Implementation of IEEE 802.11e Block Acknowledgement Policies

Orlando Cabral, Alberto Segarra, Fernando J. Velez, *IAENG, Member*

***Abstract*— Optimisation of IEEE 802.11e MAC protocol performance can be performed by modifying several parameters left open in the standard, like buffer size and acknowledgement policies. In this work, an event driven simulator was developed, and a comparison of the results between the case of standalone services and several services supported simultaneously was addressed. With only a single service, higher values for the goodput are obtained for video and background (BK) traffics, while the lowest is found for the voice application. The number of supported users is higher for voice. With mixtures of traffic, when the number of station is small the goodput is lower for voice. However, for higher number of station, the lowest values of the goodput occur for background traffic. By using the Block Acknowledgement (ACK) procedure, for video and BK traffics, the capacity is improved when the number of station is equal or higher than 16 and 12, respectively. From a detailed analysis of the video application, a reduction of more than 40% was achieved in the delay.**

***Index Terms*— Simulation, IEEE 802.11e, Block ACK, EDCA.**

I. Introduction

Recent years have seen an immense growth in the popularity of wireless services and applications that require high throughput. To support such growth, standardization bodies such as the IEEE have formed task groups to investigate and standardize features providing increased quality of service and higher throughputs for IEEE 802.11. One such extension is the block acknowledgment (ACK) policy feature, included in the ratified IEEE 802.11e amendment [1]. This feature improves system throughput by reducing the amount of overhead required by a station to acknowledge a burst of received traffic.

This work aims to provide the specification of a MAC layer simulator that evaluates the service quality in WiFi in order to enable simulations accounting for inter-working with Worldwide Interoperability for Microwave Access (WiMAX) and High Speed Downlink Packet Access (HSDPA), in the context of the IT-MOTION tool [2]. Its focus is on the policies for block acknowledgement (ACK) in IEEE 802.11e, and the final objective is to improve scheduling algorithms and to tune up several parameters like the size of the contention window the inter frame spacing, the size of transmission opportunities, and the buffer size.

The remaining of the paper is organized as follows. In Section II, a brief introduction to the IEEE 802.11e standard is presented along with the rationale behind the block acknowledgement procedure. Section III presents the state transition diagram of the simulator, and the list of events. In Section IV, details are given on the physical layer of the IEEE 802.11a standard, the one considered in this work. Section V discusses the hypothesis for system and scenarios, including details on traffic parameters. Section VI presents simulation results obtained for several scenarios in the absence and presence of the block acknowledgement procedure. Finally, conclusions are presented in Section VII as well as suggestion for future work.

II. IEEE 802.11e

A. User priorities and Access Categories

IEEE 802.11e provides medium access control (MAC) enhancements to support local area network (LAN) applications with QoS requirements in wireless environments. By using eight different user priorities (UPs), the so-called enhanced distributed channel access (EDCA) provides differentiated, distributed access to the medium for stations by using four different access categories (ACs) that provide support for the delivery of traffic with various UPs at the stations. The AC is derived from the UPs as presented in Table I. This differentiation is achieved by varying the following different values for the UP: i) the amount of time a STA senses the channel to be idle before backoff or transmission, ii) the length of the contention window to be used for the backoff, or iii) the duration a STA may transmit after it acquires the channel.

TABLE I. Mapping between user priorities and ACs.

	UP (Same as 802.1D user priority)	802.1D Designation	AC	Designation
Lowest	1	BK	BK	Background
	2	—	BK	Background
	0	BE	BE	Best Effort
	3	EE	BE	Best Effort
	4	CL	VI	Video
	5	VI	VI	Video
↓	6	VO	VO	Voice
Highest	7	NC	VO	Voice

Manuscript received March 20, 2008. The authors would like to acknowledge the fruitful contributions given by António Grilo for the implementation of the PHY part of the simulator. This work was partially funded by CROSSNET (a Portuguese Foundation for Science and Technology POSC project with FEDER funding), by IST-UNITE, by the FCT PhD grant SFRH/BD/28517/2006, and by "Projecto de Re-equipamento Científico" REEQ/1201/EEI/ 2005.

Orlando Cabral, Alberto Segarra, and Fernando Velez are with the Instituto de Telecomunicações, Universidade da Beira Interior, Covilhã, Portugal (phone: +351 275329953; fax: +351 275329972; e-mail: orlandoc@ubi.pt, vyniard@gmail.com, fjv@ubi.pt).

Details on the CSMA/CA protocol, transmission opportunities (TXOP), and inter-frame spaces (IFS) are presented in [1]. The backoff time and the backoff procedure are addressed in [1], [3], as well as the description of the details on the network allocation vector (NAV), and the use of RTS/CTS with fragmentation.

B. Block Acknowledgment (Block ACK)

The Block ACK mechanism improves channel efficiency by aggregating several acknowledgments into one frame. There are two types of Block ACK mechanisms: immediate and delayed. While immediate Block ACK is suitable for high-bandwidth, low-latency traffic, the delayed Block ACK is suitable for applications that tolerate moderate latency. The QSTA (with data to be sent) using the Block ACK mechanism is referred to as the originator, whereas the receiver is called the recipient.

The Block ACK mechanism is initialized by an exchange of Add Block Acknowledgment (ADDBA) Request/ Response frames. After initialization, blocks of QoS data frames can be transmitted from the originator to the recipient. A block may be started within a polled TXOP or by winning EDCA contention. The number of frame in the block is limited, and the amount of state that is to be kept by the recipient is bounded. The MAC packet data units (MPDUs) within the block of frames are acknowledged by a Block ACK control frame, which is requested by a BlockReq control frame. Fig. 1 illustrates the message sequence chart for the setup, data and the Block ACK transfer, and the teardown of the Block ACK mechanism. More details can be found in [4].

III. STATE TRANSITION DIAGRAM

The state transition diagram used to build the simulator is presented in Fig. 2. The following events cause transition/ change of the machine state:

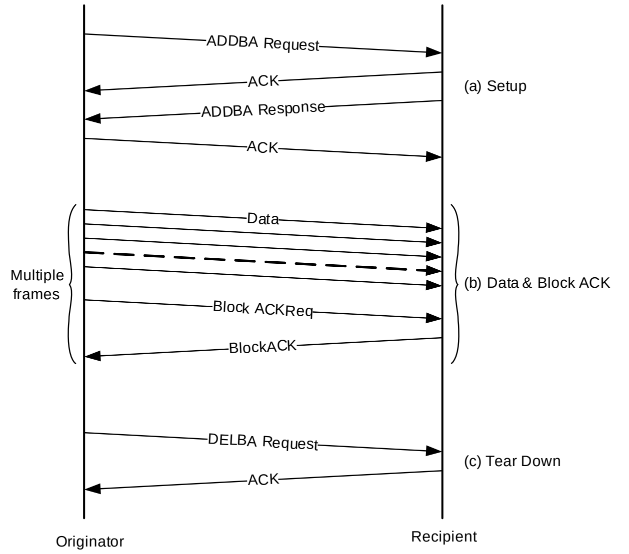

Figure 1. Block ACK sequence.

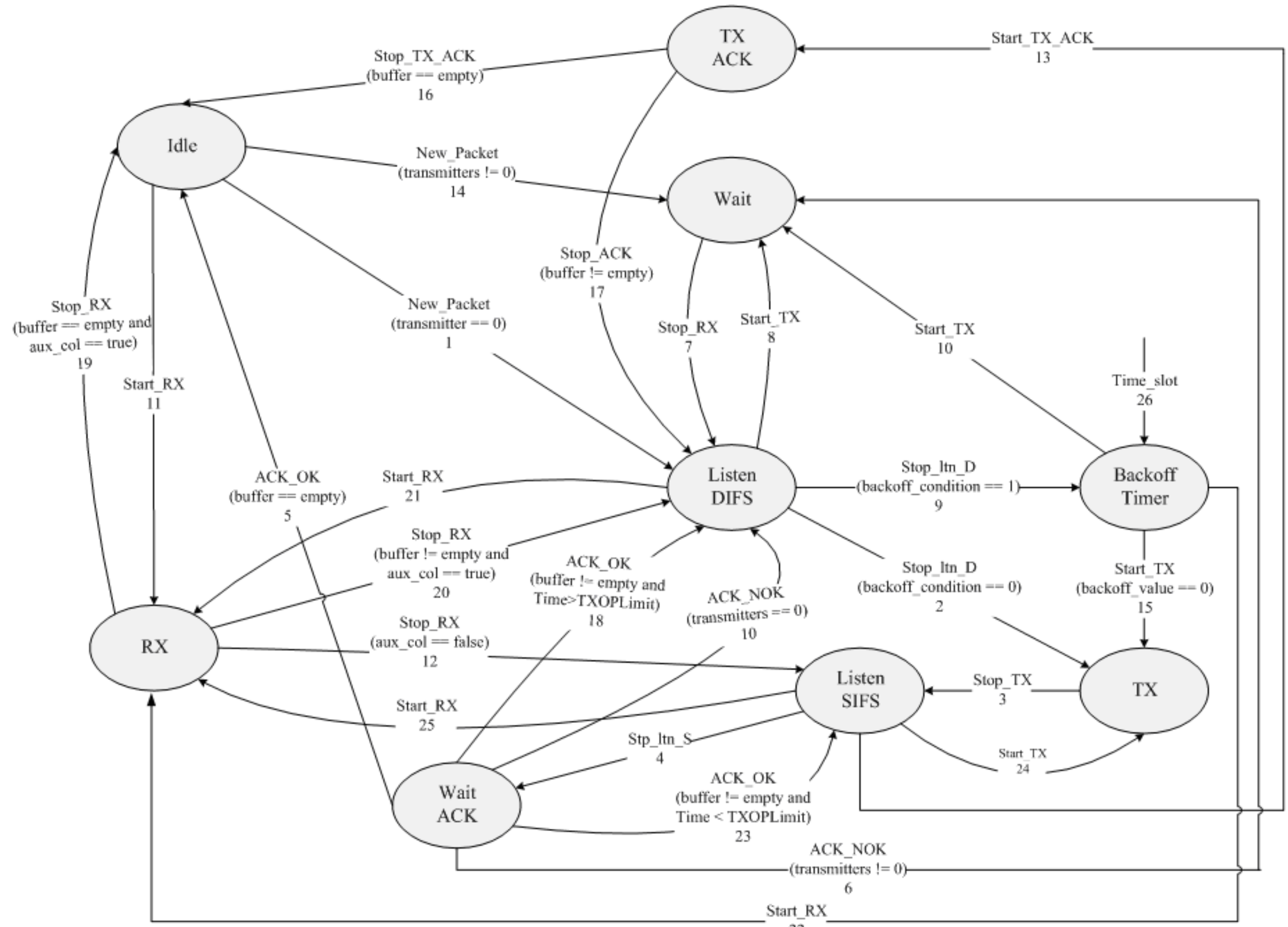

Figure 2. State transition diagram (the incoming arrow for the Backoff_Timer state means a transition from and to the same state).

NEW_PCK_BK a new packet of BK is generated
NEW_PCK_BE a new packet of BE is generated
NEW_PCK_VI a new packet of VI is generated
NEW_PCK_VO a new packet of VO is generated
STOP_LTN_DIFS end of the AIFS period for sensing the medium
STOP_LTN_SIFS end of the SIFS period for sensing the medium
TIME_SLOT the STA decrements the *backoff_value*
START_TXthe station starts to transmit
STOP_TX end of the transmission
START_RX begin of the reception
STOP_RX end of the reception
ACK_OK the ACK was received
ACK_NOK the ACK was not received

A detailed description of the possible states for the "machine" nodes, the simulation entities, the simulation variables, and the functions for events is given in [5].

IV. PHYSICAL LAYER

In this work we consider the IEEE 802.11a physical layer, PHY, specification [6], [7]. It defines an Orthogonal Frequency Division Multiplexing (OFDM) based PHY that operates in the 5 GHz frequency bands, and is able to achieve bit-rates up to 54 Mbps.

IEEE 802.11a specifies 8 different transmission modes, Table II obtained with different combinations of modulation and convolutional code rate. Each transmission mode corresponds to a different bit-rate. In Table II, the respective number of byte transmitted in one OFDM symbol (Bytes-per-Symbol, BpS). The convolutional encoder always encodes data with code rate 1/2. The 3/4 and 2/3 codes are derived from the original 1/2 code by a technique called puncturing. Puncturing is a procedure for omitting some of the encoded bits in the transmitter, and inserting a dummy "zero" metric into the convolutional decoder at the receiver, in place of the omitted bits. This technique is a simpler and more efficient way of generating a higher code rate.

In our simulations, we estimate the received power and SINR experienced at the receiver, based on the last reception of that machine, a procedure very similar with the one from [8]. In this algorithm, the sender chooses the bit-rate that achieves the highest goodput taking into account the SINR estimate. More details on the implementation of PHY into the simulator can be found in [9].

TABLE II. IEEE 802.11A PHY MODES.

Mode	Modulation	Code Rate	Bit-rate	BpS
1	BPSK	1/2	6 Mbps	3
2	BPSK	3/4	9 Mbps	4.5
3	QPSK	1/2	12 Mbps	6
4	QPSK	3/4	18 Mbps	9
5	16-QAM	1/2	24 Mbps	12
6	16-QAM	3/4	36 Mbps	18
7	64-QAM	2/3	48 Mbps	24
8	64-QAM	3/4	54 Mbps	27

V. SYSTEM, SCENARIO AND ASSUMPTIONS

Lets consider a cellular WiFi system where each cell has a set N+1 IEEE 802.11e stations communicating through the same wireless channel. While station 0 is the Access Point or QoS Access Points (QAP), the other N wireless terminals are QoS stations (QSTA). The propagation time is assumed to be absorbed by some mechanisms of the IEEE 802.11, like SIFS. Each station has four buffers whose size depends on the kind of service being dealt in order to guarantee a given value for the goodput (payload of the packet), and there is a correspondence between the buffer and the ACs.

This buffer will be filled with a MSDUs generated that characterises the service being dealt in it. If the MSDU is larger than a fragmentation threshold, it will be fragmented. In order to cope with service quality the packet transmission follows the EDCA (Enhanced Distributed Channel Access) IEEE 802.11e MAC procedure. Due to collisions or interference a packet may not be correctly received. The number of collision is represented by a global variable that checks whether there is more than one user transmitting simultaneously. The interference issues are addressed by using a radio propagation model. Each packet exits the buffer only after the reception of an acknowledgement, or if it has suffered more than a collision threshold.

In this first phase, the users are assumed to be static, and are distributed uniformly in a square area of 2500 square meter. The topology to be implemented consists of several wireless stations and an AP (Access Point). Three types of traffic sources were chosen, namely VO (high priority voice), VI (medium priority video), and low priority FTP data (as background traffic, BK). The traffic sources parameters are presented in Table III, as well as the ACs (access categories) of each type of traffic.

TABLE III. TRAFFIC PARAMETERS [6].

AC	Voice (VO)	Video (VI)	Background (BK)
Packet size	1280 bit	10240 bit	18430 bit
Packet interval	20 ms	10 ms	12.5 ms

VI. INITIAL SIMULATION RESULTS

A. Standalone services

Our simulations consider one access point with several client machines. Results include packet delay in miliseconds, goodput, in bits per second, and channel utilization in percentage. The results consider the simultaneous requirements from MAC and PHY layers. For each parameter, they are obtained as the average of 20 simulations (each with different random seeds). Simulation results with MAC layer alone were presented in [5].

Packet delay is the period of time between the instant at which the packet arrives to the buffer and the instant at which the packet is successfully transmitted. Fig. 3 presents results for delay for stand alone voice (VO), background (BK) and video (VI). In terms of grade of service, the voice application supports delays up to 30 ms, the video application supports delays up to 300 ms, while the delay for background applications can be up to 500 ms [10].

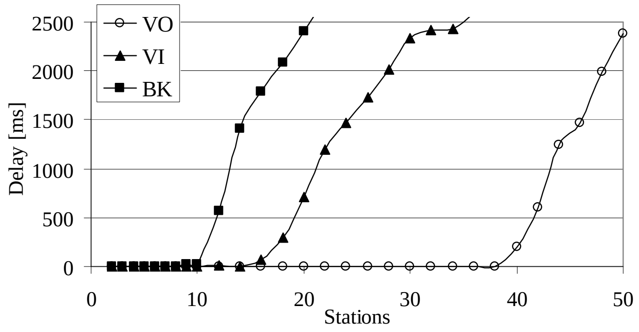

Figure 3. Delay for each application as a function of the number of station.

Hence, in the different experiences, our QAP supports up to 40 voice users, 18 video users, and 11 background users, with an appropriate degree of QoS.

Another performance measure is the maximum achieved goodput achievable for a given channel capacity. It is certain that a fraction of the channel capacity is used up in form of overhead, acknowledgments, retransmission, token delay, etc. Channel capacity is the maximum possible data rate, that is, the signalling rate on the physical channel. It is also known as the data rate or transmission rate, assumed to be variable, between 6 and 54 Mbit/s. Goodput is the amount of "user data" that is carried by the wireless network. The results for goodput as function of the number of station are presented in Fig. 4 for the same set of experiences. Its maximum value is 16 Mbit/s.

Due to the scarcity of wireless bandwidth, it is important to evaluate its relative usage, i.e., its average utilization. For this purpose, we have computed the average data rate for the client stations. Since the distribution of the users in the square is uniform, it is easy to compute the probability of a given transmission mode, and the average available data rate in the client stations. The highest value obtained for the utilization, the ratio between the goodput and the average data rate, is around 80%, for VI, while the lowest one occurs for VO traffic. This is justified because the packet size is much higher for VI than for VO.

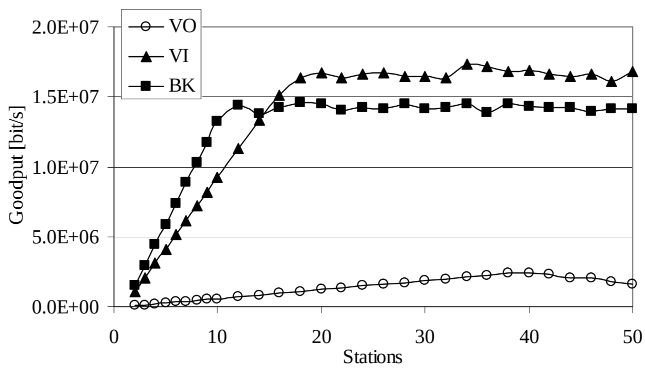

Figure 4. Goodput as a function of the number of node for VO, VI, and BK applications.

B. *Mixtures of applications*

The mixed scenario considers several applications being used simultaneously. The assumptions for the usage for each application are the following: 50% for VO, 30% for VI, and 20% for BK. Results were obtained by considering that the three applications are being used simultaneously, and the considered performance metrics were the same as before. They were obtained as an average over 20 simulations when the number of station is lower than 12, over 15 simulations when the number of station is equal or higher than 12 but lower than 22, and over 10 simulations for the remaining cases.

Fig. 5 presents the delay for each access category. For BK traffic the delay starts to increase considerably with more than 12 stations, while for VI traffic the delay starts to increase around 34 stations. In contrast with BK and VI, VO applications present a negligible delay. While delay can be tolerated for BK traffic, the results for VI justified the exploration of the Block ACK procedure through the application of several policies, also via simulation.

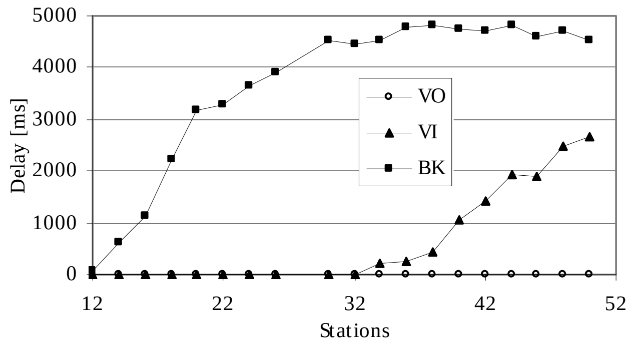

Figure 5. Delay with mixtures of applications.

Fig. 6 presents the number of collision as a function of the number of station. The number of collision is very low only for BK traffic. For VI and VO, the collisions occur even with lower number of station, although it does not degrade performance substantially.

Fig. 7 presents results for the goodput obtained for applications of each access category as well as the total one. When the number of station is low, the maximum goodput is obtained for VI traffic while the minimum goodput is obtained for VO traffic, this is because VO packets have smaller payload and an inter arrival time higher than the one for VI traffic, which has a higher payload and a smaller inter arrival time. When the number of station is high the maximum goodput still occurs for VI while its minimum value occurs for BK applications. This is because this access category has a very high CW which slows down the access of the traffic from this queue to the shared medium.

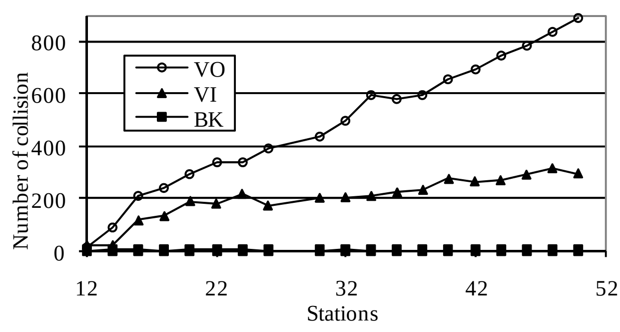

Figure 6. Number of collision with mixtures of applications.

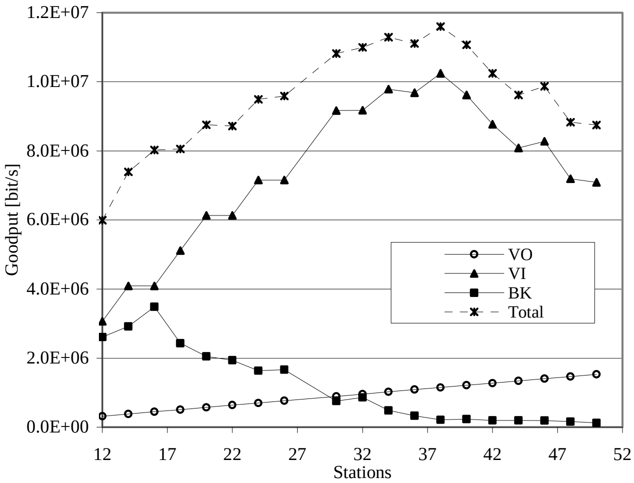

Figure 7. Goodput with mixtures of applications.

C. Block ACK for fixed buffer size

As previously mentioned, the Block ACK mechanism improves channel efficiency by aggregating several acknowledgments into one frame. In simulations with Block ACK, standalone services were considered, and the same scenario was assumed. The only difference is the use of the Block ACK procedure; a buffer size of 16 fragments was assumed in these experiences, and only BK and VI traffic were considered. For VO, it is certain that Block ACK is not a solution because, in order to fill up a buffer with 16 packets, the delay of the first one would be of 320 ms, too high for this application. Fig. 8 presents results for the delay as a function of the number of station. It starts to increase to values of the order of tens of millisecond for a number of station higher than 12 station for BK, and around 16 stations for VI.

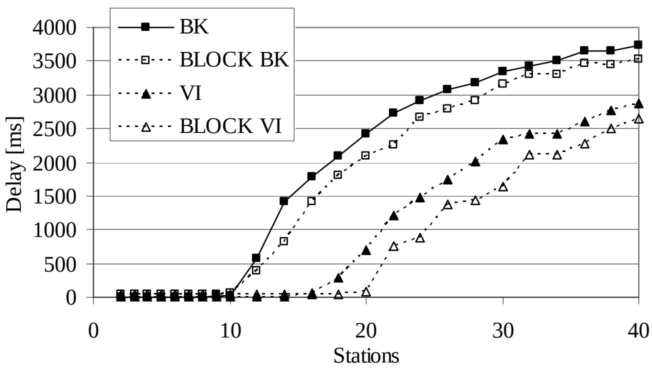

Figure 8. Delay for BK with and without Block ACK.

With Block ACK the delay is lower. The improvement (reduction) for BK traffic is 300ms in average, for more than 12 stations, while, for VI traffic, it is 420ms in average, for more than 16 stations.

Fig. 9 presents the goodput as a function of the number of station, also for BK and VI. With Block ACK, the respective improvement for BK and VI is of 2 Mbit/s and 2.2Mbit/s in average, after 12 stations.

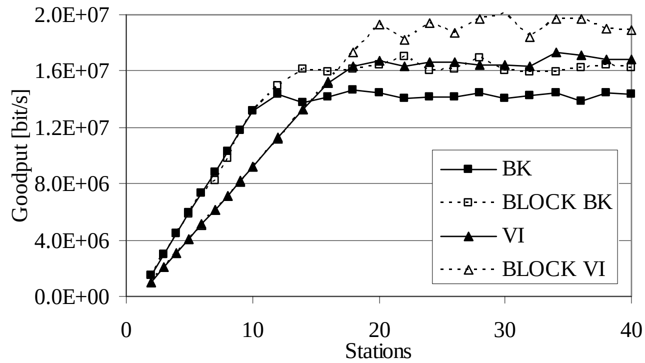

Figure 9. Goodput for BK with and without Block ACK.

D. Optimum buffer size discovery

After showing the advantage of Block ACK for a specific value of the buffer size, it is important to seek the optimum Block ACK policy, i.e., optimum block size. As immediate Block ACK is suitable for high-bandwidth low-latency traffic, after proving that the Block ACK procedure provides higher capacity and lower delays, we decided to explore which should be the policy regarding the buffer size, and relatively high values can be tolerated for the delay in BK traffic, we only considered the video service class. Simulations were performed for buffer sizes of 4, 6, 8, 10, 12, 14, 16, 18. Figs. 10 and 11 present the results for delay and goodput, respectively. These charts resulted from 70 simulations for each point, for an actual time (simulated) of 10 s. The 95% confidence intervals are only presented in Figs. 12-13 for buffer sizes of 4, 10 and 16. If we consider a maximum threshold of 300 ms for delay, by using the block ACK procedure, we can support two extra, corresponding to an increase of capacity of 2 Mbit/s, Fig. 13.

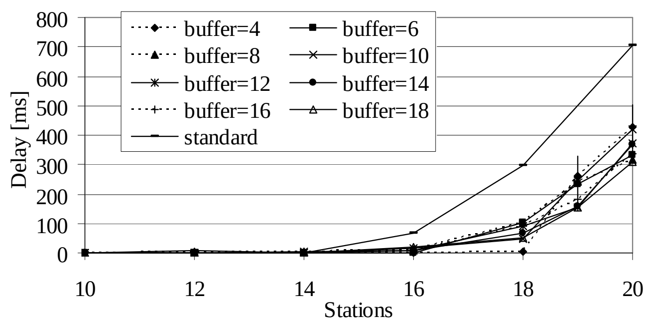

Figure 10. Delay for VI traffic with several buffer sizes for the Block ACK procedure.

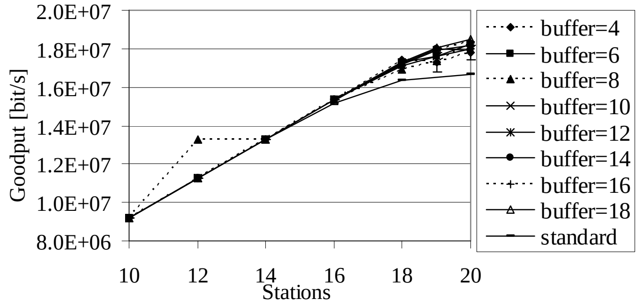

Figure 11. Goodput for VI traffic with several buffer sizes for the Block ACK procedure.

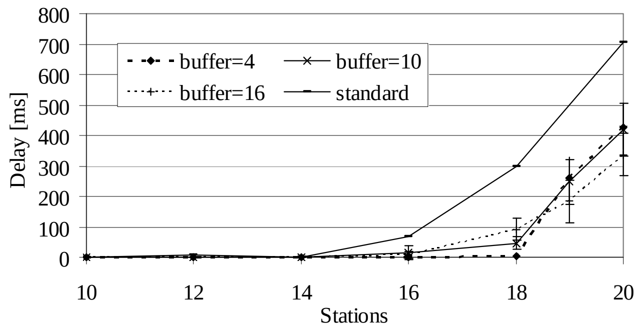

Figure 12. Delay for VI traffic with several buffer sizes for the Block ACK procedure for buffer sizes of 4, 10, and 16.

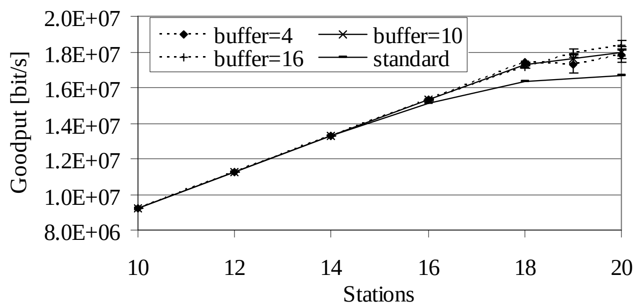

Figure 13. Goodput for VI traffic with several buffer sizes for the Block ACK procedure for buffer sizes of 4, 10, and 16.

Although a clear trend is not noticeable regarding the dependence on the block size, we believe that if the simulated time increases the trend will be noticeable.

VII. Conclusions and Future Work

In the simulations, with single services, higher values of goodput are found for the VI and BK applications, not only because the frames transmitted in these services are longer than the ones for VO but also due to the higher application data rate. As a consequence, the number of supported stations for VI and BK is lower than the number of supported VO stations. It should be noticed that the use of small CW for the VO access category may not be the best option since, when the number of station is higher than 40, the goodput starts to decrease (due to a small CW size). As a consequence of successive collisions the retransmission threshold is overcome causing the increase of packet losses. As the VO traffic is bi-directional, collisions occur very frequently because, apart of the access point, all the stations are contending to access the medium.

With multi-services, higher goodputs are also obtained for VI traffic. This is mainly due to the high payload of the frames, and the CW[VI]=2·CW[VO]. When the number of station is small, the goodput for VO applications is the lowest. However for an higher number of station the lowest goodput occurs for BK traffic. For VO applications the goodput is always increasing while for BK and VI traffic there is a value for a minimum value for the number of station after which the individual goodput of each station starts to decrease, explained by the small CW and AIFS for VO.

The Block ACK procedure provides an improvement in all the measured aspects. For VI application, the goodput reaches its maximum value of 20 Mbit/s and maintains an approximate constant value, beyond 16 stations, a value around 2.2 Mbit/s higher than without Block ACK. For BK traffic the increase is around 2 Mbit/s, beyond 12 stations. When the number of station is lower than 16, for VI, and lower than 12, for BK traffic, the goodput is higher without Block ACK. If the number of station is higher than 16, for VI, and 12, for BK traffic, the delay is lower with Block ACK procedure, Fig. 14. However, for the lowest values of the number of station, the use of Block ACK worsens system performance. From a detailed analysis of the VI application, a reduction of more than 40% was observed in the delay.

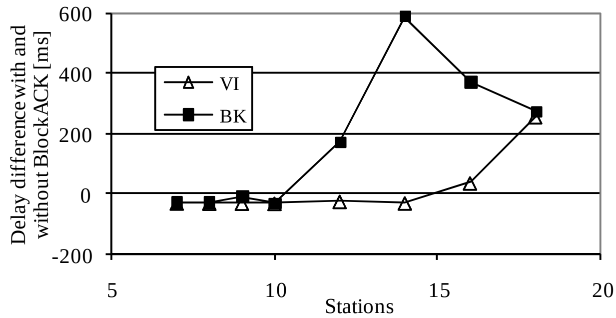

Figure 14. Difference for delay with and without Block ACK.

As a trend was not found for the optimum block/buffer size by simulating standalone services, future work will include exploring the results with mixtures of applications (with Block ACK) for different policies. Perhaps there is also a need for a different approach to account for delay since, although there is higher channel utilization by using Block ACK, if an application that uses all packets misses one it will have to stop, in order to wait for the missing packet.

References

[1] IEEE Std. 802.11e; Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications, 2005.
[2] D. T. Phan Huy, V. Monteiro, A. Gameiro, X. Yang, J. Rodriguez, R.Tafazolli, "System level performance evaluation of MATRICE air interface," in *Proc. of 13th IST Mobile & Wireless Communications Summit 2004*, Lyon, France , June 2004.
[3] IEEE Std. 802.11; Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
[4] Jijun Luo Mukerjee, R. Dillinger, M. Mohyeldin, E. Schulz, E., "Investigation of radio resource scheduling in WLANs coupled with 3G cellular network," *IEEE Communications Magazine*, Vol. 41, No. 6, pp. 108-115, June 2003.
[5] Orlando Cabral, Alberto Segarra, Fernando J. Velez, "Event-Driven Simulation for IEEE 802.11e Optimization," *IAENG International Journal of Computer Science (IJCS)*, Vol. 35 , No. 1, pp. 161 -173 , Feb. 2008.
[6] Quiang Ni, "Performance Analysis and enhancements for IEEE 802.11e Wireless Networks," *IEEE Network*, Vol. 19, No. 4, July/Aug. 2005, pp. 21-27.
[7] IEEE Std. 802.11a; Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band, 1999.
[8] D. Qijao, S. Choi, "Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation", in *Proc. of IEEE ICC'2001 - Int. Conf. on Communications*, Helsinki, Finland, June 2001.
[9] A Grilo, *Quality of Service in IP-based WLANs*, Doctoral thesis, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, June. 2004.