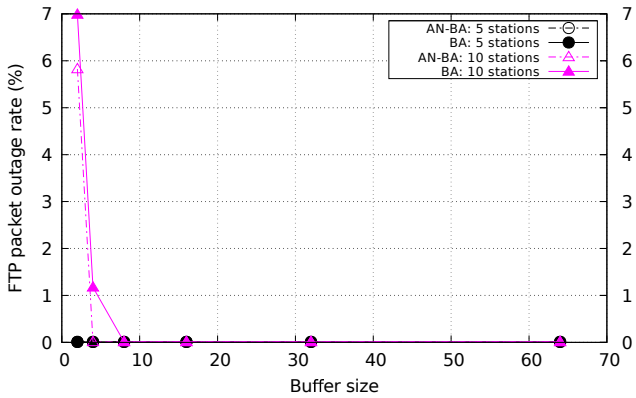

Fig. 3. FTP packet outage rate within FTP over UDP sessions for $\lambda = 0.8$ as a function of the buffer size for 5 and 10 stations

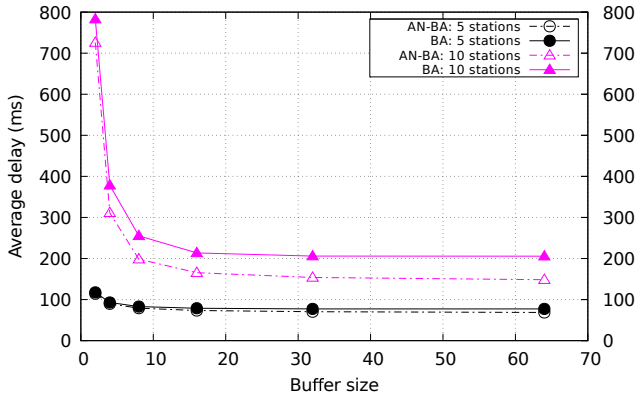

Fig. 4. Average delay within FTP over UDP sessions for $\lambda = 0.8$ as a function of the buffer size for 5 and 10 stations

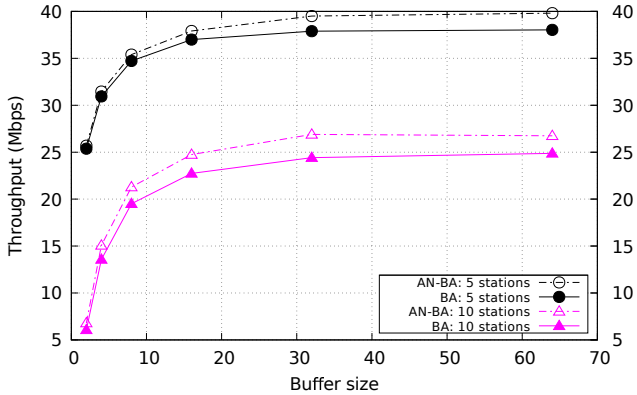

Fig. 5. Average throughput within FTP over UDP sessions for $\lambda = 0.8$ as a function of the buffer size for 5 and 10 stations

maximum buffer size value (Fig. 8), hence the BA and AN-BA achieve same performances. On the other hand stations under AN-BA method managed to transmit more packets while maintaining a higher throughput of 1.58 Mbps. Additionally, we can notice that the throughput in all cases is almost the same between 32 and 64 buffer size values. This is due to the ns-3 constraint of sending PHY packet data units (PPDUs) of

at most 5 milliseconds.

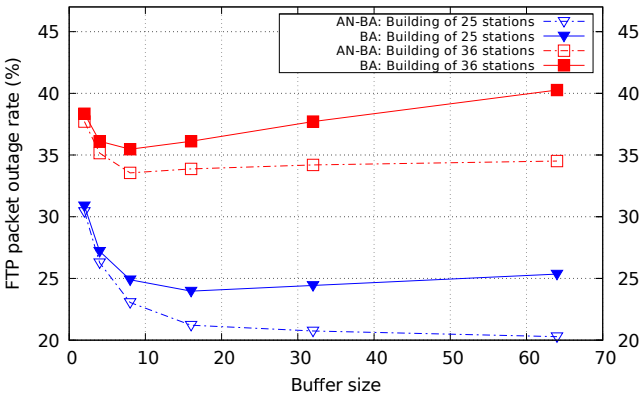

Fig. 6. FTP packet outage rate within FTP over UDP sessions for $\lambda = 0.8$ as a function of the buffer size within a building of 25 and 36 stations

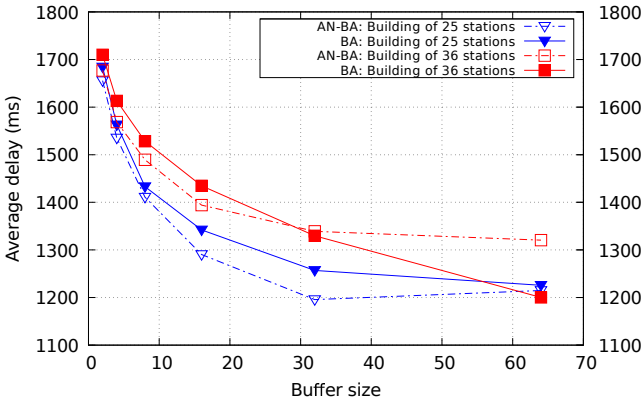

Fig. 7. Average delay within FTP over UDP sessions for $\lambda = 0.8$ as a function of the buffer size within a building of 25 and 36 stations

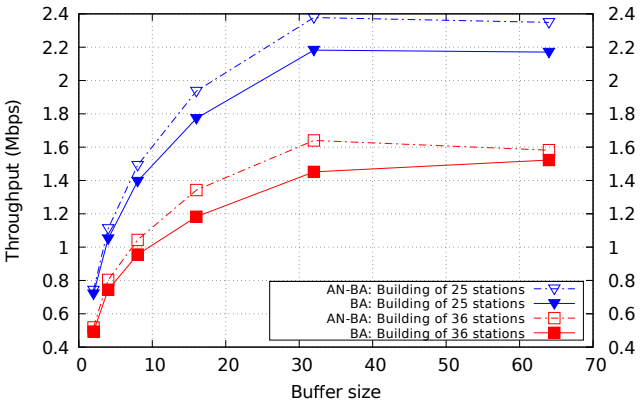

Fig. 8. Average throughput within FTP over UDP sessions for $\lambda = 0.8$ as a function of the buffer size within a building of 25 and 36 stations

Similarly, the AN-BA method demonstrates same improvements either for throughput, average delay, or FTP packet outage rate when changing T_{update}. The gap between the two mechanism grows with the decrease of the T_{update}.

V. Conclusion

This paper exposes an adaptive negotiation of the block acknowledgment session which is an evolution of the existent block acknowledgment mechanism proposed in IEEE 802.11e. It consists in an optimal and flexible way of negotiating the buffer size within a transmission. We showed in this paper that the method enhances performance especially for high number of users. AN-BA simulation results demonstrate that the packet outage rate is reduced by 17% in some configurations compared to the BA mechanism, while ensuring higher average throughput and lower latency. Meanwhile, the AN-BA could be supported by the IEEE 802.11ax. For instance, the originator could insert the control bits within one of the HE-A-Control fields where some bits are so far empty. At the reception side, the recipient could include them instead of two null bits within ACK or BA. Likewise, one should note that backward compatibility should be maintained with regard to the possibility of inserting the two bits within BA, ACK, and QoS data. Future work will involve the analytical framework of the method compared to the original one, independently of the application.

References

[1] 802.11-2012, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Mar. 2012.

[2] 802.11a-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz band, Sep. 1999.

[3] 802.11ac-2013, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz, Dec. 2013.

[4] M. Gast, 802.11ac: A survival Guide, O'Reilly Media, Aug. 2013.

[5] E. Perahia and R. Stacey, Next Generation wireless LANs: 802.11ac and 802.11n, Cambridge University Press, May 2013.

[6] B. Bellalta and K. Kasek Szott, "AP-initiated Multi-User Transmissions in IEEE 802.11ax WLANs," arXiv, 1702.05397v1, submitted on Feb. 2017.

[7] M. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi," IEEE Wireless Communications, vol. 24, issue 3, Jun. 2017.

[8] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANs, " IEEE Wireless Communications, vol. 23, issue 11, Feb. 2016.

[9] E. Khorov, A. Kiryanov, and A. Lyakhov, "IEEE 802.11ax: How to Build High Efficiency WLANs," International Conference on Engineering and Telecommunication (EnT), pp. 14-19, Nov. 2015.

[10] IEEE P802.11ax/D1.3, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Enhancement for High Efficiency WLAN", Jun. 2017.

[11] nsnam, "The ns-3 network simulator," 2017. [Online]. Available: https://www.nsnam.org/.

[12] 802.11e-2005, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, Nov. 2005.

[13] I. Tinnirello and S. Choi, "Efficiency analysis of burst transmissions with block ACK in contention-based 802.11e WLANs," IEEE International Conference on Communications (ICC), vol. 5, pp. 3455-3460, May 2005.

[14] 802.11n-2009, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 5: Enhancements for Higher Throughput, Oct. 2009.

[15] M. Gast, 802.11n: A survival Guide, O'Reilly Media, Apr. 2012.

[16] V. Pitchaiah, "Improved Block Acknowledgment Setup Mechanism," U.S. Patent WO2017030723, Oct. 2017.

[17] IEEE 802.11, doc.:IEEE 802.11-14/0980r16, "TGax Simulation scenarios," Jul. 2015.