



HAL
open science

Dynamic frame aggregation scheduler for multimedia applications in IEEE 802.11n networks

Emna Charfi, Cédric Gueguen, Lamia Chaari, Bernard Cousin, Lotfi Kamoun

► **To cite this version:**

Emna Charfi, Cédric Gueguen, Lamia Chaari, Bernard Cousin, Lotfi Kamoun. Dynamic frame aggregation scheduler for multimedia applications in IEEE 802.11n networks. Transactions on emerging telecommunications technologies, 2015, 10.1002/ett.2942 . hal-01249802

HAL Id: hal-01249802

<https://hal.science/hal-01249802v1>

Submitted on 3 Jan 2016

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.

Dynamic Frame Aggregation Scheduler for Multimedia applications in IEEE 802.11n Networks

Emna CHARFI¹, Cédric GUEGUEN², Lamia CHAARI¹, Bernad COUSIN², Lotfi KAMOUN¹

¹Laboratory of Electronics and Information Technologies (LETI), University of Sfax, National School of Engineering, B.P. 3038 Sfax, Tunisia

²University of Rennes 1 - IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

Charfi.emna@gmail.com, cedric.gueguen@irisa.fr, lamia.chaari@tunet.tn, bernard.cousin@irisa.fr,
lotfi.kamoun@isecs.rnu.tn

Abstract

Providing Quality of Service (QoS) to real time applications over Wireless Local Area Networks (WLANs) is becoming a very challenging task due to the diversity of multimedia applications. Concurrently, there are numerous WLANs devices that are rising recently. Mainly, we focus on IEEE 802.11n since it was designed to support a high data transmission rate (toward 600 Mbps) based on frame aggregation schemes. The aggregation mechanism accumulates many frames before transmitting them into a single larger frame, thus reducing overhead and increasing efficiency and throughput. Yet, this scheme cannot provide QoS satisfaction for delay sensitive application even if it supports higher throughputs. Indeed, aggregation headers cause supplementary delays particularly when aggregating infrequent packets with small sizes. To overcome this limitation, we propose in this paper a new Dynamic Frame Aggregation (DFA) scheduler to provide QoS satisfaction to real time services. To achieve this goal, we defined new scheduling parameters such as QoS delays to avoid accumulation of non-scheduled packets. Hence, the DFA scheduler serves packets and dynamically adjusts the aggregated frame size based on these QoS delays. Conducted simulations illustrate the performance of our proposed DFA scheduler in term of satisfying QoS, throughput, loss and delay requirements of voice and video traffics.

Keywords: IEEE 802.11n, frame scheduler, urgency delay, waiting delay, optimized frame aggregation.

1 Introduction

Nowadays, Wireless Local Area Network (WLAN) has experienced tremendous growth with the proliferation of IEEE 802.11 standards [1][2][3][4][5]. Basically, recent WLAN standardizations aim to satisfy Quality of Service (QoS) requirements for real time applications such as audio and video flows that have strict requirements in term of rate, delay, and loss. In fact, the original IEEE 802.11 standard doesn't provide QoS satisfaction for real time services. All recent WLANs standards are derived from the original IEEE 802.11 standard [1] that was mainly designed for data applications without considering traffic differentia-

tion as well as QoS requirements. From this fact, the 802.11e [3] amendment introduces service differentiation scheme supported by a new access mechanism, called Enhanced Distributed Channel Access (EDCA) [6]. Among this scheme, application traffics are classified into prioritized Access Categories (ACs) where high priority traffic has higher chance to be transmitted than lower priority traffic.

However, EDCA mechanism was not able to guarantee QoS for applications having strict QoS requirements such as real time services [7]. Concurrently, there exists a large traffic diversity from multimedia applications such as voice, video telephony, video conferencing and high-definition television (HDTV) that have to be transmitted with high data rates.

With this aim in mind, IEEE 802.11n [4] provides high throughput at the MAC layer achieving up to 600 Mbps [8][9]. This high throughput has been achieved via many enhancements at both Physical and MAC layers. In the physical layer, 802.11n utilizes a MIMO technology where multiple antenna elements can be combined to achieve either higher PHY data rates (in Spatial Division Multiplexing (SDM) mode) or higher range (in Space Time Block Coding (STBC) mode). Moreover, 802.11n uses channel bonding, where two 20 MHz channels of legacy 802.11 can be combined to a single 40MHz channel, thus increasing the PHY data rate. A key MAC enhancement is the frame aggregation mechanism [10] which increases the payload size by transmitting multiple frames into a single frame. Indeed, IEEE 802.11n defines two types of aggregation mechanism: Aggregated MAC Service Data Unit (A-MSDU) and Aggregated MAC Protocol Data Unit (A-MPDU). The principle of A-MSDU is to allow multiple MAC Service Data Units (MSDUs) from the same sources to be sent to the same receiver concatenated in a single MPDU. The principle of A-MPDU is to join multiple MPDUs to be sent to the same receiver with a single PHY header. By this concept, new aggregation headers are introduced and become parts of the transmitted frame.

In one hand, the presence of such headers has a negative effect in the case of delay sensitive multimedia applications, particularly when aggregating frames of small payloads [11]. Furthermore, when the frame queue is empty, the MAC layer has to wait for frames to fill the A-MPDU inducing an additional delay. Thus, the aggregation mechanism can badly affect the delay of multimedia applications especially in a differentiated service network [12].

On the other hand, additional delay can occur when packets are poorly scheduled. For example, with Priority Queuing (PQ) scheduler [13][14], differentiated traffic are scheduled based on their Delay Target (DT), which is the expected time elapsed between the moments once a source sends a packet, to the moment it reaches its destination. Nonetheless, there are accumulations of non-scheduled packets which are waiting in the queue for a long duration. Hence, an efficient scheduling mechanism is a key to provide the QoS required by real-time services. In this context, numerous proposals of frame scheduling mechanisms exist in the literature, but there are all restricted to IEEE 802.11e network with EDCA or HCCA mechanisms [15][16][17]. However, scheduling real time applications in the context of a high throughput 802.11n WLAN has received little prior attention [18].

In this paper, we focused on the effect of frame aggregation on the support of voice and video applications as well as the impact of using PQ scheduler on delay, throughput or loss rate performance. To achieve this aim, we proposed a new dynamic frame aggregation (DFA) scheduler that considers QoS requirements of real time applications. Our dynamic scheduler takes decision based on the Urgency Delay (UD) as well as the Waiting Delay (WD) of each packet, contrarily to PQ scheduler which is based only on DT. We dynamically adjust the frame aggregation size based on the delay requirements of a scheduled packet. The main objectives of this scheduler are reducing the transmission delay while maintaining high throughput and low loss rate in 802.11n network based on the aggregation mechanism with prioritized application traffics such as voice, video, and streaming.

The rest of the paper is organized as follows. Section 2 briefly provides background information on IEEE 802.11n aggregation mechanisms. We describe the main motivations of this work in Section 3. In Section 4, we discuss some of the relevant research works in the field. We present the design of our proposed DFA scheduler in Section 5. The algorithm description is provided in Section 6. In Section 7, we focus on the others schedulers that have been discussed in Section 4 to evaluate the performances of our scheduler. Performance analysis and results discussion is given in Section 8. We finally draw a conclusion of our work in Section 9.

2 IEEE 802.11n aggregation scheme

Since the main limitation of the legacy 802.11 MAC layer is the overhead produced by MAC header, using larger frames is one solution to reduce the overhead caused by MAC header. In this context, IEEE 802.11n proposed an aggregation scheme where a number of frames are transmitted together into aggregated frames. In fact, the aggregation scheme reduces the time of overhead transmission, and reduces the waiting time caused by random backoff period during successive frame transmissions. IEEE 802.11n adopts two approaches for the aggregation data. The first one is the Aggregated MAC Service Data Unit (A-MSDU), and the second is Aggregated MAC Protocol Data Unit (A-MPDU).

2.1 A-MSDU aggregation

Fundamentally, A-MSDU is designed to tolerate multiple MSDUs having the same source to be transmitted to the same receiver concatenated in a one MPDU. The top MAC layer receives frames from the Link Layer and these buffered frames are then aggregated to form a single A-MSDU. For each MSDU subframe in an A-MSDU frame, the MSDU subframe includes the Subframe Header, the MSDU data payload and the Padding field, as it is shown by Figure 1. The Subframe Header includes three fields: the Destination Address (DA), the Source Address (SA) and Length (L) which indicates the size of MSDU data payload. The A-MSDU aggregation is only tolerable for frames having the same source and destination. The maximum length A-MSDU that a station can receive is either 3839 bytes or 7935 bytes. A single A-MSDU contains multiple MSDU subframes. A single A-MSDU frame forms a PSDU (Physical Service Data Unit) frame after the addition of a MAC header and an FCS field.

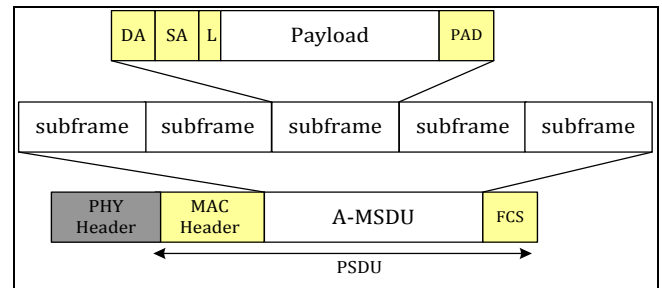


Figure 1. A-MSDU aggregation

2.2 A-MPDU aggregation

The principle of A-MPDU is to send multiple MPDU subframes, intended to be sent to the same destination, with a unique PHY header in the goal to reduce the overhead PHY header. For each A-MPDU, every MPDU subframe includes an MPDU frame, the MPDU delimiter and the padding bytes. Multiple MPDU subframes are concatenated into one larger A-MPDU frame. All the MPDU subframes within an A-MPDU should be addressed to the same receiver, but the MPDU subframe could have different source addresses. With A-MPDU, is fully formed MAC PDUs are logically aggregated at the bottom of the MAC layer. A short MPDU delimiter is associated to each MPDU. Multiple MPDU subframes are concatenated into one larger A-MPDU frame. The MPDU delimiter is 32 bits in length and consists of a 4-bit rescheduled field, a 12-bit MPDU length field (L), an 8-bit CRC field, and an 8-bit signature field. A station advertises the maximum A-MPDU length. The advertised maximum length may be one of the following: 8191, 16383, 32767, or 65 535 bytes. Figure 2 represents the two level of aggregation scheme A-MSDU and A-MPDU.

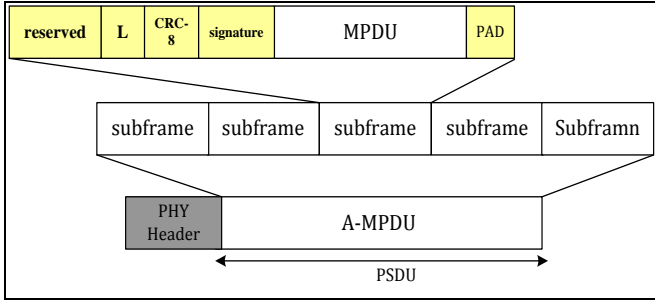


Figure 2. A-MPDU aggregation

2.3. Two-level aggregation

A two-level frame aggregation consists of a mix of A-MSDU and AMPDU over two stages as it shown in Figure 3. The basic process is explained as follows: In the first stage, several MSDUs are accumulated to form an A-MSDU frame based on A-MSDU aggregation concept explained above. In the second stage, the A-MPDU concept is involved to construct an A-MPDU frame based on accumulating several MPDU subframes. Note that each MPDU includes in its turn numerous A-MSDUs subframe which have the same destination. This concept is not mandatory for IEEE 802.11n.

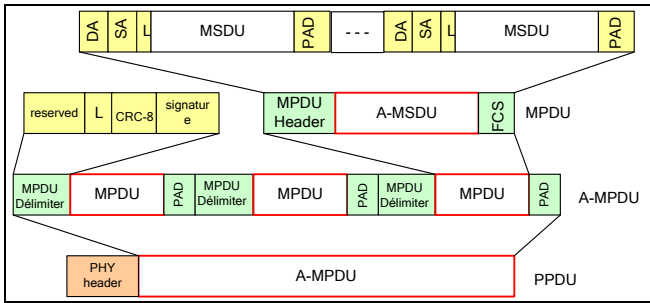


Figure 3. Two-level Aggregation

3 Motivations

3.1. QoS Mechanisms

In the previous section, we have illustrated the amazing growth in the area of wireless networking technologies with the aim of providing more performance for intensive multimedia applications. To provide QoS satisfaction, network designers must integrate QoS mechanisms to allow differentiation between all traffic types that have strict requirements in terms of delay, throughput, and loss. Basically, there are three major QoS mechanisms which are: Classification, admission control and scheduling. First of all, packet classification is used to categorize packets into flows and distinguish between different kinds of traffic. Once packets are classified, an admission control mechanism is performed to take a decision of accepting or not the incoming flow frame according to its QoS requirements and the available network resources. Afterwards, packets are scheduled and queued into memory buffers. There are numerous schedulers that exist among the literature such as

First In First Out (FIFO), Priority Queuing (PQ), Fair Queuing (FQ), Round Robin (RR), and Random Access (RA), Weighted Fair Queuing (WFQ), and Weighted Round Robin (WRR) [19] in addition to other well known opportunistic scheduling algorithms such as Proportional Fair (PF), Maximum Signal to Noise Ratio (MaxSNR), and Weighted Fair Opportunistic (WFO) [20].

Among all existing schedulers, we will focus on PQ scheduler. Our choice was based on the two following reasons:

i) PQ serves packets according to their queuing classification which is more appropriate for the aggregation concept in which packets has to wait each other to form an aggregated frame.

ii) Opportunistic schedulers allocate radio resources taking into account the radio conditions which should be unchanged to the instant of transmitting the frame. However, these ratio conditions may be modified when transmitting an aggregated frame since a long waiting period may be required to fill the aggregating frame. Hence, using opportunistic schedulers is not pertinent for IEEE 802.11n aggregation scheme.

3.2. PQ scheduler

Basically, IEEE 802.11e was designed to provide QoS differentiation among different users and it classifies queues into 4 Access Categories (ACs): voice (AC_VO), video (AC_VI), background (AC_BK), and best effort (AC_BE). The PQ scheduler uses multiples queues with different levels of priority, and packets are placed in one of the queues along with their classification. Queues with higher priority are the first scheduled with considering the FIFO schedule in each queue. The Delay Target (DT) associated with the flow of a received packet determines the packet waiting queue and thus the packet priority. DT is the difference between the moment when a source wants to send a packet and the moment when the packet should reach its destination. DT is an important metric which affects users' satisfaction since real-time applications have strict requirements in term of delay. Based on this delay, the PQ scheduler takes decision to serve packets in the highest priority queue thus having the lower DT. In consequence, with this scheme, voice packets are always the first scheduled since they have the lower DT, video and streaming packets will be accumulated causing an increased waiting delay for oldest packets when there exists an overabundance of voice packets. Figure 4 presents an example of the behavior of a PQ scheduler where different AC queues. In this example, there are three flow types: voice, video, and streaming. Each flow generates packets from instant t_0 to t_4 . Without loss of generality, we assume that all packets have the same size. Packets (vo_t_j) , (vi_t_j) and (str_t_j) are respectively the generated voice, video, and streaming packets at the instant t_j where $j=0, 1, 2, 3, 4$.

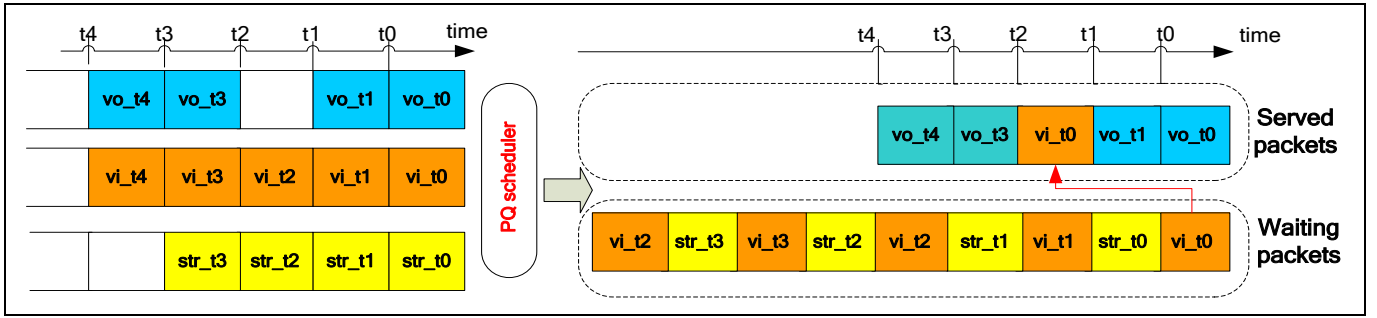


Figure 4. PQ scheduler concept (example)

At $t = t_0$, there is one packet in each priority queue, (vo_t_0) will be scheduled while (vi_t_0) and (str_t_0) will be stored in the waiting queue with this order since voice packets have the lower DT. We assume that the sending delay of one packet is elapsed between (t_i) and (t_{i+1}) . At $t = t_1$, audio packet (vo_t_1) is scheduled, video and streaming packets (vi_t_1) and (str_t_1) are stored in the waiting queue. At $t = t_2$, there is no audio packet available, hence (vi_t_0) packets will be scheduled since it is the oldest packets in the video queue. With this principle, packets continue to be scheduled at $t = t_3$ and $t = t_4$. As a consequence, voice packets are always scheduled while video packets (and even more so with streaming packets) are stored in the waiting queue producing an important waiting delay.

3.3. Aggregation Delays

The aggregation scheme is based on aggregating packets from different applications to compose a larger packet which will be sent to same destination. Although this mechanism improves the average throughput, it has a negative effect on the delay performance of some applications. As a matter of fact, varying the aggregation size has an impact on the packet delay. This is due to two reasons. In one part, in the case of aggregation of packets with small sizes, many headers are added causing additional delays. In another part, when the size of aggregated frame is raised, the delay is considerably increased due to the time added when waiting for other packets in the queue to construct the A-MPDU frame. In fact, for low rate applications such as VoIP, the required delay to fill the aggregated frame will be increased so affecting the QoS. Hence, aggregation scheme badly affects delay since waiting for further packets increases highly the delay of earliest packets. Consequently, we can conclude that the use of aggregation for low rate applications degrade the end-to-end delay. Fig.5 draws the advantages and the drawbacks of the aggregation mechanism. All these factors justify the inefficiency of the aggregation mechanism in the case of delay sensitive applications especially voice and video services. Hence, an optimization of the aggregated frame size is needed to outperform the increased delay.

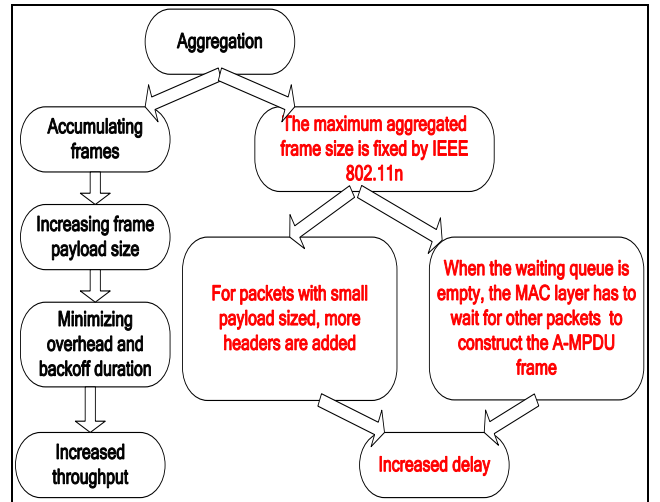


Figure.5. Advantages and drawbacks of aggregation scheme

3.4. PQ scheduler: problem specification

In this section, we study the performance of PQ scheduler on satisfying QoS requirements of voice, video, and streaming applications, while supporting the 802.11n frame aggregation scheme. We consider the flow throughput, the delay of serving a packet, and the number of dropped packets as performance metrics. In fact, a better scheduler is the one that serves packets with reduced delays and engenders lowest number of rejected packets. Moreover, to be realistic, we consider two network states: low saturated and saturated networks. In a saturated network, all nodes have always packets to transmit and keep contending to the channel. Thus, the priorities queues are always occupied. In a low saturated network, nodes keep always contending to the medium by transmitting smallest amount of data among the time. Fig.6 depicts the serving delay under the above two network stated, and Fig.7 portrays the percent of dropped packets among the time. Considering these results, it can be observed that PQ scheduler badly treats packets when the network is saturated compared to a low saturated state. In fact, under the first condition, the serving delay of these packets will be increased and the number of dropped packets will be important. Lower delays and lower dropped frame rate are obtained under a low saturated network. Indeed, in the last condition, the aggregated frames can contain almost all the received packets since few packets are present. That's why the serving delay as well as the

number of dropped packets will be reduced. As a consequence, PQ scheduler presents a high inefficiency when the network is saturated due to the high number of dropped packets and the increased delay to serve packets.

For that reasons, we propose a new scheduling scheme called DFA scheduler in order to overcome the drawbacks of PQ scheduler under the saturated condition. Hence, the main goals of our proposal are to reduce the serving delay as well as the number of rejected packets to satisfy the QoS requirements of real time applications.

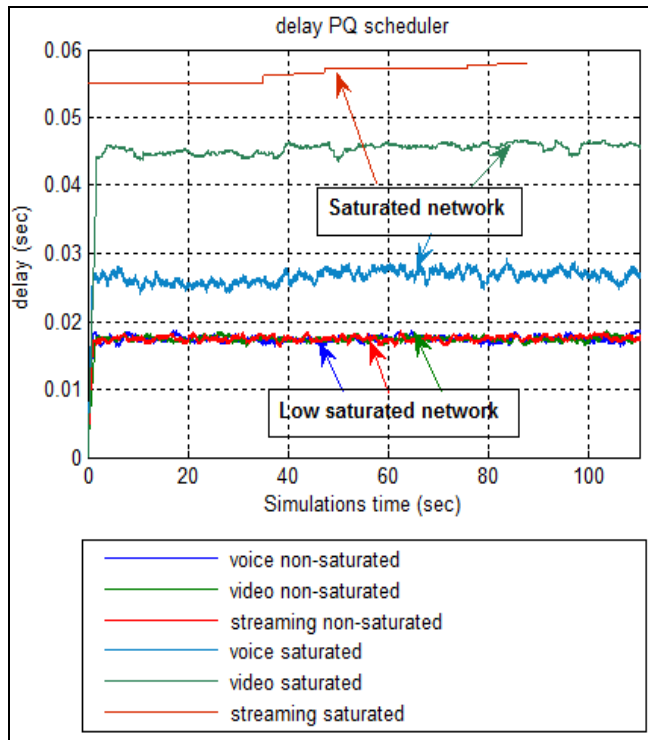


Figure.6. Delay for PQ scheduler

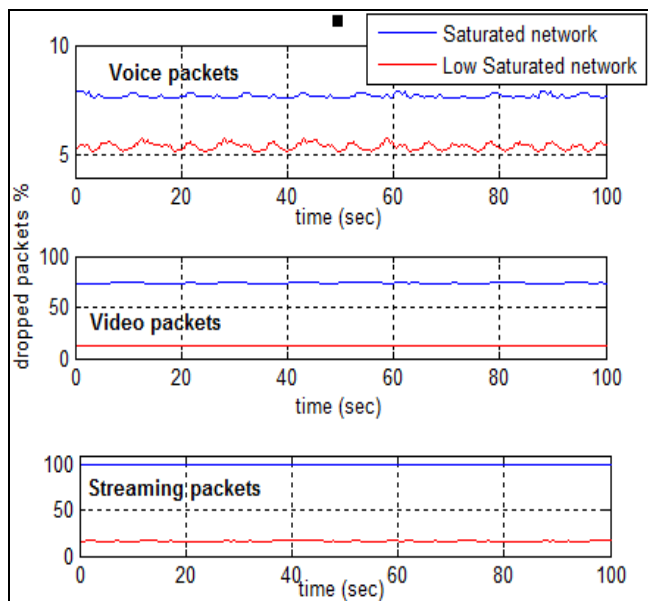


Figure.7. Dropped packets for PQ scheduler

4 Relayed works

In the research literature, there are numerous studies which focus on evaluating the performance of the aggregation mechanism for guarantying QoS fairness for real-time flows in IEEE 802.11n WLAN. Similarly, there are some others works that focused on proposing QoS mechanisms such as scheduling with the aim to provide QoS satisfaction for real-time applications when using the aggregation mechanism. In this section, we study specific features of different proposed schedulers that operate according to this aim.

Authors in [21] handle the weakness of 802.11 aggregation schemes by providing a detailed analysis of packet delay. They considered the packet delay as the amount of time separating the instant of generating this packet and the instant of successfully receiving it. They proved that larger frames increases delays, while transmitting smaller frames is more appropriate for real-time applications since it reduces delays.

Similarly, authors in [22] focused on the negative impact of aggregation on some application kinds. In fact, the aggregation mechanism causes larger delays and header overhead especially when aggregating frames of small payloads. To overcome this limitation, authors proposed a new aggregation scheme called mA-MSDU to reduce the A-MSDU and A-MPDU header overhead and to support applications with small frame size such as VoIP. With mA-MSDU aggregation scheme, small headers are introduced and error control is enabled over the aggregated MSDUs. Authors proved by simulation the performance of this scheme mainly for packets with small sizes.

To satisfy QoS requirements of some application kinds, authors in [23] developed an enhanced MAC layer that supports both IEEE 802.11n aggregation scheme and IEEE 802.11e EDCA service differentiation mechanism. Using this modified MAC layer improves QoS metrics such as delay and throughput of these kinds of applications. Furthermore, authors proved that when enforcing strict priorities collisions are reduced to zero and the waiting time can be minimized for typical usage scenarios.

Similarly, authors in [24] presented a design of a high throughput MAC supporting QoS requirements which combine the 802.11e Hybrid Coordination Function HCF with the 802.11n frame aggregation scheme to provide QoS satisfaction. The proposed design includes some QoS mechanisms such as: admission control, calculating allocated the Transmission Opportunity Period (TXOP), and a scheduler. They showed by simulation that the proposed new MAC protocol is efficient since it improves capacity for real time traffic, and enhances channel utilization, and reduces packet delay for best-effort traffic.

Authors in [25] are interested also on delay inefficiencies of IEEE 802.1n aggregation scheme and proposed a scheduling algorithm in order to overcome this limitation. Firstly, they focused on benefits and drawbacks of both A-MSDU and A-MPDU in very high throughput network. Then, they suggested a frame aggregation scheduler that dynamically chooses the length of an aggregated frame and the used aggregation technique. This scheduler is based on estimat-

ing the optimal time deadline of each frame as well as estimating the best aggregation scheme to use. The proposed scheduler outperforms the legacy A-MPDU performance and avoids the tradeoff between throughput and delay.

In the same way, authors in [26] proposed a dynamic scheduler to adjust to frame aggregation size with the aim to outperform the limitation of this scheme especially in term of delay. This scheduler considers the specific QoS requirements of multimedia applications, and adjusts the aggregated frame size based on frame access category. Within this scheduler, packets which are insensitive to delay such as Background and Best effort ACs, are forced to wait for other packets.

Furthermore, authors in [27] are interested on the negative effect of the IEEE 802.11n aggregation mechanism in term of delay constrained multimedia applications in a WLAN. To increase the network efficiency, authors proposed three methods based on frames aggregation and cooperation among nodes. These methods use capture effect, power control scheme and directionality of antenna to authorize the concurrent transmission of several frames in a WLAN. The proposed schemes increase the system throughput greatly, in addition to improving VoIP performance in term of delay.

Another issue was addressed in [28] where authors investigated the case of an erroneous transmission of an aggregated frame. They proposed a new concept called Aggregation with Fragment Retransmission AFR to retransmit only the corrupted fragments of the aggregating frame in this case. Further, they developed an analytic model to evaluate the performance of such concept in term of delay and throughput and to estimate the optimal fragment and frame sizes.

In this work, we addressed the same issue as these previous works since our main goal is providing QoS satisfaction for real time applications among 802.11n aggregation scheme. For this purpose, we proposed a dynamic frame aggregation DFA scheduler that includes both scheduling packets and optimizing the frame aggregation size. The originality of our scheduler resides on considering new parameters such as Urgency Delay (UD) and Waiting Delay (WD) when it decides to serve packets. The principle of our proposal such as its performance analysis will be discussed in the next sections.

5 Dynamic Frame Aggregation scheduler design

As it is explained in the above section, the aggregation scheme is not an appropriate mechanism for real time applications since it has a negative impact on delay even if it guarantees higher throughputs. Thus, there is a tradeoff between throughput and delay. The increased delay is caused by the WD as well as by the delay elapsed to form an aggregated frame. Therefore, an efficient scheduler that serves packets based on their urgency delay and optimizes the aggregated frame size is required. For that reason, we propose a DFA scheduler that is implemented within the access point AP with the aim to reduce delay while maintaining higher throughputs. The scheduler parameters are listed in Table.1.

Table.1. DFA scheduler parameters

	PQ scheduler	Proposed DFA scheduler
Parameters	DT (Delay Target)	UD (Urgency Delay) WD (Waiting Delay)
Aggregated frame size	Fixed by IEEE 802.11n	Optimized according to UD

The operation of the proposed DFA scheduler can be divided into three main tasks as it is shown in Figure.8. The first task is packet generation. The second task is selecting the packet to be scheduled. The third task is dynamically adjusting the aggregated frame size.

➤ Task 1: Packet generation with random distribution

We consider in our work that there are three types of data sources: voice, video, and streaming. After packet generation, each frame will be mapped into ACs based on the EDCA scheme and will be stored into one of the three priority queues. In our traffic model (which takes into account a worse case), these packets are generated with a random distribution. Mainly, voice and streaming packets are generated with a uniform random distribution while video packets are generated with an exponential random distribution.

➤ Task 2: Selecting prioritized packets to be scheduled

This task can be divided into two steps. In the first step, at instant "t" an Urgency delay $UD(p, t)$ is associated to each packet "p". $UD(p, t)$ refers to the left time to serve a packet, it is equal to (1):

$$UD(p, t) = DT(p) - WD(p, t) \quad (1)$$

Where $DT(p)$ is the difference between the moment when a source wants to send a packet "p" and the moment when the packet "p" should reaches its destination. In other words, it is the delay target by the application. It is the maximum delay that a packet can support. After DT, an unreceived frame is not useful any more at the receiving side, thus can be dropped. For each $(t < T_0(p))$, we have $UD(p, t) = DT(p)$. However, $UD(p, t)$ is reduced by $WD(p, t)$ when this packet is not scheduled. Where $WD(p, t)$ refers to the waiting time of packet "p" in the scheduler at time t. This is the difference between t and the arrival time of the packet in the scheduler ($T_0(p)$) when he has not been sent, as it is given by (2). Otherwise, it refers to the difference between the arrival time of the packet in the scheduler and the time off.

$$WD(p, t) = t - T_0(p) \quad (2)$$

In the second step of this task, the scheduler selects prioritized packet to be scheduled. Indeed, the packet having the lowest UD will be the first scheduled. Hence, the proposed scheduler gives the priority to packets according to their $UD(p, t)$.

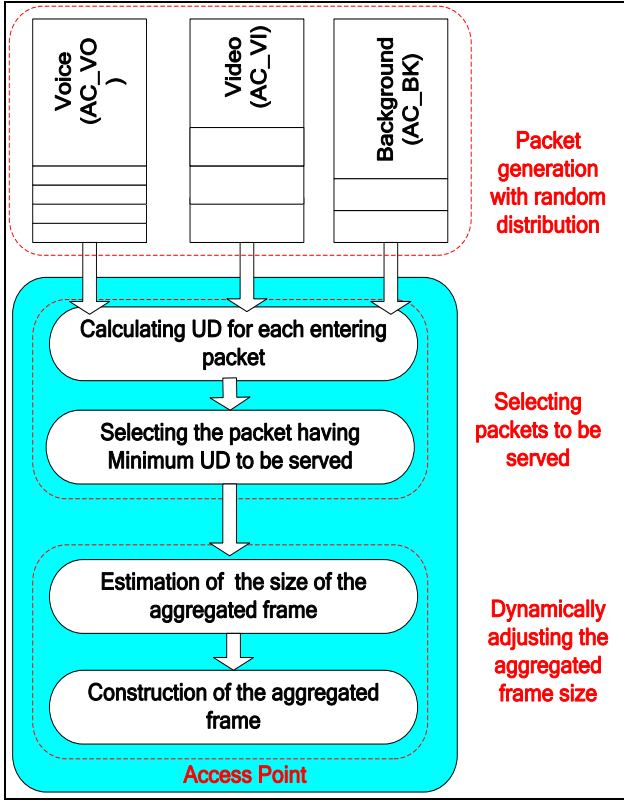


Figure.8. DFA scheduler design

➤ *Task 3: Dynamically adjusting the aggregated frame size*

Once one packet has been selected to be scheduled, the scheduler dynamically adjusts the size of the aggregated frame considering the UD of selected packets. In fact, the transmission delay of the aggregated frame should be lower than or equal to UD of the first selected packet. Based on this condition, the scheduler predicts the payload size of the aggregated frame.

6 DFA scheduler algorithm

In this section, we analyze the different steps involved in the proposed DFA scheduler. The functional bloc diagram is shown in Fig.10. As it is explained in the above section, in our traffic model there are three sources types that generate packets randomly among the time, and they will be classified into three queues. At an instant $t = t_i$, the DFA scheduler selects the appropriate packet to be scheduled. For that purpose, it calculates the $UD(p,t)$ of all non-scheduled packets based on expression (1). We can differentiate two types of non-scheduled packets:

- i. The non-scheduled packets which have already been scheduled by a previous intervention of the scheduler, but they were not scheduled.
- ii. The non-scheduled packets which has been generated after the previous intervention of the scheduler.

Note that packets which have a negative or a null $UD \leq 0$ will be dropped since they are not useful any more. After associating an UD to each packet, DFA scheduler selects the packet that has the lowest UD to be the first scheduled. Then, the scheduler decides to construct an aggregated

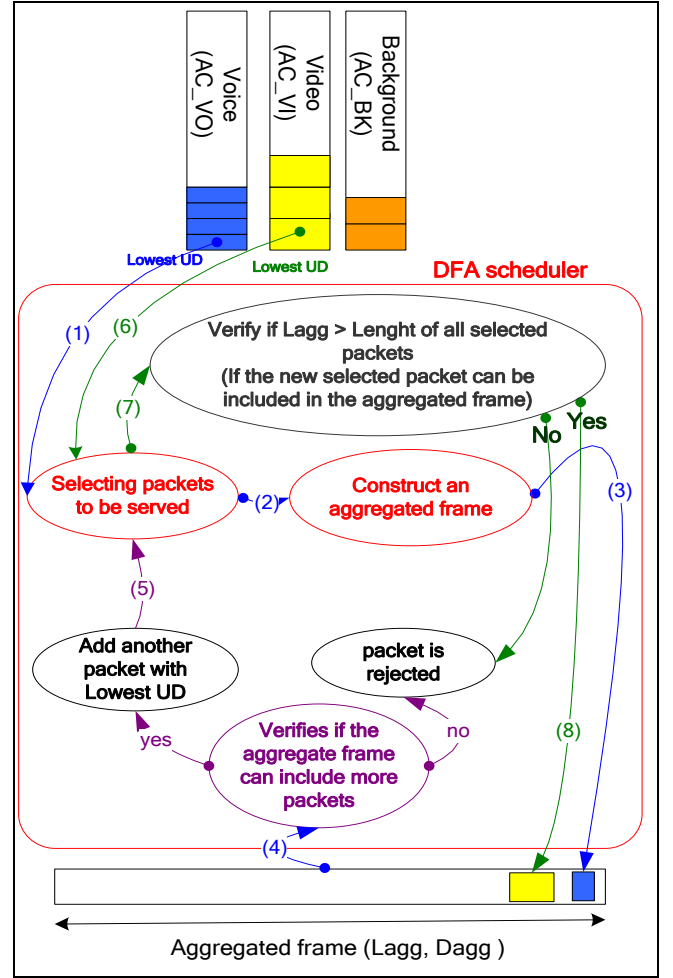


Figure.9. DFA functional description

frame which should be transmitted before D_{agg} . D_{agg} is the expected transmission delay of the aggregated frame. It must be lesser than, may be equal to, UD of the first selected packet. In fact, an aggregated frame contains numerous packets that have been selected according to their UD . Since the first selected packet had to be transmitted before UD , the transmission delay of the aggregated frame should not go above this value. From this fact, the scheduler calculates the expected aggregated frame size L_{agg} , based on (2):

$$L_{agg} = lowestUD \times PHY_{rate} \quad (2)$$

Where PHY_{rate} refers to IEEE 802.11n physical rate. Once L_{agg} is fixed, the DFA scheduler checks if L_{agg} is greater or lesser than the size of the selected packet $L_{selected_packet}$, in other words, it verifies if the current constructed aggregated frame has sufficient place to include another packet. Based on A-MPDU aggregation scheme, $L_{selected_packet}$ includes some overheads such as MPDU Delimiter (MD), MAC headers (MAC_{hdr}), FCS, and Padding fields (PAD) which are all added to the payload of the each selected packet. Hence, $L_{selected_packet}$ is expressed as (3):

$$L_{selected_packet} = MD + MAC_{hdr} + L_{payload_packet} + FCS + PAD \quad (3)$$

Therefore, if $(L_{agg} \leq L_{selected_packet})$, the DFA scheduler decides to add others packets to form the expected aggregated frame, otherwise it maintains only L_{agg} . Mainly, the DFA scheduler selects another packet that has the lowest UD among the remaining non-scheduled packets and verifies if L_{agg} is reached or not. By this way, the scheduler continues to select packets which have lowest UD until achieving L_{agg} . Afterward the scheduler enters in a sleep mode for a D_{agg} period, and then it resumes these steps.

We give in Fig.9 a functional description of the proposed DFA scheduler. In this example, the scheduler firstly selects the packet which has the lowest UD (step1), and then this selected packet is used to construct the aggregated frame (steps 2 and 3). After that, the DFA scheduler verifies if the expected aggregated frame can include other packets (step 4). If agreed, it decides to add another packet from the existing non-scheduled which has the lowest UD (step 5), otherwise remaining packets will be rejected. If the

aggregated frame can contain another packet, the scheduler continues to serve packets as previously (step6). Then, the scheduler verifies if the new selected packet can be encapsulated in the constructed aggregated frame (step 7). In steps 6, 7, and 8, the scheduler serves another packet that will be encapsulated in the aggregated frame. After each added new packet to the formed aggregated frame, step 4 is re-involved.

7 DFA scheduler performance evaluation

Fundamentally, the DFA scheduler is based on two main steps that are: selecting packets to be scheduled, and adjusting the aggregated frame size as it is illustrated in Figure 8. From this fact, we evaluate the performance of our proposal in comparison with three other schedulers which are given by Fig.11. Principally, we intend to compare the performance of the DFA scheduler with and without using the above two steps. We will compare our scheduler with the three following algorithms:

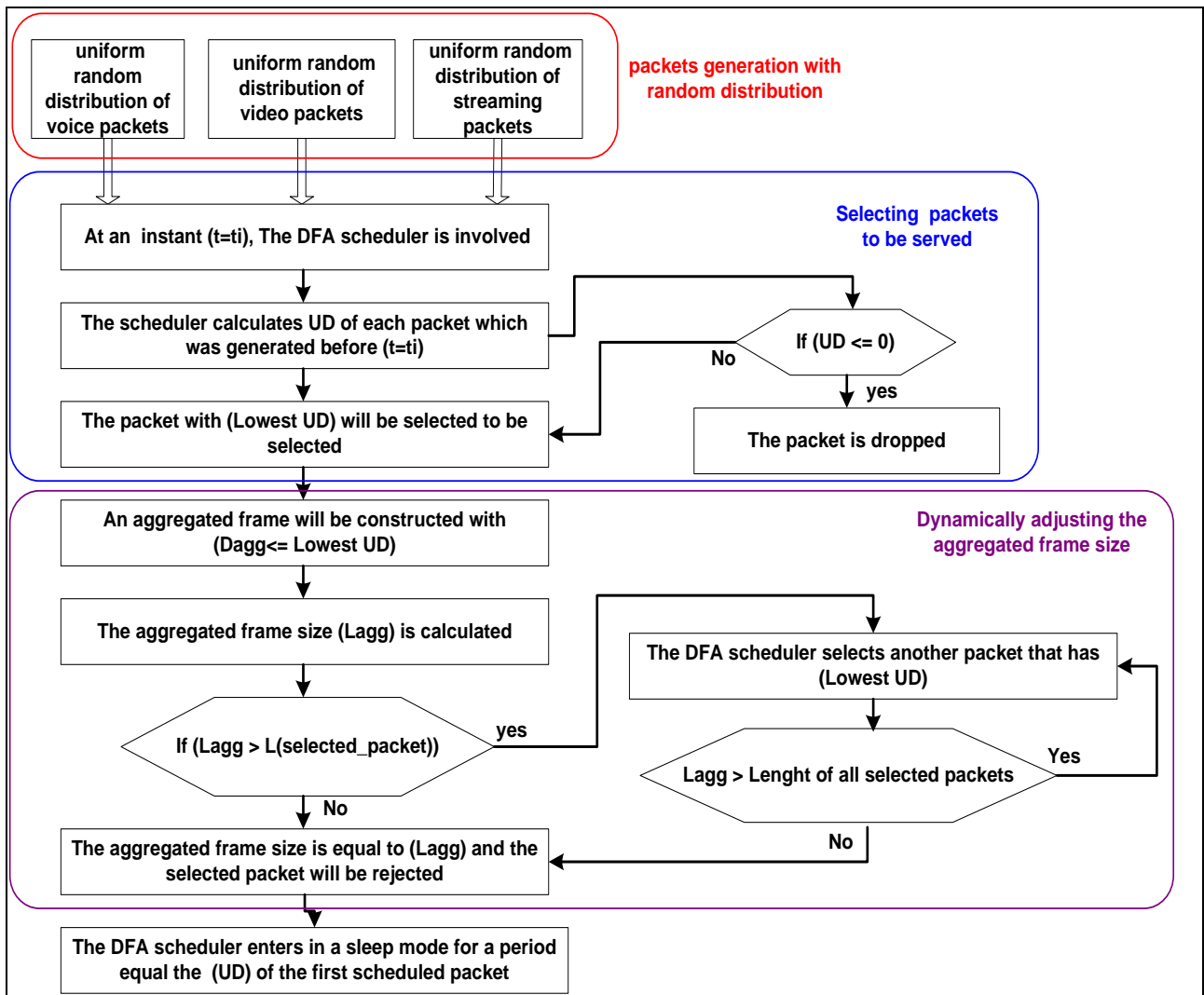


Figure.10. DFA functional diagram

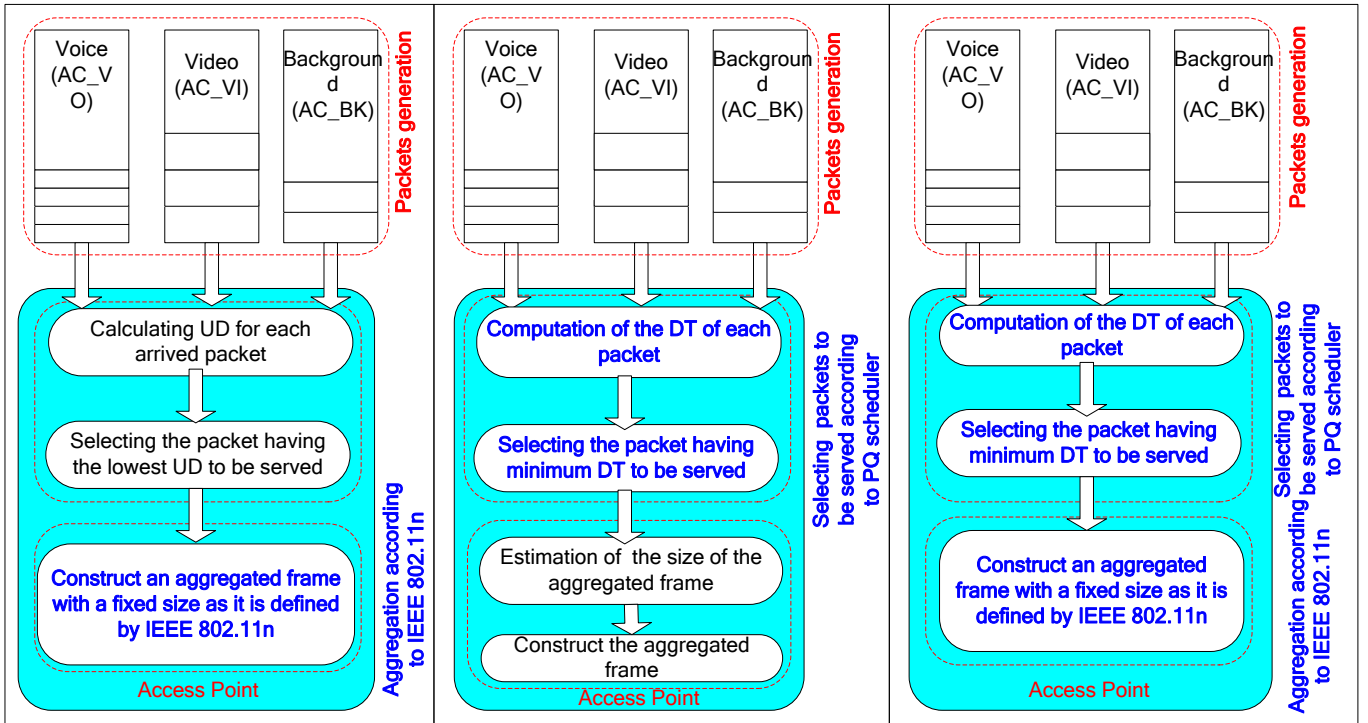


Figure.11(a) : Urgency delay scheduler Figure 11(b): Optimized aggregation scheduler Figure 11(c): PQ scheduler
Figure.11. Studied schedulers

- Urgency Delay (UD) scheduler*: In this scheduler, packets are scheduled based on their UD; the packet which has the lower UD delay in the highest priority queue will be the first scheduled. Once selecting packets, the AP constructs the aggregated frame with a fixed size as it is defined by IEEE 802.11n. Contrarily to DFA scheduler where the size of the aggregated frame is dynamically adjusted.
- Optimized AGGregation (OP.AGG) scheduler*: In this scheduler, packets are scheduled based on their DT; the packet that has lowest DT will be the scheduled. The aggregated frame size will be adjusted according to DT of the first selected frame.
- PQ scheduler*: In this scheduler, packets are scheduled based on their DT, and then the aggregated frame size is fixed as it is defined in IEEE 802.11n.

The parameters of each scheduler as well as the size of the aggregated frame are listed in Table .2.

Table.2. Different scheduler parameters

	Scheduler Fig.11(a)	Scheduler Fig.11(b)	Scheduler Fig.11(c)
Description	UD scheduler	OP.AGG scheduler	PQ scheduler
Parameters	UD, DT, WD	DT	DT
Aggregated frame size	Fixed according to IEEE 802.11n standard	Adjusted according to DT	Fixed according to IEEE 802.11n standard

8 Performance analysis

In this section, we present the results of DFA scheduler performance evaluation obtained through simulations. We analyzed the efficiency of the proposed *DFA scheduler* in comparison with *UD scheduler*, *OP.AGG scheduler*, and *PQ scheduler*. Firstly, we describe the simulations settings as well as the model of traffic generation. After that, we discuss the DFA scheduler performance in terms of data rate, delay, and percentage of dropped frames. The considered voice, video, and streaming parameters are listed in Table.3, and the main MAC parameters are listed in Table.4.

8.1. Traffic Generation

As we mentioned in the above section, packets are generated according to a random distribution. We suppose that there are an AP and three sources types: voice, video, and streaming which generate packets along the simulation duration. All packets belonging to same AC will be placed in the same queue. We choose the A-MPDU aggregation mechanism in all these schedulers with an aggregated frame size equal to 32767 bytes as it is defined by IEEE 802.11n. Main simulation parameters are listed in Table.4.

Table.3. Traffic parameters

	Voice	Video	streaming
payload size/frame (B)	160	660	1500
DT (ms)	50	150	250

Table.4. Simulation parameters

SIFS	16 μ s
DIFS	34 μ s
MAC header	36 bytes
MPDU delimiter	4 bytes
FCS	4 bytes
BA	112 bits + PHY header
Slot duration	9 μ s
Basic rate	54 Mbps
PHY rate	216 Mbps

8.2. Delay and throughput analysis

In this section, we aim to investigate the performance of the DFA scheduler in term of delay and throughput behavior. The measured delay is considered as the elapsed delay between the instant of packet generation until its serving instant. Figures 12, 13, and 14 plot the delay variation among the time for voice, video, and streaming packets respectively. On the other part, we consider the throughput (called also traffic rate) is as: (4)

$$Th = \frac{L_{payload_packet}}{Delay_{transmission}} \quad (4)$$

Where $L_{payload_packet}$ refers to the frame payload size, and $Delay_{transmission}$ corresponds to required delay to transmit this packet. This delay includes the inter-frame spacing delays such as *DIFS* and *SIFS*, the backoff delay $T_{backoff}$, the required delay to transmit a block acknowledgment, T_{BAR} and T_{BA} , the waiting delay WD , and the transmission delay of the aggregated frame in which this packet is encapsulated L_{agg} . Therefore, throughput is given by (5):

$$Th = \frac{L_{payload_packet}}{DIFS + T_{backoff} + WD + T_{phy_hdr} + \frac{L_{agg}}{PHY_{rate}} + 2SIFS + T_{BAR} + T_{BA}} \quad (5)$$

Figures 15, 16, and 17 draw the average throughput variation along the simulation duration for voice, video, and streaming packets respectively.

For all traffic types, we observe that:

- i) When the proposed *DFA scheduler* is enabled, the serving delay is highly decreased and the throughput is greatly decreased.
- ii) Scheduling packets based on *PQ scheduler* badly affect delay and throughput. In fact, it causes highest delays with lowest rates.
- iii) *UD scheduler* provides lower delay and better throughput compared to *OP.AGG scheduler*.
- iv) All delays values satisfy the required QoS delay since packets that go above this constraints are systematically dropped.

Our proposed *DFA scheduler* provides the best delay and throughput performance compared to all other schedulers. This is explained by the two tasks: scheduling packets according to their urgency delay, and dynamically adjusting the size of the transmitted aggregated frame.

To further evaluate the impact of using these two parameters on the network, we interpret in the following the performance of each task separately. We will firstly examine the performance of schedulers based on UD parameter (such as *DFA and UD schedulers*) versus schedulers based on DT parameter (such as *PQ and OP.AGG schedulers*).

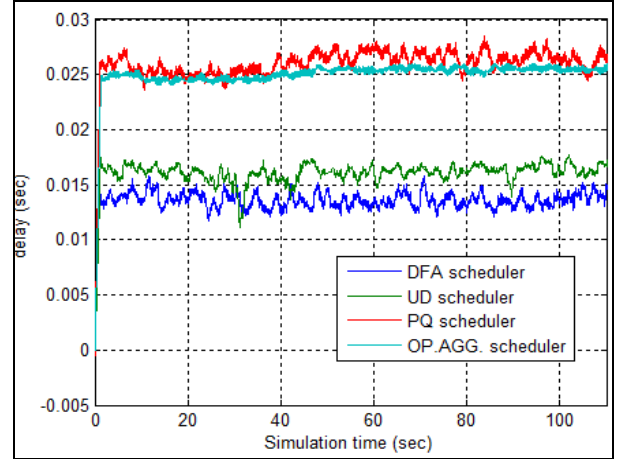


Figure.12. Delay variation for voice packets

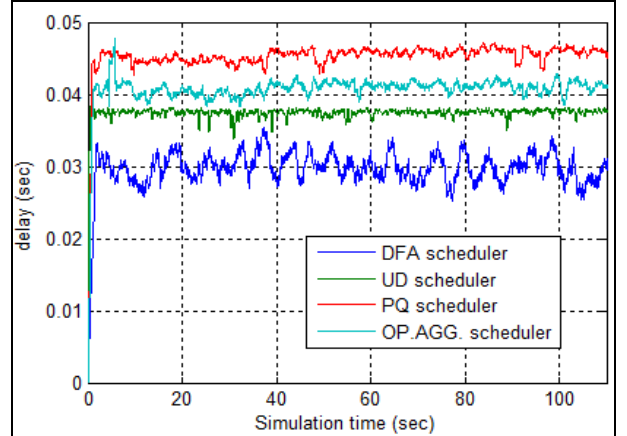


Figure.13. Delay variation for video packets

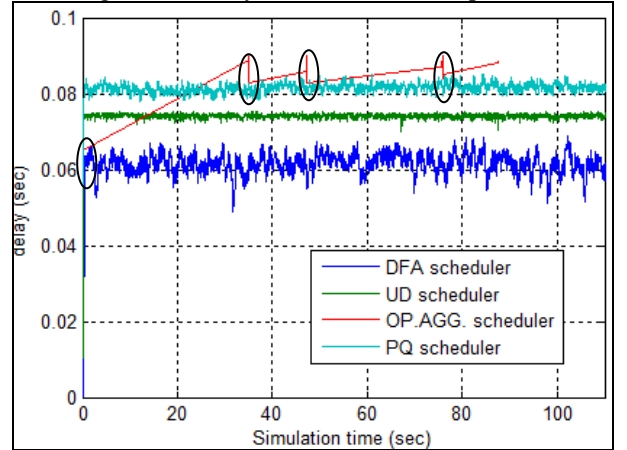


Figure.14. Delay variation for streaming packets

On the other part, we will examine the performance of schedulers that adjust the aggregation frame size (such as *DFA* and *OP.AGG* schedulers) versus schedulers that uses a fixed length of the aggregated frame (such as *PQ* and *UD* schedulers). Moreover, we aim to identify which parameter UD or adjustment of the frame aggregation has the more significant impact on the network performance.

From delay and throughput curves, we note that scheduling packets based on their UD is better than scheduling packets based on their DT parameter, since higher delay with lower rates are obtained with the latter one. As a result, *DFA* and *UD* schedulers provide better delay and throughput performances compared to *PQ* and *OP.AGG* schedulers.

Such performance is obtained by considering the waiting delay of each packet. In fact, serving packets based on DT, the instant from which the entered packets are waiting in the priority queue is not considered. So, packets that are not selected must wait to be scheduled. In that case, they will be scheduled lately or they will be dropped if their waiting delay goes above the tolerated delay to serve a packet. Knowing that the serving delay is the difference between the instant “t” at which the packet is selected, and the instant of arriving at the priority queue t_o . Hence, the serving delay is increased with *PQ* and *OP.AGG* schedulers. Contrariwise, based on UD parameter, *DFA* and *UD* schedulers consider the waiting delay when scheduling packets. In that case, oldest packets will be the first scheduled and so the serving delay of these packets will be reduced. For the above reasons, and knowing that rate is inversely proportional to delay, *DFA* and *UD* schedulers provide lower delays with better rates compared to *PQ* and *OP.AGG* schedulers.

On the other side, in order to analyze the impact of adjusting the payload size of the aggregated frame on delay and throughput behaviors, we compare *DFA* scheduler versus *UD* scheduler, and *PQ* scheduler versus *OP.AGG* scheduler. As first comment, we mention that adjusting the aggregated frame size causes a reduced delay with an increased throughput for all traffic types. In fact, *DFA* scheduler provides lower delays and better rates compared to *UD* scheduler. Similarly, *OP.AGG* scheduler is better than *PQ* scheduler.

These performances are explained by the optimization of the payload of the aggregated frame according to the QoS delay of the first selected packet which was encapsulated in the aggregated frame. Indeed, when the size of the aggregated frame is adjusted considering the left time to serve frames, the delay of serving is reduced. Otherwise, when using a fixed aggregated frame size, selected packets will be scheduled regardless the above factor. Therefore, the serving delay may be increased with a fixed aggregated frame. From this fact, we observe, for all traffic types, that schedulers using a fixed aggregated frame have higher delay with low rate compared to those using an adjusted one.

Consequently, our proposed *DFA* scheduler provides lowest delays with highest rate for all traffics, since as we

aforementioned it combines the two tasks which have a positive impact on delay and throughput performance.

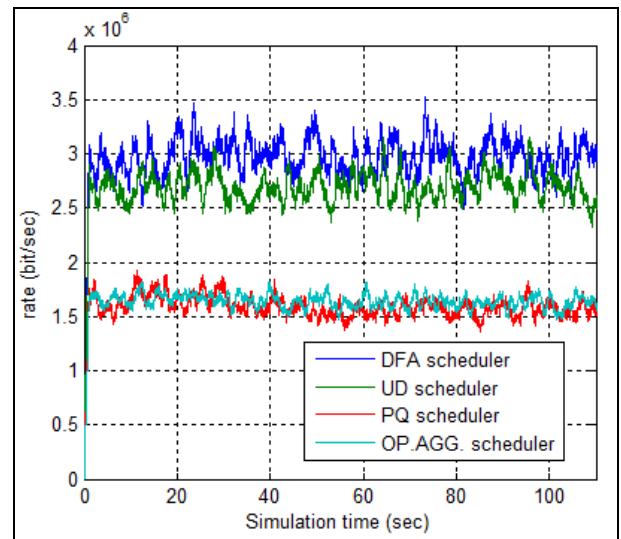


Figure.15. Throughput for voice packets

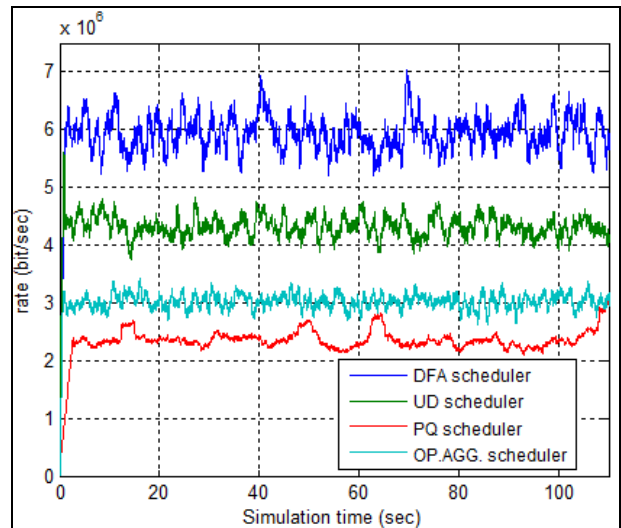


Figure.16. Throughput for video packets

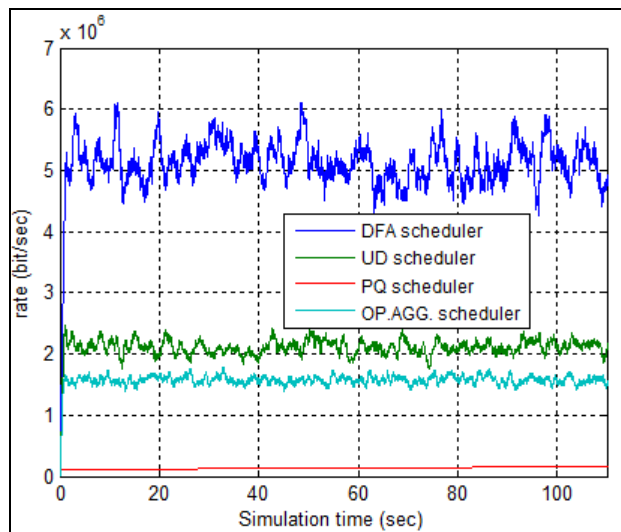


Figure.17. Throughput for streaming packets

We have proved by simulations as well as by scheduler analyses that considering the waiting delay of each packet when serving it, as well as adjusting the aggregation frame, are two main factors that improve the network performance. However, we may specify which factor of them has the more significant impact on reducing delay and increasing rate. Within this framework, we compare the behavior of *UD scheduler* and *OP.AGG scheduler*. The first one serves packets according to UD parameter and fill an aggregated frame with a fixed payload size. So that, it provides lower delays with better rates compared to the latter scheduler that uses DT with an optimized aggregated frame. Thus, scheduler's performances depend to a large degree on considering the waiting delay of scheduled packets. In fact, when serving packets according to their UD, the serving delay will highly be reduced since the QoS delay is respected. Otherwise, if the size of the aggregated frame is adjusted regardless the tolerated delay to serve selected packets, the serving delay may increase.

For streaming packets, we note that serving delay with *PQ scheduler* behaves differently to other schedulers, since only packets arriving at circled instants are scheduled. In addition, there is no scheduled packet after 90 sec.

8.3. Dropped packet analysis

We intend by this section to analyze the effect of the proposed *DFA scheduler* on the number of scheduled packets in comparison with other schedulers. Figures 18, 19, and 20 draw the percentage of dropped packets along time for voice, video, and streaming respectively.

Firstly, we mention that, for all traffic types, our proposed *DFA scheduler* guarantees lowest values of dropped packets compared to the other schedulers. Therefore, the *DFA scheduler* improves the number of scheduled packets in the network. Furthermore, it is obvious that the highest number of dropped packets is achieved with *PQ scheduler*.

Moreover, we note that, for video and streaming, the number of dropped packets is as important as *PQ scheduler*.

More packets are scheduled with *DFA scheduler* is explained by considering the waiting delay of each packet. By this factor, the scheduler respects the QoS delay of each packet. In fact it reduces the number of accumulated packets that will be rejected once the tolerated serving delay is exceeded. Contrarily to *PQ scheduler* that drops more and more packets due to not considering the priority to serve oldest frames.

From the above curves, we firstly mention that *PQ scheduler* rejects most of the streaming packets, up to 72% video packets, and fewer than 8% voice packets. When scheduling packets based on DT, voice packets will be the most scheduled particularly under saturated network condition. Indeed, voice packets have the lower DT compared to video and streaming packets. Hence, video and streaming packets are accumulated until some will be rejected.

Based on our proposed *DFA scheduler*, around 25% streaming packets, 15% video packets, and only 2% voice packets are dropped. Therefore, *DFA scheduler* has a positive impact on serving video and streaming packets and it overcomes the inefficiency of *PQ scheduler*. Indeed, consi-

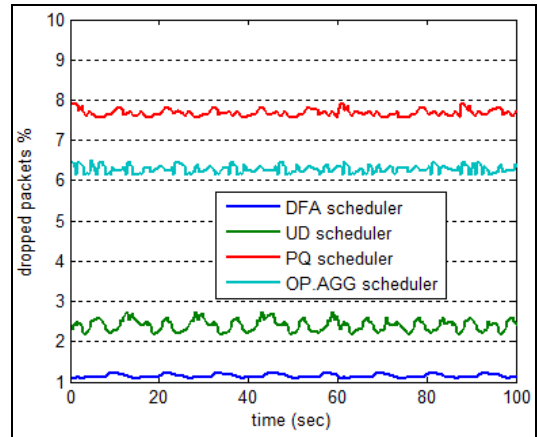


Figure.18. Voice dropped packets

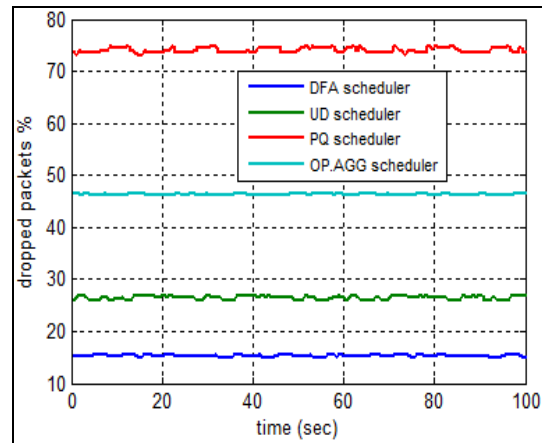


Figure.19. Video dropped packets

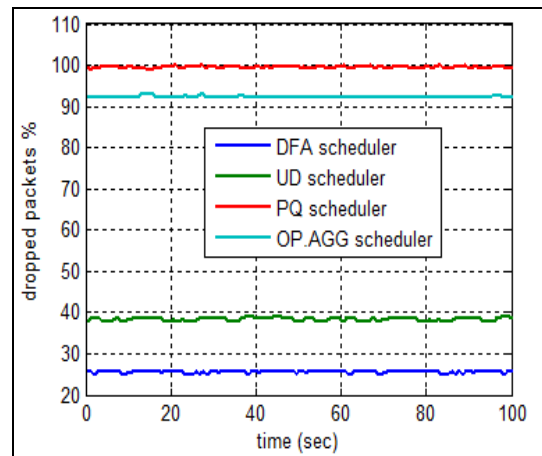


Figure.20. Streaming dropped packets

dering the waiting delay has a positive impact on the number of scheduled packets. By this task, there is more chance to select packets which are coming first, so serving them with a reduced delay.

From this analysis, we conclude that a maximum number of scheduled packets is granted when frames are scheduled based on our proposed *DFA scheduler*, while *PQ scheduler* causes an important number of rejected packets. For voice applications, worst performances are obtained with *UD scheduler* since with *PQ scheduler* these packets have the larger probability to be scheduled. Moreover, we note that a highest number of dropped packets is caused by *PQ sched-*

uler for streaming and video applications. Hence, when serving packets based on their DT, streaming and video are mostly non-scheduled since voice packets have always the highest priority to be scheduled.

8.4. Discussion

In this section, we summarize the overall performance of the proposed *DFA scheduler* in term of delay, throughput, and dropping packets for each traffic type as it is listed in Table.5. All the above results prove that better QoS performances are obtained with *DFA scheduler*. We mention that *DFA scheduler* provides better QoS delay and throughput performance, as well as it improves the number of scheduled packets in the network. Such performances are explained by scheduling packets based on their UD, as well as optimizing the aggregated frame size. In fact, using UD avoids storing high number of packets in the priority queue as well as increasing the waiting delays of earliest entering packets mainly for video and streaming.

Table.5. QoS evaluation of different schedulers

Sched- uler QoS	DFA sched- uler	UD sched- uler	OP.AGG sched- uler	PQ sched- uler
delay	lowest	low	important	highest
Throughput	highest	important	low	lowest
Frame Drop	lowest	low	high	highest

9 Conclusions

In this paper we have proposed a dynamic frame aggregation scheduler for high throughput WLAN 802.11n. Firstly, we were motivated from PQ scheduler and aggregation mechanism drawbacks. Indeed, in saturated network these two concepts fail in providing QoS for delay sensitive applications such as voice and video. Mainly, PQ scheduler causes increased waiting delays for non-scheduled packets. Similarly, supplementary delays are created with the IEEE 802.11n aggregation mechanism. To overcome these drawbacks, we have proposed a DFA scheduler which defines new parameters to serve packets. By this scheduler, packets are scheduled according to their urgency delay, and then the aggregated frame will be constructed based on the same parameter. Therefore, waiting delays are reduced and the aggregated frame size is optimized. Simulation results prove that DFA scheduler is able to guarantee delay and throughput requirement for real time flows particularly of voice and video. Indeed, serving packets based in their urgency delay is a key to maintain required QoS.

References

[1] IEEE, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std 802.11-1999.

[2] IEEE, Part 11, [Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band](#), IEEE Std 802.11g-2003.

[3] IEEE, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, November 2005.

[4] IEEE 802.11n-2009 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, 2009.

[5] 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 Amendment 2: MAC Enhancements for Robust Audio Video Streaming, (2012).

[6] Jeng, F., Wanjiun, L., Meng, C., "A Differentiated Service Model for Enhanced Distributed Channel Access (EDCA) of IEEE 802.11e WLANs", *Mobile Networks and Applications*, vol. 12, no. 1, pp. 69-77, 2007.

[7] Tinnirello, I., Choi, S. "Temporal fairness provisioning in multi-rate contention-based 802.11e WLANs." in *Proc. IEEE WoWMoM, June 2005*, pp.220-230.

[8] Lin, X., Cai, X., Xuemin, S., Jon, W., Lin, C., "Supporting voice and video applications over IEEE 802.11n WLANs", *Wireless Networks*, vol. 15, no. 4, 2009, pp. 443-454.

[9] Konstantinos, P., Theodoros, S., Henrik, L., Nitin, V., "Experimental characterization of 802.11n link quality at high rates" in *Proc. WINTECH*, 2010, pp. 39-46.

[10] Richard van Nee, (2011) "802.11n: The Global Wireless LAN Standard" *Springer, Globalization of Mobile and Wireless Communications, Signals and Communication Technology*, pp. 103-118, 2011.

[11] A. Saif, M. Othman, S. Subramaniam and N. AbdulHamid, "Impact of aggregation headers on aggregating small MSDUs in 802.11n WLANs" in *Proc. ICCAIE*, 2010, pp 630 – 635.

[12] Yaw-Wen Kuo "Design and evaluation of a high throughput MAC with QoS guarantee for wireless LANs", in *Proc. MICCC*, 2009, pp 869 – 873.

[13] Wang, Lan, Min, Geyong; Kouvatso, Demetres D.; Jin, Xiaolong "An analytical model for the hybrid PQ-WFQ scheduling scheme for WiMAX networks", in *Proc. VITAE*, 2009, pp. 492-498.

[14] A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling," *Quality of Future Internet Services*, pp. 1-13, April 2000.

[15] Hammouri, M.M.; Daigle, J.N.; "A distributed scheduling mechanism to improve quality of service in IEEE 802.11 ad Hoc Networks", in *Proc. ISCC*, 2011, pp. 1-6.

[16] Hongli Luo, Mei-Ling Shyu, "An Optimized Scheduling Scheme to Provide Quality of Service in 802.11e Wireless LAN", in *Proc. IEEE ISM*, 2009, pp. 651-656.

[17] Eun-Chan Park Dong-Young Kim Chong-Ho Choi Jungmin So, "Improving Quality of Service and Assuring Fairness in WLAN Access Networks", *Mobile Computing, IEEE Transactions*, vol. 6, no. 4, pp 337-350, 2007.

[18] Selvam, T.; Srikanth, S.; "A Frame Aggregation Scheduler for IEEE 802.11n", in *Proc. NCC*, 2010, pp. 1-5.

[19] Davide Astuti "Frame Handling" *Seminar on Transport of Multimedia Streams in Wireless Internet*, pp 1-10.

- [20] Cedric Gueguen , Sebastien Baey: "A Fair Opportunistic Access Scheme for Multiuser OFDM Wireless Networks" *In Proc. European Association for Signal Processing (EURASIP), Journal on Wireless Communications and Networking. Special issue: "Fairness in Radio Resource Management for Wireless Networks", 2009.*
- [21] Peyman Teymoori, Nasser Yazdani, Seyyed Alireza Hoseini, Mohammad Reza, "Analyzing Delay Limits of High-Speed Wireless Ad hoc Networks Based on IEEE 802.11n." *in Proc. IST, 2010, pp. 489-495.*
- [22] Anwar Saif, Mohamed Othman, Shamala Subramaniam, NrAsilaWati AbdulHamid , "Impact of aggregation headers on aggregating small MSDUs in 802.11n WLANs". *In Proc. ICCAIE, 2010, pp. 630 – 635.*
- [23] Hoffmann, O.; Schaefer, F.-M.; Kays, R.; Sauer, C.; Loeb, H.P.; "Prioritized medium access in ad-hoc networks with a SystemClick model of the IEEE 802.11n MAC". *In Proc. PIRMC, 2010, pp. 2805-2810.*
- [24] Yaw-Wen Kuo; Tsern-Huei Lee; Yu-Wen Huang; Jing-Rong Hsieh; "Design and evaluation of high throughput MAC with QoS guarantee for wireless LANs". *in Proc. MICC, 2009, pp 869-873.*
- [25] Selvam, T.; Srikanth, S.; "A Frame Aggregation Scheduler for IEEE 802.11n". *in Proc. NCC, 2010, pp 1-5.*
- [26] Hajlaoui, Nasreddine ; Jabri, Issam; Taieb, Malek; Benjema, Maher "A frame aggregation scheduler for QoS sensitive applications in IEEE 802.11n WLANs" *in Proc. ICCIT, 2012, pp 221-226*
- [27] Jibukumar, M.G.; Datta, R.; Biswas, P.K.; "New Frame Aggregation Schemes for Multimedia Applications in WLAN." *In Proc. NOMS, 2010, pp 424-431*
- [28] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "Aggregation With Fragment Retransmission for Very High-Speed WLANs". *IEEE/ACM Trans. Netw., vol. 17, no. 2, pp. 591-604. 2009*