University of Piraeus

Department of Digital Systems

MSc “Digital Communications and Networks”

Performance study of 802.11n WLAN and MAC enhancements in ns-3

Master thesis

by

Ioannis Selinis

Supervisors:

Prof. Panagiotis Demestichas

Dr Seimak Vahid

Piraeus 2014

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have acknowledged all the sources of information which have been used in this thesis.

Acknowledgments

I would like to thank my supervisor Dr Seimak Vahid for his guidance and support and all the people at CCSR and University of Piraeus who made it possible to start and pursue this MSc thesis to completion: Panagiotis Demestichas, Kostas Tsagkaris, Marcin Filo, Konstantinos Katsaros.

I also offer my sincere gratitude to people, who supported me and gave me the strength to carry on: my family, my friends, and those I disappointed, but they stood by me during this year.

Finally, many thanks to Nikos, and Niklas.

Living is easy with
eyes closed.

Abstract

Recently, significant research effort has been focused in the design of Wireless Local Area Networks (WLANs). WLANs are attractive because they offer important advantages, including interoperability, mobility and cost-effective deployment. However, the ever-increasing demand on very high data rates and coverage have boosted the expansion of IEEE 802.11 a/b/g WLANs to new standards; IEEE 802.11n.

In this thesis, performance of the IEEE 802.11n is studied with focus on Medium Access Control (MAC) layer and Physical Layer (PHY) enhancements. The new features that IEEE 802.11n introduced in MAC layer are the frame aggregation schemes, the Block Ack and reverse direction. The new features aim in reducing the overhead of the MAC layer. While, in the PHY layer uses MIMO technology and channel bonding to improve throughput and coverage of WLANs.

Many simulations were performed to provide a better knowledge of these new modules, in various network environments. The simulation tool used is the Network Simulator tool known as NS-3. Many improvements and implementations were implemented in NS-3 to support these new features that IEEE 802.11n introduced.

Table of *Contents*

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In the recent years, much interest has been involved in designing wireless local area networks (WLANs) due to its advantages; such as interoperability, mobility and cost-effective deployment. However, the increasing demand for higher data rates and coverage has boosted the expansion of IEEE 802.11 a/b/g WLANs to new standards operating in unlicensed bands (2.4, and 5GHz), and in other available spectrums. IEEE 802.11n amendment introduced new technologies in MAC and PHY layers, improving network performance.

The new features that IEEE 802.11n introduced in MAC layer were the frame aggregation schemes, the Block Ack and reverse direction that aim to reduce the overhead of that layer. These amendments increased throughput to 65 Mbps, while, in PHY layer the use of MIMO technology and channel bonding pushed throughput further, reaching its maximal value of 600 Mbps.

Although, 802.11n was presented in 2009 and two major enhancements on last year; IEEE 802.11ac and IEEE 802.11ad, performance of the former has not yet, been fully exploited. In this thesis, we conducted research of IEEE 802.11n WLAN protocol and evaluate its performance under various network environments. For the evaluation of 802.11n performance a Network Simulator tool was used, known as NS3.

1.2 Challenging issues

The main challenges that this research is considering are:

- Implementing new MAC features in a software platform, NS3. Evaluate the performance of these features under various network topologies. Therefore, it isn’t easy to extract all the information that packets carry in order to test performance of aggregation schemes. The solution to that is to add some additional tags in each packet in order to get accurate results and test them under different conditions.

- There are many factors that can influence the performance of 802.11n, such as APs, EDCA parameters, distance, hardware, environment, and interference from other stations. Therefore, it will be difficult to evaluate the effects of a particular factor on the system’s performance, and find a suitable value of an EDCA parameter in order to get better performance. Thus, simulations are repeated many times, covering many system types.

1.3 Contributions

This thesis evaluates the efficiency of MAC enhancements of 802.11n WLAN protocol. Furthermore, the network performance is examined when the EDCA parameters are differentiated from their initial values under fully buffered and unsaturated conditions.

The contributions are as follows:

- Analysis of system performance of IEEE 802.11n WLAN under fully buffered and unsaturated conditions.
- Performance evaluation of 802.11n under various values of EDCA parameters.
- Gathering valuable data for future study on 802.11n and its successors.

The structure of this thesis is:

Chapter 2 includes some background theory about IEEE802.11 standard, a brief presentation of MAC and PHY layers. In addition to this, an overview of the related work, according to the literature is give, focusing mainly on the mechanisms that are used in MAC layer.

Chapter 3 provides a picture of the software tool that is used in this research.

Chapter 4 covers all the work that was done in NS3 in order to support TxOP, Adaptive fragmentation, A-MPDU, and Two-Level aggregation scheme. Moreover, it describes the necessary simulations that took place and gives a comprehensive analysis of the results. It also includes a brief description of the software tool and its WiFi structure.

Chapter 5 gives a brief discussion of the further work that can be followed. Furthermore, it summarises the results and presents the conclusions that can be drawn from this research project.

Chapter 2

Background Theory and Related work

2.1 Frame Format

There are three main frame types defined in the IEEE 802.11; the Management frames, Control frames, and Data frames. Each frame comprises a set of fields as a MAC PDU, a Preamble, and a PLCP header. Figure 1 illustrates the 802.11 frame format.

The Mac Protocol Data Unit (MPDU) consists of a MAC header, a frame body, and a frame check sequence. The MAC header contains the addressing information, the state of the frame and some additional useful information. There are 4 addressing fields, but in some types of networks only the first two or three are used. The frame control field includes 16 subfields, like the number of retries, the encryption that is used, a flag if fragments or more data exist for that station, and the type of that frame. Also, a sequence control field is a part of the MAC header, indicating the sequence number of a packet and the number of each fragment (if fragmentation has occurred).

The frame body follows the MAC header which contains the useful payload. The size of this field varies from 0 to 2312 bytes. The last component of the MPDU is the FCS field. It contains a number that is calculated by the source and added to the frame. When the destination node receives the frame, the FCS number is recalculated and compared with the existing one. If these two numbers do not match, an error is assumed. The destination tries to detect it according to the 32-CRC, else discards the packet.

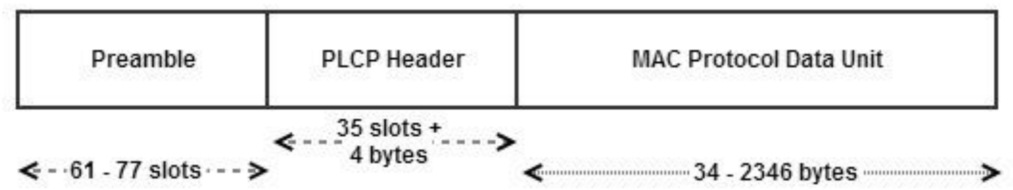

Figure 1: IEEE 802.11 frame format.

A new field was introduced in the IEEE 802.11e standard, named QoS Control field. This element is added in the tail of the MAC header, providing QoS-related information about the frame. Its length is 2 bytes and consists of 5 subfields. The QoS Control field is present in all data frames in which the QoS subfield is set to 1 [9].

The 802.11n brought some major changes in the frame format, as the insertion of new fields and the change in the length of the frame body. Three types of preamble are defined according to [10]:

- Non-HT legacy PPDU
- HT-Mixed PPDU
- HT-Greenfield PPDU

The non-HT legacy PPDU is used whenever there are non-HT stations in the system, while the HT-Mixed PPDU is used when legacy (802.11a/b/g) and HT (802.11n) devices coexist in the same BSS. These modes are mandatory, while the HT –Greenfield is optional, operating in networks with only HT nodes. All these types are shown in the *Figure 2*.

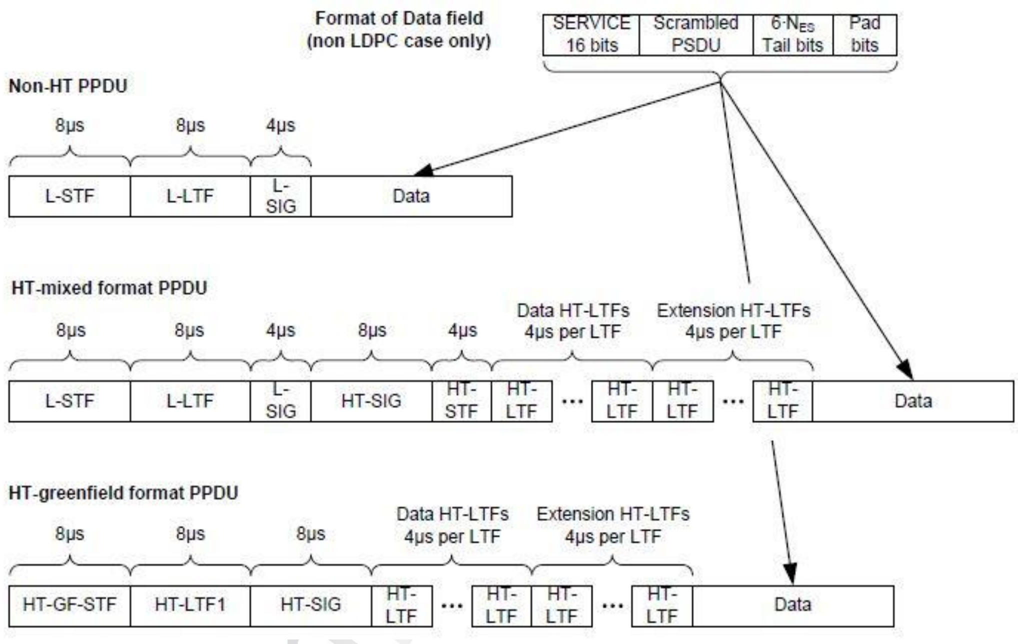

Figure 2 : PPDU format ([10], Figure 20-1).

For more details about the subfields that are depicted in Figure 10, refer to Clause 20 in [10].

In the MAC header, an extra field is added at the end of the header; the HT Control. The HT Control field is present in QoS data and management frames. Its length is 4 bytes and consists of 9 subfields. The last major change in the MAC frame format is the frame body length that is up to 7951 bytes. *Figure 3* illustrates this format, with all the fields that are included in a MPDU frame.

Address 1 contains the MAC address of an intermediate node (usually an AP) that identifies a MAC group of entities, while the transmitter's address is carried in the Address 2 field. The 3rd addressing field contains the MAC address of the final recipient and the 4th is used only in Wireless Distribution Systems.

Bytes:	2	2	6	6	6	2	6	2	4	0 - 7951	4
	Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	HT Control	Frame Body	Frame Check Sequence (FCS)

<------------------------------------ MAC Header ------------------------------------>

Figure 3 : MAC Frame format.

2.2 Medium Access Control (MAC) Layer

Medium Access Control (MAC) layer provides, mainly, addressing and channel access control which make multiple stations in a system communicate. In terms of addressing and channel access, IEEE 802.11 shows similar operation to IEEE 802.3 (Ethernet). However, it differs from Ethernet, as wireless channel varies from wired in many ways. As a result, a number of new features have been introduced to manage mobility, hidden nodes and channel conditions.

Legacy MAC protocol defines two different access methods; Distributed Coordination Function (DCF) and Point Coordination Function (PCF). The latter is a centralized method above DCF and is optional for the stations, while the former is the basic access method for all wireless stations. DCF is a distributed algorithm where every node senses the channel for a period, before a transmission. Another optional feature of the MAC layer is the fragmentation, preventing the stations from occupying the channel for long time of periods when they operate in low PHY rates.

2.2.1 Legacy MAC Layer

2.2.1.1 Distributed Coordination Function (DCF)

As mentioned before, the most commonly used scheme is DCF. *Figure 4* shows a simple example of Basic Access mechanism using the DCF process. A station operating in DCF with a new packet to transmit, listens the channel for a period of time equal to Distributed Interframe Space (DIFS). If the channel is sensed to be idle for that period (DIFS), the station transmits. Otherwise, if the channel is sensed to be busy (immediately or during the DIFS), the transmission of the packet is deferred until the channel is sensed to be idle for a period of DIFS. The probability a station transmits in a randomly chosen slot time is given by [5]:

$$\tau = \frac{2\,(1-2\mathrm{p})}{(1-2\mathrm{p})(\mathrm{W}+1)+pW(1-(2p)^{m}))}, \qquad (1)$$

where *W* is the Contention Window and *p* is the conditional collision probability. When the backoff stage is equal to 0, no exponential backoff is considered and the probability, τ, is given by:

$$\tau = \frac{2}{W+1}. \qquad (2)$$

Since there is always a chance of two or more stations transmitting at the same time, after a DIFS interval, a collision might occur. To avoid collisions, a station generates a random backoff time before transmitting and after the DIFS period. Here, lies the Collision Avoidance feature of the protocol (CSMA/CA). In addition, an extra period ("post" backoff) is initiated by a station, after its successful transmission. This Post backoff reduces the probability of capturing two consecutive times the channel, by the same node. It also ensures at least one backoff interval between two successive transmissions.

Backoff period is slotted and a station is allowed to transmit only at the beginning of each slot time. The size of a slot time depends on the Physical Layer protocol. Many backoff algorithms have been proposed aiming at minimizing delays, collisions and improving network capacity. However in the IEEE 802.11 DCF scheme, a Binary Exponential Backoff (BEB) is adopted.

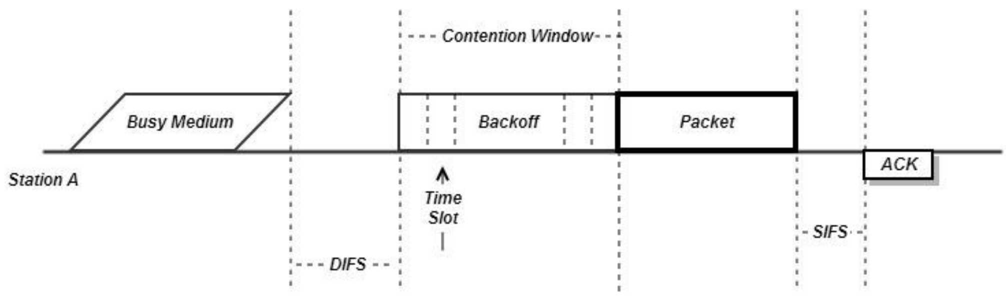

Figure 4 : Basic Access mechanism using the DCF scheme.

In each packet transmission, a backoff time is selected uniformly between 0 and the Contention Window (CW). After each unsuccessful transmission, the contention window is doubled, up to its maximum value, while it is reset to its minimum value after a successful transmission. The bounds are defined as CW_{min} and $CW_{max} = 2^m * CW_{min}$. At each transmission, the backoff time is uniformly chosen in the range [0, CW - 1].

$$\begin{cases} CW \leftarrow \min(2 * CW, CW_{max}), \text{ upon collision} \\ CW \leftarrow CW_{min}, \text{ upon success.} \end{cases} \tag{3}$$

According to [1], a multiplicative increase and decrease algorithm scheme (MIMD) is used to change the contention window. A similar mechanism is presented in [2], with a Multiplicative Increase and Linear Decrease (MILD) algorithm. These only deal with the throughput without taking into account any fairness issues. In the MILD scheme, a collided node increases its CW by multiplying it by 1.5, while a successful node decreases its CW by one unit. The current CW is included in each transmitted MSDU (additional overhead) and in each successful transmission all the nodes copy and use that CW. The operation of the MILD algorithm can be summarized as follows:

$$\begin{cases} CW \leftarrow \min(1.5 * CW, CW_{max}), \text{ upon collision} \\ CW \leftarrow \max(CW - 1, CW_{min}), \text{ upon successful tx} \\ CW \leftarrow CW_{copy}, \text{ for overhearing nodes after a successful tx.} \end{cases} \quad (4)$$

A similar approach was presented in [3], taking into account the fairness issues, but without using the CW copy mechanism that is used in [2]. A Linear/ Multiplicative Increase and Linear Decrease (LMILD) backoff algorithm is used, which is based on the knowledge of the packet collisions on the channel. When a collision occurs, the senders increase their CWs multiplicatively, while neighboring nodes increase their CWs linearly. After a successful transmission every node decreases its CW linearly. LMILD scheme improves throughput when a great number of nodes operate in the network, while it fails when the number of nodes is less than 10. The operation of the LIMLD scheme is summarized as follows:

$$\begin{cases} CW \leftarrow \min(m_c * CW, CW_{max}), \text{ after collisions} \\ CW \leftarrow \min(CW + \ell_c, CW_{max}), \text{ neighboring nodes after collision} \\ CW \leftarrow \max(CW - \ell_s, CW_{min}), \text{ after a successful tx.} \end{cases} \quad (5)$$

H. Minooei and H. Nojumi introduced a backoff method, when stations are unsaturated [4]. Two changes are made in the BEB algorithm. The first is to avoid or reduce contention among stations under unsaturated conditions, by choosing from the following intervals:

$$\begin{cases} [CW_{i-1}, CW_i], \text{ for } i = 1, 2, \ldots, m \\ [1, CW_0], i = 0. \end{cases} \quad (6)$$

The second change has to do with the decrement of the backoff counter after a successful transmission. Instead of resetting the CW to its minimum value, the CW is decreased by one unit, because for that period of time the optimal backoff stage is the one that is reached.

G. Bianchi in [5] presented an analytical model, computing the saturation throughput of the IEEE 802.11 DCF. It is shown that the performance of the basic method depends on the CW_{min} and the number of stations in the WLAN. Contrarily, when RTS/CTS mechanism is used, the performance is independent from the system parameters and outperforms over the basic mode in most of the cases. Furthermore, according to (2), the backoff window that maximizes the system throughput is defined as:

$$W_{opt} \approx n\sqrt{2T_c}\,, \tag{7}$$

where T_c is the duration of packet collisions on the channel in the terms of the number of time slots and where n is the number of nodes. In the IEEE 802.11 standard, the values of Contention Window and the number of backoff stages are fixed and predefined and cannot be changed as the number of the stations varies (*Table I*). The number of backoff stages can be derived by the equation of the CW (for the CW_{min} and CW_{max}) defined by:

$$CW = 2^m - 1, \tag{8}$$

where for simplicity it can be stated that it is equal to

$$k = m_{max} - m_{min} + 1, \tag{9}$$

where m_{max} is the m when CW = CW_{max} and m_{min} when the CW = CW_{min}.

Table I
Slot Time, Backoff Stage (m), Minimum and Maximum Contention Window values for the IEEE 802.11a/b/g.

PHY standard	Slot Time	CW_{min}	CW_{max}	Backoff Stages
802.11a	9 μs	15	1023	7
802.11b	20 μs	31	1023	6
802.11g	9 or 20 μs	15 (31)	1023	7 (6)

Since the CSMA/CA does not rely on the capability of the stations to detect a collision, the receiver should inform the stations about the state of the transmission. After a successfully reception of a packet, the destination station sends an ACK to the transmitter after a short period of time called Short Interframe Space (SIFS). SIFS is shorter than DIFS (as seen on *Table II*), thus no other station will be able to transmit until the end of the ACK transmission. If the transmitter does not receive an ACK within a specified period, assumes that a collision occurred in the channel and reschedules to re-send the packet, after a random backoff. DIFS duration can be calculated using:

$$DIFS = SIFS + 2(Slot\ Time). \tag{10}$$

Table II
IEEE 802.11 Interframe Spacing

	802.11a	**802.11b**	**802.11g**
SIFS (μs)	16	10	10
DIFS (μs)	34	50	28 or 50

Another feature of the DCF protocol is a virtual carrier sensing, known as Network Allocation Vector (NAV). The NAV contains the duration of a transmission cycle and intents to inform the rest of the nodes for the time that the medium will be occupied by that station, in order to save power from carrier sensing. This information is included in the RTS/CTS frames before the data exchange. In [6] an adaptive backoff algorithm was proposed based on the NAV. The NAVC algorithm adjusts according to the network traffic and the channel conditions, combining the BEB mechanism.

The RTS/CTS mechanism is another technique of the DCF. It is an optional feature but offers a lot of advantages due to its four-way handshaking. In conjunction with NAV, the hidden node problem is partially solved. A node that will receive a CTS frame, without sending RTS, knows that the channel is occupied by another node. Also, it reduces the length of the frames involved in the contention window. Collisions may occur but only on the RTS frames and the senders can early detect them by the lack of CTS responses. A simple example of the RTS/CTS technique is depicted in *Figure 5*.

I can be observed that the medium is occupied by the Station A, while Station B freezes its backoff until the end of the transmission. After the reception of the ACK and an idle period of DIFS, Station's B backoff counter will be reactivated and resumed from the point it was stopped (Remaining Backoff).

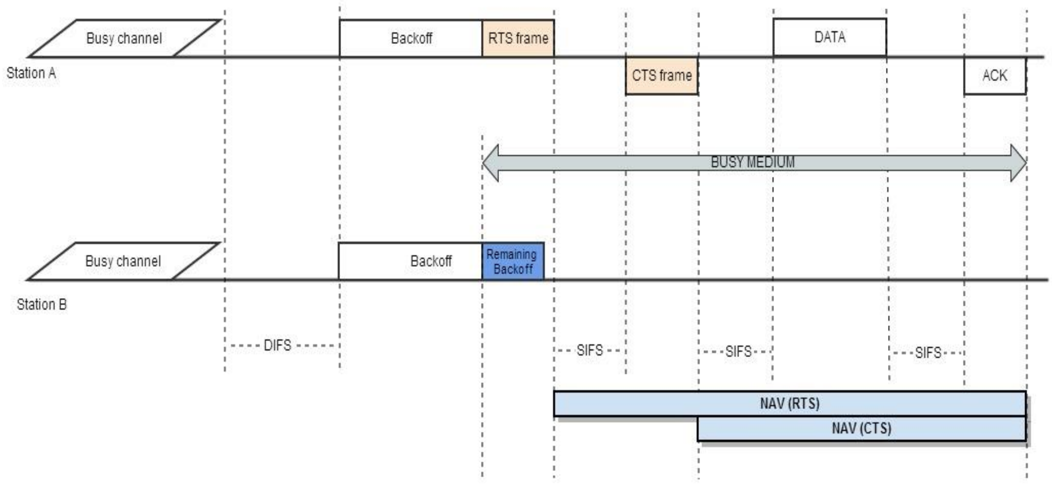

Figure 5 : Example of RTS/CTS access technique.

2.2.1.2 Point Coordination Function (PCF)

Point Coordination Function (PCF) is the second of the original operation algorithms of IEEE 802.11. It is a centralized, polling-based mechanism which requires the presence of a coordinator, such as an Access Point. Hence, it can be used only in Infrastructure BSSs, where APs exist. The PCF is located directly above the DCF and when it is required to be used time is divided into superframes. Each superframe consists of two periods; a Contention Period (CP) and a Contention Free Period (CFP). DCF is used in the CP, while PCF in the CFP. A CFP starts only after a transmission of a beacon frame from the Point Coordinator (PC). In that beacon frame the NAV duration is set to its maximum value to prevent the other stations operating in DCF from transmitting data.

A new Interframe Space is presented in that mechanism; the PCF Interframe Space (PIFS). The PIFS is shorter than the DIFS and greater than SIFS. Hence the AP has more priority over the DIFS stations. The duration of the PIFS can be calculated as follows:

$$PIFS = SIFS + Time\ Slot. \tag{11}$$

When the PC accesses the medium, keeps it until the end of the Contention Free Period and uses a Round-Robin algorithm to poll the stations. A station is not allowed to transmit if hasn't polled by the PC. The coordinator polls the stations with a CF-Poll, according to its polling list. When a node receives a polling frame, always responds to it with a CF-ACK and data. If a station has no data to send, it then responds with a CF-ACK and a null frame. Upon to a reception of a payload frame, the PC replies with a CF-ACK and polls the next node. A CF-ACK can be combined with a new CF-Poll and may also contain a payload frame.

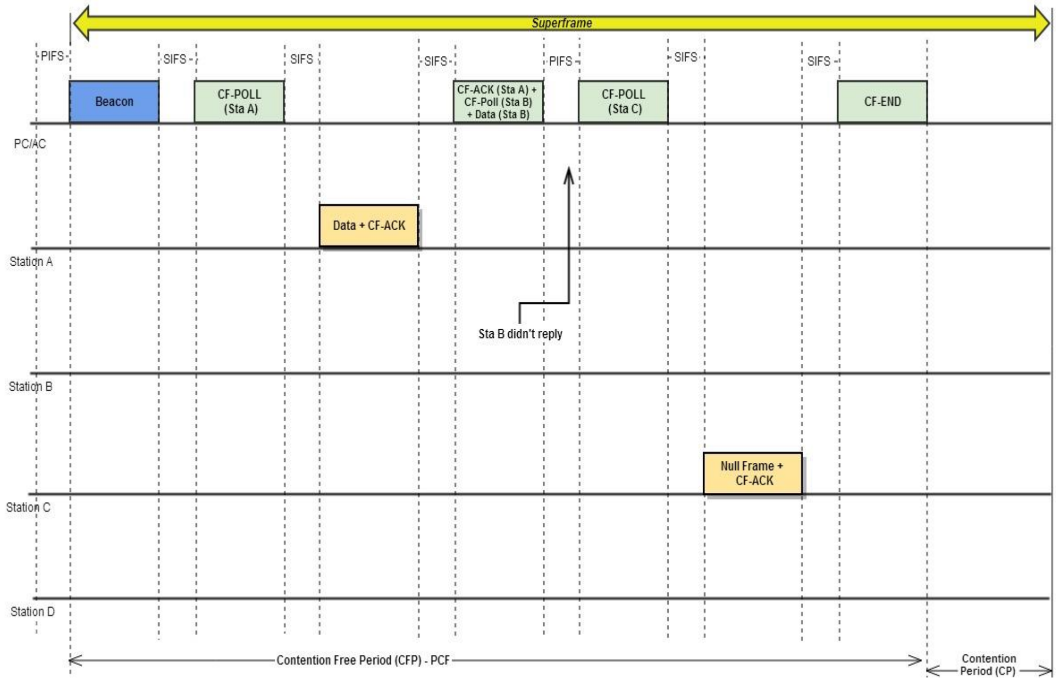

Figure 6 : Example of PCF mechanism in IEEE 802.11.

If a station does not answer to a CF-Poll, the PC waits for a PIFS period and then transmits a new CF-Poll. This is the maximum acceptable idle period in a CFP, to prevent DCF stations from granting the channel.

The CFP is terminated when the PC/AP transmits a CF-END in the end of that period, which was determined in the beacon frame. The NAV duration is reset and follows the Contention Period, where DCF is operated. To avoid fairness issues, between AP and STAs, at least one frame has to be sent during the CP.

A. Koepsel, J-P Ebert, and A. Wolisz in [7] showed that the PCF outperforms the DCF under high load traffic conditions, although two main drawbacks arise from the use of that mechanism. The first one occurs when a delay appears between the scheduled beacon frames. Beacons are transmitted in periodic intervals, which are set by the AP and stored in the beacons. Every node is informed for the next target beacon transmission time (TBTT). However, a node may occupy the channel at the TBTT, resulting in delaying the transmission of a beacon. This happens due to the contention mechanism of the DCF. The CFP is shortened and fewer nodes will be polled in this period. In the worst case scenario a beacon may be delayed by 5ms when the RTS/CTS mechanism is used in the slowest transmission scheme. The second drawback is that a polled station can transmit frames of any length (0 – 2304 bytes), without limited by the PC; causing unexpected delays. This may lead to some nodes not to be polled in that period and have to wait until the next CFP to be polled. *Figure 6* depicts an example of the PCF mechanism. Most of the cases described previously are illustrated in *Figure 6*.

In [8] an improved PCF mode is presented, where the AP dynamically decides the order of the nodes, based on the channel conditions. After the beacon's transmission, the PC polls all the stations requesting the size of data that each station contains. The following step is to calculate the order of the polling list based on the channel conditions of each node and the history of the previous polling list. It has been showed that the throughput is improved by 14% of the original PCF. However the protocol still suffers from a high time complexity and it is not suitable for real time applications; especially in dense networks.

2.2.1.3 Fragmentation mechanism

An optional feature for improving the performance when the wireless channel conditions are poor is fragmentation. A station uses fragmentation to divide the initial packet into smaller frames. These fragments are sent separately to the same destination, with a size according to a certain threshold. Each one of these frames includes a MAC and a PHY header increasing the overhead.

On the other hand, more benefits emerge when the size of a packet is large and the station operates in low transmission rate. Thus, a station occupies the medium for a shorter period of time for transmitting a fragment; improving the fairness among the users. Additionally, the probability of a collision decreases due to the transmission of smaller packets. Even so, if a collision or an erroneous fragment is received, only the specific fragment will be retransmitted and not the whole packet.

A station applies a fragmentation only in unicast packets and while that feature is enabled. While the packet is received in the MAC from the upper layers, is fragmented according to a threshold. The minimum threshold for the fragmentation is set to 256 bytes. Every fragment consists of a MAC, a PHY header and a Cyclic Redundancy Check (CRC) field (as seen in *Figure 7*). It also contains a Sequence Control Field which holds the sequence and the fragment number. The sequence number remains the same for all the fragments of a packet, while the fragment number starts at 0, for the first fragment and increases by 1 for each successful received fragment. When a retransmission is necessary, the fragment number does not change. The Frame Control Field which contains only one bit, informs the receiver for the existence of more fragments. When that bit (the More Fragments bit) is 0, indicates the last fragment or that no fragmentation is occurred. All fragments are sent as independent transmissions which mean that ACK are sent between fragments. The backoff process does not occur between the fragments, thus a station waits only for a period of SIFS before the next transmission.

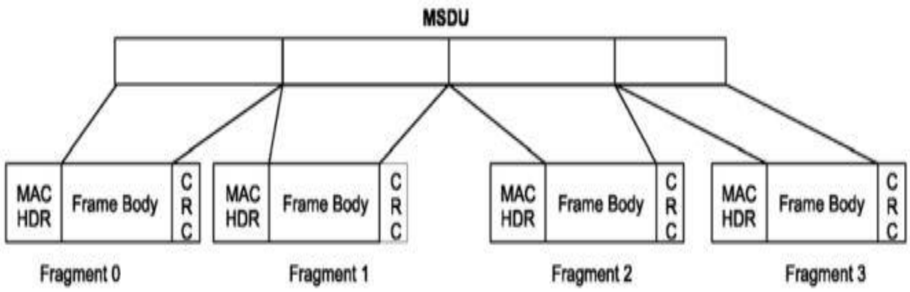

Figure 7 : Fragmentation of a MSDU [10].

The process of combining multiple fragments into a single MSDU (MAC Service Data Unit) is called defragmentation. Receiver uses the information that is held in header and reconstructs packet, by combining the fragments in order of the Sequence Control Field. If the fragment with the More Fragments bit equal to 0 has not yet been received, then the destination knows the MSDU is not complete. If all the fragments are received correctly, the single MSDU is forwarded to the higher layer. In both processes, there is a maximum time of sending / receiving a MSDU. If that time is exceeded, then the station discards the fragments from its queue.

2.2.2 QoS enhancements - Hybrid Coordination Function (HCF)

In 2005 a new amendment to the IEEE 802.11 standard was presented [9] for supporting QoS on the MAC layer, known as IEEE 802.11e. This coordination function called Hybrid Coordination Function (HCF) is only usable in QoS networks; except mesh topologies. The HCF combines functions from the DCF and PCF with some improvements; QoS mechanisms and frame subtypes to allow transmissions of QoS data in both CP and CFP. The coordination function that is used in a contention period is called Enhanced Distributed Channel Access (EDCA), while in a contention free period the HCF Controlled Channel Access (HCCA).

A new feature that is included in HCF is the Transmission Opportunity (TXOP) which is the time that a station is allowed to transmit frames. TXOPs are defined by a start time and a maximum length. A TXOP that is used during the CP is called EDCA TXOP, while during the CFP is called HCCA TXOP. Under a TXOP, a transmission cycle must end before the next TBTT, and its must not be longer than the TXOP duration. It is possible that no frames are transmitted during the TXOP.

2.2.2.1 Enhanced Distributed Channel Access (EDCA)

EDCA is a contention-based mechanism using 8 different User Priorities (UPs), providing differentiation among the QoS packets. The UPs are combined to four Access Categories (ACs) as shown in *Table III*. Each one of these entities has different parameters, operating similar to DCF and is known as Enhanced Distributed Channel Access Function (EDCAF).

Table III
UP-to-AC mappings [10].

Priority	User Priority (UP)	Access Category (AC)	Designation
	1	AC_BK	Background
Lowest	2	AC_BK	Background
↓	0	AC_BE	Best Effort
	3	AC_BE	Best Effort
Highest	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voice
	7	AC_VO	Voice

Instead of DIFS, a queue shall wait for an Arbitration Interframe Space (AIFS) to access the medium. This interval is defined by the value of AIFSN [AC] and varies for the four categories. The minimum value of an AIFSN [AC] is 2, which is equal to the DIFS value. The duration of AIFS is computed according to:

$$AIFS[AC] = SIFS + AIFSN[AC] * SlotTime. \tag{12}$$

The smaller the AIFS [AC] is, the higher the access priority becomes for a queue.

Another EDCAF parameter is the contention window from which the random backoff is computed and is randomly selected from [1, CW]. The limits of the CW are variable and assigned by an AP or the BSS Coordinator. Note that the backoff counter decreases by one slot before the end of the AIFS. As in DCF, the CW doubles in every retransmission until its maximum value (CW_{max}) or the retry limit.

Collisions may occur between the queues, which are handled in a virtual way. More precisely, the highest priority packet among the colliding packets is chosen and transmitted, while the others behave as an external collision which takes place in a DCF. Their CW is doubled and performs a random backoff. In an internal collision, unlike an external collision, the retry bit is not changed.

The following enhancement is very important, as it provides the opportunity to send multiple frames in a single TXOP. Note that during a TXOP period, only packets from the same AC are sent. Between the frames intermediate ACKs. A value of 0 corresponds to a transmission of a single MSDU, including the ACK from the receiver. It may need to fragment a packet if the transmission duration exceeds the TXOP limit. This usually occurs when a node operates in low PHY rates. If a retransmission occurs in a lower PHY rate, it is possible the TXOP to be exceeded. In this case, the node will not transmit more than one frame during the TXOP. Given the EDCA TXOP feature, the stations that operate in low transmission rates, do not grant the channel control for long time periods.

Several studies have done for adjusting TXOP according to some parameters. J. Majkowski, and F. C. Palacio in [11] presented an adaptive TXOP mechanism based on the average number of packets allocated in AP queues. The TXOP limit can be calculated, using the following equation:

$$TXOP_{Limit} = N * Pq\_Length. \tag{13}$$

Where N is the average number of packets in a queue, and Pq_Length is the average packet length; computed as the transmission time. Though the proposed algorithm is very accurate to estimate the N, it works better under saturation conditions. Only downlink stream is considered, from a QoS AP to nodes, making it the most congested node in the network. Thus, one main issue is derived from this work which related to fairness between downlink and uplink traffic.

To overcome the fairness issue between the uplink and downlink traffic, another adaptive algorithm was presented [12], named DTXOP. DTXOP adjust dynamically the TXOP for each AC, according to the traffic. In addition, when upstream demand is greater than downstream, reduces the AP's TXOP limit. Though the Dynamic TXOP improves the throughput, the fairness between the streams and increase the time when the system is saturated, it fails to meet QoS requirements for the real-time applications. As the number of nodes is enlarged, the average packet delay is increased dramatically; especially for the uplink traffic.

The EDCA parameters are defined in some Beacon frames, in all Probe Response and (Re) Association frames. If a station does not receive such frame, then uses the default parameters. *Figure 8* depicts the EDCA mechanism with four queues, as it is in a station.

Table IV
Contention Window boundaries in EDCA scheme.

Access Category (AC)	**CW_{min}**	**CW_{max}**
AC_BK	aCW_{min}	aCW_{max}
AC_BE	aCW_{min}	aCW_{max}
AC_VI	$\frac{aCW_{min}+1}{2}-1$	aCW_{min}
AC_VO	$\frac{aCW_{min}+1}{4}-1$	$\frac{aCW_{min}+1}{2}-1$

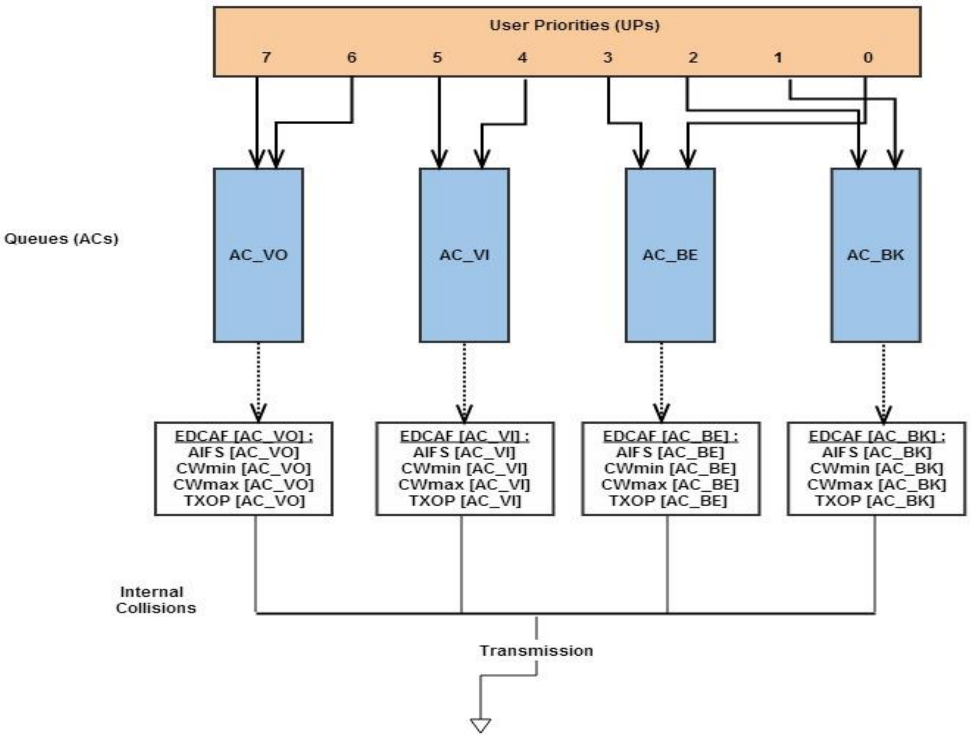

Figure 8 : EDCA mechanism according to IEEE 802.11e.

The Contention Window boundaries are defined for each physical layer and calculated according to the *Table IV*. *Table V* summarizes all the EDCA parameters with their default values as defined for IEEE 802.11a, and IEEE 802.11g (only g) standards.

Table V
Default values of the EDCA parameters for IEEE 802.11a/g.

AC	CW_{min}	CW_{max}	AIFSN	TxOP
AC_BK	15	1023	7	0
AC_BE	15	1023	3	0
AC_VI	7	15	2	3.008ms
AC_VO	3	7	2	1.504ms

2.2.2.2 HCF Controlled Chanel Access (HCCA)

The HCCA is the second access mechanism that was defined in IEEE 802.11e standard and considered of high complexity. It is consisted from a centralized coordinator (HC), just like PC in the PCF, but differs from this in two ways. The former is that HCs can poll STAs during both the CP and the CFP (see *Figure 9*). The latter is the transmission of multiple frames within a given polled TXOP.

Unlike the PC in the PCF, the HC waits only for a PIFS idle time to seize the medium, having a higher priority over the EDCA nodes. The period that a HC occupies the channel is called Controlled Access Phase (CAP). A CAP is initiated when the HC wants to send data and ends when the coordinator does not reclaim the channel after the end of a TXOP. A CFP starts only after a TBTT and when the Beacon frame has transmitted.

The Beacon frames include an additional element, the CF Parameter Set, whereby the stations set their NAVs. This feature prevents the non-polled stations from transmitting during the contention free periods. During the CFPs, a HC sets its own NAV too, when other stations transmit. If the HC has no more nodes to poll or data to send, a QoS CF-poll frame is transmitted, informing the stations to reset their NAVs. Moreover, a NAV is reset when a station receives a CF-End (+ ACK) frame.

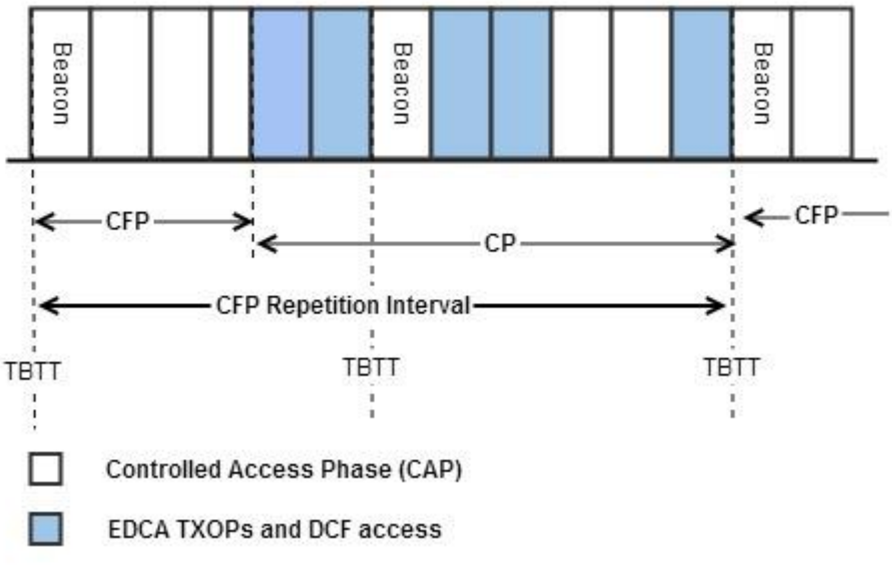

Figure 9 : CAP/CFP/CP periods ([10], figure 9-22).

Within a polled TXOP, the transmission of the frames must not exceed the TXOP limit. Furthermore, a TXOP should not last more than the TBTT. If a QoS Station has no traffic to send or the packets are too long to be transmitted within the TXOP and fragmentation cannot occur, a null frame is sent (QoS (+) Null frame). Thereby, the HC is informed about the node's condition (e.g. queue size) and adjusts the TXOP according to its requirements. However, the TXOP limit is strongly dependent on the PHY mode and IFS overhead. In the case of empty queues, the HC may reclaim the TXOP or if the transmission ends earlier than the TXOP limit and there is enough time left. An additional option is the piggybacked frames. This feature is allowed only during a CFP or polled-TXOP and is very useful for increasing the throughput (*Figure 10*).

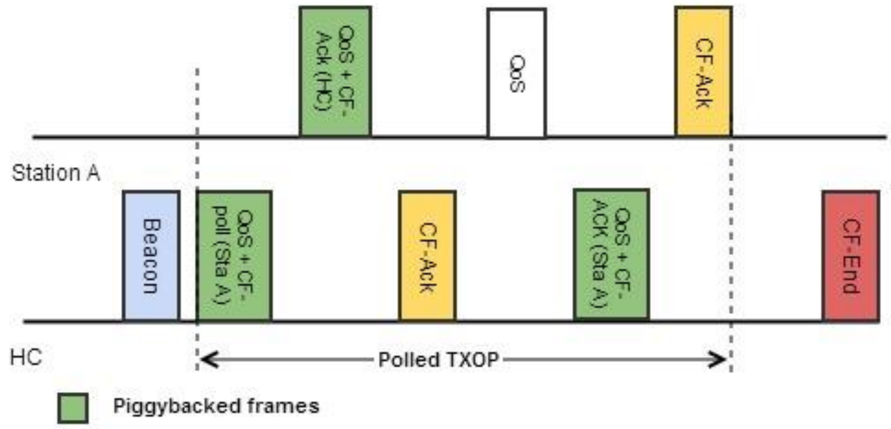

Figure 10 : A simple example of piggybacked frames within a Polled-TXOP (CFP is considered).

2.2.2.3 Block Acknowledgement (Block Ack)

The Block Ack mechanism was introduced in IEEE 802.11e standard to improve channel efficiency by aggregating multiple acknowledgments into one. It's an optional feature and there are two types of the Block Ack scheme: Immediate Block Ack and Delayed Block Ack. The first is suitable for any kind of traffic, while the second one is only suitable for non-real-time applications that are tolerable to delays.

This mechanism is used only when QoS is supported and the participants are capable to manage these frames. After that, an exchange of ADDBA Request/Response frames follows, between two nodes which negotiate the parameters that will be used. The originator can transmit blocks of MPDUs to the receiver only if the Block Ack agreement has been established. The maximum number of MPDUs that can be acknowledged in a single Block Ack is 64. Hence, the transmission window is not greater than 64.

All frames have a sequence control value that indicates the number of MPDU for this block. The recipient reassembles all completed frames from its buffer, with a preceding value of the current starting sequence control, and forwards them to its upper layer. The Block Ack contains a bitmap which holds the sequence numbers of the packets, as the status of these frames. If a packet was not received correctly, the originator retransmits as long as its lifetime limit has not been exceeded.

The Block Ack agreement is torn down with a DELBA frame, sending by the originator. Also, it is possible to be terminated after a Block Ack timeout. *Figure 11* illustrates the procedure of the Block Ack mechanism, including the three phases; setup, data transmission and tear down.

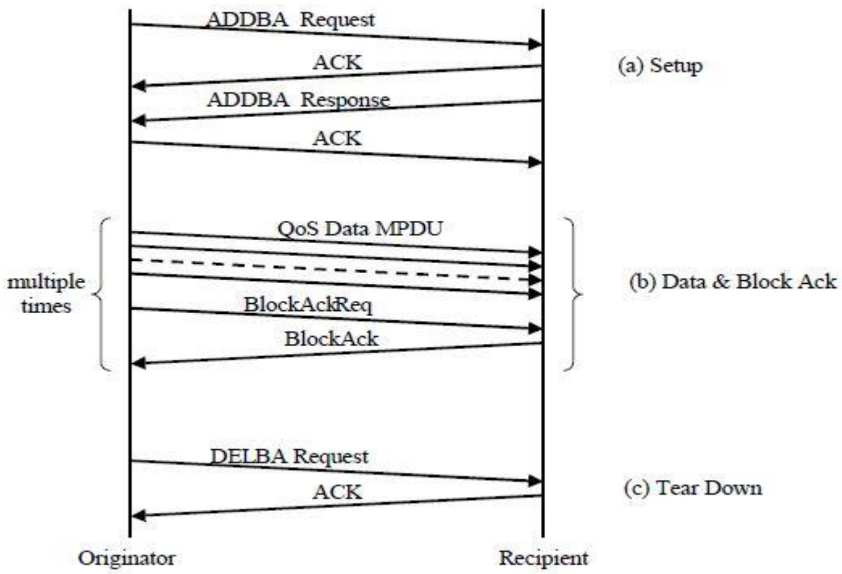

Figure 11 : Block Ack mechanism ([10], figure 9-25).

2.2.2.4 No Acknowledgement (No Ack)

This mechanism is determined by the policy at the QoS stations. It can be used in delay-intolerant networks and when the probability of the lost frames is low. On the other hand, when this policy is used, the reliability of the traffic is reduced and thus, a protective mechanism should be used (e.g. RTS/CTS) from other nodes.

2.2.2.5 Direct Link Protocol (DLP)

In legacy 802.11 WLANs all data frames are sent through and received from an AP. However, this protocol allows the stations to communicate directly. The DLP has to be established between the stations, in order to be able to communicate without requiring any intermediate APs.

2.2.3 Medium Access Control (MAC) enhancements (802.11n)

Although the IEEE 802.11e provides QoS, TXOP (burst mode) and the Block Ack scheme, the MAC efficiency remains quite low due to the overheads of that layer [13]. In 2009, the IEEE 802.11n amendment was released with a goal of achieving more than 100 Mbps at the MAC layer in a 20 MHz channel bandwidth. Four enhancements have introduced in order to reduce the overhead of that layer, with the aggregation mechanism to be the most important.

In the following sub clauses the MAC enhancements are described which are introduced in 802.11n.

2.2.3.1 Aggregation

The aggregation mechanism combines multiple data packets from the upper layer into a single frame. Therefore, the MAC overhead is reduced by using the same MAC/PHY header, preamble and avoiding the wasted time of IFS and backoff procedure.

Two aggregation schemes are defined in the standard, known as Aggregate MAC Service Data Unit (A-MSDU), and Aggregate MAC Protocol Data Unit (A-MPDU). Also, a third scheme is applicable by combining the A-MSDU and A-MPDU, known as Two-Level aggregation. A performance evaluation of the aggregation mechanisms form many aspects, is shown in [13], [14].

2.2.3.1.1 A-MSDU scheme

MSDU aggregation performs on the higher MAC layer and concatenates multiple MSDUs into one frame. All the MSDUs must be from the same Access Category and destined to the same receiver. Its maximum length can be either 3839 or 7935 bytes; 256 bytes shorter than the maximum PSDU length. As soon as this length or the waiting time of the packets (delay) reaches its maximum value, the A-MSDU is created and forwarded to the lower layer.

An A-MSDU frame consists of multiple subframes; each one of these contains a header, a payload, and extra padding bytes. The subframe header includes three fields; the Destination Address, the Source Address, and the Length of the MSDU. The size of the MSDU in each subframe may be different, with a maximum length of 2304 bytes. The last field, the padding field, adds 0-3 bytes in each subframe, except for the last one. The number of the extra bytes depends on the rule that every subframe should be a multiple of four bytes, so the receiver to be able to calculate the beginning of the next subframe.

After the aggregation is completed, only one MAC header, and a FCS field will be added in the packet. The entire A-MSDU is retransmitted if any subframe is corrupted, as only a single FCS is used. This can greatly affect the performance under error-prone channels. *Figure 12* describes a simple structure of an A-MSDU scheme. Summarizing, the following rules dictate the A-MSDU mechanism:

- All MSDUs belong to the same AC and have the same TID.
- All MSDUs have the same Destination Address (DA) and Source Address (SA).

The DA and SA match to the same Receiver Address (RA) and Transmitter Address (TA) in the MAC header. Thus, only unicast packets are allowed. An A-MSDU expires only when the lifetime of all constituent MSDUs have expired.

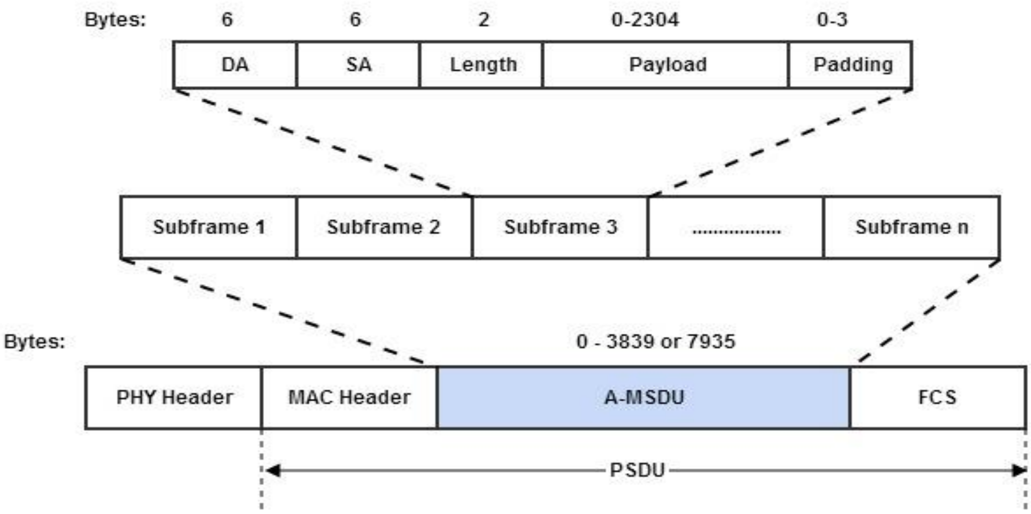

Figure 12 : A-MSDU scheme.

J.Deng, and M. Davis in [15] proposed an adaptive aggregation algorithm (AAM) improving the throughput by taking into account the packet delay and the traffic conditions. The AAM calculates the number of aggregated frames, with respect to the previous aggregate frame. If the buffer has insufficient packets for selection, the algorithm waits for more packets to arrive. It selects the N smallest packets when the maximum delay time is exceeded, and performs aggregation according to the target aggregation threshold (it is determined by the user). However, it is not considered the QoS mechanisms, like the TXOP and ACs or is not examined in multirate system. Also, it is compared with a FIFO aggregation algorithm that is based on fixed number of aggregate frames and not on TXOP.

A model to predict network throughput is presented in [16]. Also, a new A-MSDU adaptation algorithm, under error-prone channels is studied. It is shown that the optimal aggregated frame is more sensitive to BER, rather than the number of the stations. Due to that a feedback mechanism should be used, providing the node with the channel SNR information. Before a transmission, the sender receives an estimation of the BER and then aggregates the frames according to that. Though, in dense networks the average delay may increase, due to the extra overhead of the feedback mechanism, which is very critical for the real time applications. Also, the transmitter should receive the SNR information faster than the channel changes. Another drawback is the performance in QoS networks, as it is not studied here.

2.3.1.2 A-MPDU scheme

Aggregation of MPDUs happens at the bottom of the MAC, unlike A-MSDU. It groups multiple frames into a single one, with a maximum size of 65535 bytes. Also, it can hold maximum 64 subframes, due to block ack bitmap limitations. MAC headers are attached in each MPDU frame, as this procedure occurs in the higher MAC. An A-MPDU frame can contain multiple MPDU frames with different TID (Traffic ID, following the TXOP rules) and destination address. Only an HT AP and an HT mesh station can transmit A-MPDU containing MPDUs with a group address, following the rules that described in [10] (sub clause 9.12.4).

Each subframe consists of three fields, a delimiter (MPDU delimiter), a MAC PDU and padding. A structure of an A-MPDU is presented in *Figure 13* . A MPDU delimiter defines the MPDU position and length inside the aggregate packet. Its size is 4 bytes and contains 4 sub fields as shown in that figure. Following the delimiter field is the MPDU with variable size. All MPDUs are limited to 4095 bytes as the length of a PPDU cannot exceed the 5.46 ms time limit. As in A-MSDU, padding bytes are appended to make each A-MPDU subframe, except the last one, a multiple of four bytes.

Note that every MPDU has its own MAC header, while a single PHY header is used for the entire A-MPDU frame. As a result, MPDU aggregation adds more overhead over A-MSDU. However, outperforms over MSDU aggregation in error-prone channels, due to FCS field that every MPDU contains. So, if a subframe is corrupted then only this MPDU is retransmitted and not the entire aggregated frame.

An analytical study of frame aggregation mechanism, under error-prone channels, is presented in [17]. As the Frame Error Rate (FES) increases, the use of large frames degrades the network's throughput. In this case the use of smaller MPDU subframes can be more beneficial; here lies the effectiveness of the proposed aggregation algorithm. Its MPDU subframe size is dynamically adjusted according to FER. In error-prone channels smaller frame aggregation size is used in order to meet the QoS requirements. However, it’s not defined how a frame size changes. Until now, two mechanisms exist in order to modify the MSDU length; fragmentation and A-MSDU, neither of these are mentioned.

A similar work is presented in [18], [19]. In [18] the authors proposed an adaptive aggregation scheme, based on QoS requirements of each AC. The applications which are insensitive to delay wait in the queues until the target aggregation threshold is reached. On the other hand, voice and video packets are immediately transmitted. For the voice frames only MSDU aggregation is used, while for the video frames A-MPDU. Performance of this scheme, only in lightly loaded conditions, is presented in [19]. Results showed that this scheme improves the throughput of the QoS sensitive applications. However, in error-prone or heavy load channels, this scheme may suffer from the use of A-MSDU for voice packets. Also, it may lead to starvation of the lowest ACs.

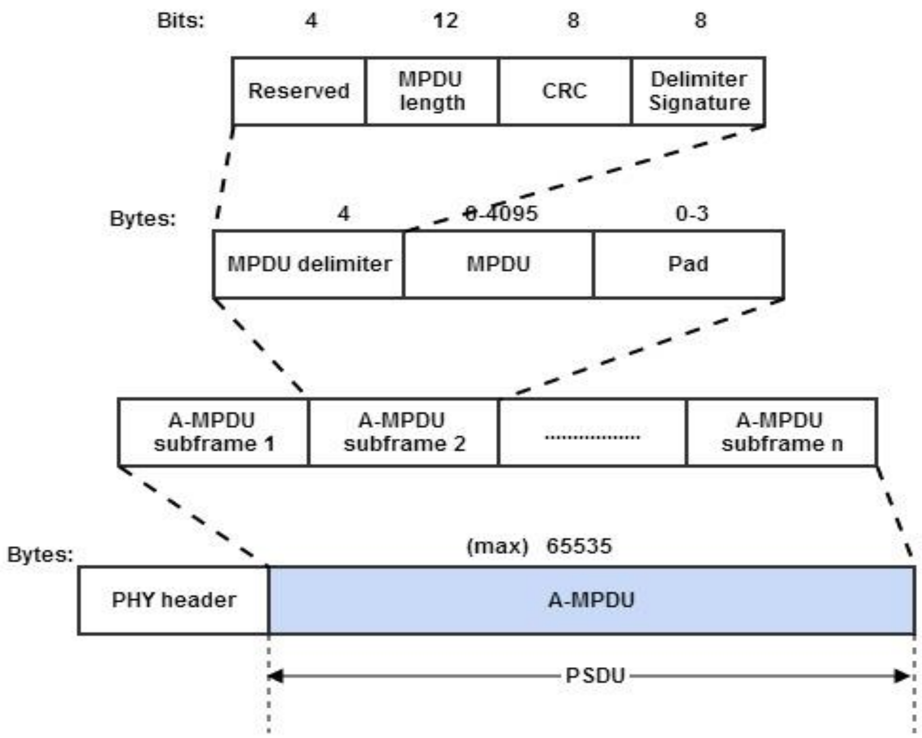

Figure 13 : A-MPDU scheme.

In order to improve fairness among the ACs and avoid starvation of Background and Best-Effort, a Smart Aggregation Mechanism is proposed [20]. When an AC gains access to the medium, aggregates that frame with others, belong in higher priority classes. Aggregated frames are consisted of packets with different TID. Although this scheme improves fairness and throughput for all ACs, different TxOP is used for every priority class which may lead to fairness issues among users.

Authors in [21] proposed an adaptive Contention Window to improve fairness among users. CW and frame aggregation size are customized according to the number of stations. CW is increased for low-rate nodes, while decreased for the high-rate. On the other hand, the number of aggregated frames is raised for high-rate station, and reduced for the low-rate. Note that time-fairness is assured due to relation between CW and number of aggregated frames. Even though this mechanism provides fairness for the users, it does not for the different ACs.

2.3.1.3 Two-Level aggregation scheme

A third aggregation mechanism is also available by combining the previous two, known as Two-Level aggregation scheme. In the first stage, all MSDUs are aggregated according to rules of A-MSDU mechanism.MAC header and FCS are attached in each of them. Next, every A-MSDU frame is concatenated with other MPDUs, while delimiter and padding are added (see *Figure 14*). As in A-MPDU, TIDs can be different in MPDUs. Also, MPDUs are limited to 4095 bytes, while the entire A-MPDU is up to 65535 bytes.

All types of mechanism improve MAC throughput, with Two-Level scheme outperforms over A-MPDU and A-MSDU [14], [22]. Due to first level aggregation mechanism, larger packets are aggregated in Two-Level mechanism comparing to others. However, that aggregation scheme is more complex and can increase the overall delays, especially in error-prone or dense networks.

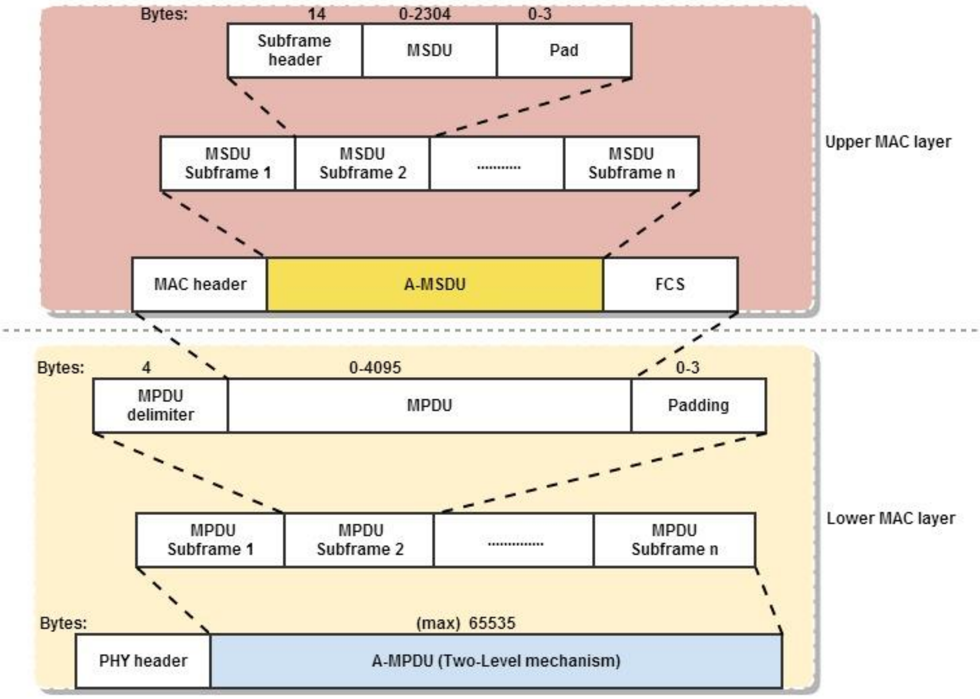

Figure 14 : Two-Level aggregation scheme.

In order to avoid additional dummy MPDU delimiters, an adaptive Two-Level aggregation algorithm proposed [23]. In general, A-MSDU is more efficient than A-MPDU for small frames, due to MAC overhead. Here, lies the operation of the proposed algorithm; when the upper MAC detects that the size of a frame is less than a threshold (L_{min}), it performs aggregation in order to make it larger than L_{min}. Throughput for small packets is increased, while for larger frames complexity is avoided. Though, extra information is needed to define threshold in various channels.

2.2.3.2 Reverse Direction Protocol (RDP)

Reverse Direction is an optional mechanism used to reduce MAC overhead and improve QoS. Due to its bi-directional nature, it increases the efficiency of networks, especially on VoIP or TCP traffic flows. During TXOP, receiver grants permission of sending frames to transmitter. These nodes are known as RD initiator and RD responder, respectively.

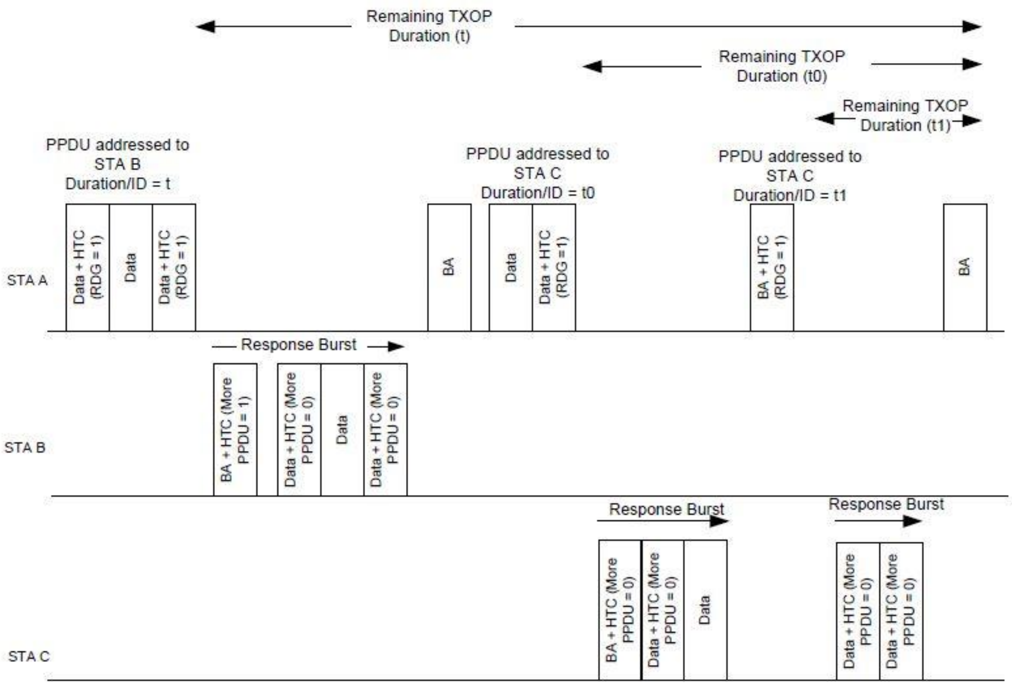

Figure 15 : Example of the RD exchange sequence as presented in [10].

A reverse direction transmission starts, when the initiator includes an RD-grant in one of its frames. After receiving RD-grant, the responder can use it, sending Block Ack with data frames. There isn't any constraint for the frames' TID, unless AC Constraint subfield is set to 1. RD initiator can transmit its next PPDU after SIFS period of received frames, or if the medium is idle for PIFS. On the other hand, RD responder frames are separated by SIFS or

RIFS. This procedure is repeated several times, until the end of TXOP. Also, it is possible multiple nodes to access the channel during a TXOP, by granting permission of the RD initiator. An example of an RD exchange sequence is given in *Figure 15*.

2.2.3.3 Block Acknowledgement – High Throughput (HT)

Block ack mechanism was originally defined in 802.11e, with two flavors: immediate and delayed block ack. Both schemes are enhanced in 802.11n amendment to improve even more MAC efficiency. These original mechanisms were modified slightly to support HT stations: HT-immediate and HT-delayed block ack. All four schemes are supported by HT-stations, although the original flavors would only be used for interoperability with legacy stations.

The distinction between the original and the HT block ack schemes is memory decrease for storing the block ack scoreboard. As fragmentation does not provide any benefits at higher data rates, a 64-bit scoreboard (Compressed block ack) can be used to support 64 MSDUs / MPDUs. This reduces both overhead and memory requirements in the recipient. On the other hand, HT-delayed block ack differs from its initial mechanism in the manner in which BAR and BA frames are acknowledged. These modules are not in the scope of this thesis and only mentioned[1].

2.2.3.4 Reduced Interframe Space (RIFS)

A new interframe space introduced in 802.11n, reducing overhead and increasing network efficiency. This reduced IFS (RIFS) is used in bursting mode to separate frames. RIFS is much shorter than SIFS, with a value of 2μs. RIFS bursting is not always permitted, as most of legacy devices can't detect a second frame arriving less than SIFS after the first frame. An AP can prevent RIFS bursting by setting the RIFS Mode subfield to 0, otherwise RIFS is used.

[1] More information about block ack mechanisms is provided in [10], [24].

2.3 Physical (PHY) Layer

IEEE 802.11 standard specifies different PHY layers for data transmissions over different spectrum bands. The original standard in 1997, defined three techniques on PHY for the 2.4 GHz Industrial. Scientific and Medical (ISM) radio band. Infrared radiation (IR) was the first technology that used infrared signals. It relies on optical signals in 800-900 nm band, supporting data rates of 1 and 2 Mbps. This technology was not as successful as the next two; frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).

FHSS divides frequency spectrum into channels. Packets are split up and transmitted on these channels, in a random pattern, known only to transmitter and receiver. On the other hand, DSSS spreads data across a wider frequency band according to a mathematical key. This key is known to receiver for decoding the data. FHSS technology works best for small packets in high interference environments, while DSSS for large packets in low interference channels.

Two years later, in 1999, a new technology was approved by IEEE, the 802.11a amendment. It uses orthogonal frequency division multiplexing (OFDM), supporting data rates up to 54 Mbps in 5 GHz band. In early 2000, 802.11b standard was ratified operating in the 2.4 GHz frequency band and supporting rates up to 11 Mbps. Unlike 802.11a, it uses DSSS technology and complementary code keying (CCK) as its modulation technique. Due to desire of higher data rates in the 2.4 GHz, a third standard was introduced in 2003. 802.11g uses OFDM technology with a maximum data rate of 54 Mbps, and is fully compatible with 802.11b devices.

Inefficiency of IEEE 802.11a/b/g standards in terms of throughput and coverage, led to many enhancements in PHY and MAC layer. IEEE 802.11n published in 2009, introducing new technologies and approaches in these layers. Use of MIMO technology and wider channels boosted the maximum data rate from 54 to 600 Mbps. It operates on unlicensed bands (2.4, and 5 GHz), using Orthogonal Frequency Division Multiplexing (OFDM) and it provides compatibility with legacy devices. Moreover, the IEEE 802.11ac-2013 enhances the amendment n, supporting higher raw rates and Multi-user MIMO technology. Although these standards provide higher data rates, the requirement for new, available spectrum led to the necessary exploitation of higher frequency bands. The IEEE 802.11ad operates in the 60 GHz millimeter-wave spectrum, while IEEE 802.11af in TV white space spectrum (54-790 MHz).

The newest and emerging standards add functionality to existing WLAN technology, overcoming many issues of the current protocols. Among these upcoming standards are 802.11ah, and 802.11ax. The former is anticipated to be published by the end of March 2016. The former aims to improve spectrum efficiency, and area throughput at 2.4 and 5 GHz bands. IEEE 802.11ax is a successor of 802.11ac and in very early stage, as HEW SG announced the approval of this project in March 2014.

2.3.1 Orthogonal Frequency Division Multiplexing (OFDM)

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier technique, where multiple symbols are transmitted simultaneously over a medium. Channel is divided into a number of lower-rate parallel sub-channels, increasing robustness against interference and frequency-selective fading.

In mobile wireless channels, reflection, scattering, and diffraction are factors affecting the received signal strength. All these factors result in multiple versions of transmitted signal that arrive at the receiving end, displaced with respect to one another in time and spatial orientation. This effect is called multipath. . Sometimes, a direct Line-of-Sight (LOS) (dominant stationary; nonfading) signal component exists between the transmitters and the receiver. As mentioned before, the reflected waves arrive at the receiver at different times, causing inter-symbol-interference (ISI), which may lead to a fail transmission and degradation of the system performance. Some of the techniques proposed in the literature, aiming in mitigating ISI are reducing symbol rate and using adaptive equalization techniques or higher transmission rates. However, the proposed techniques pose some important drawbacks as well including degradation in data rate, difficulties in operating of adaptive equalization at such high rates and delays greater than one symbol time, respectively.

To overcome ISI and the drawbacks of previous techniques, a new technique was developed in the mid-1960s [25]. In a multi-carrier system, the total bandwidth is divided into *N* non-overlapping frequency sub-channels (Frequency Division Multiplexing, FDM). Each sub-band carries a single-separate symbol and is independent from the others. Therefore, only a small percentage of these sub-carriers will be affected by multipath, and not the entire packet. To assure an Inter-Block-Interference (IBI) and Inter-Carrier-Interference (ICI) free transmission, guard bands are added between the sub-channels which can either by a cyclic prefix (CP) or zero padding (ZP). . However, in either case (CP or ZP) the addition of guard bands leads to waste of available bandwidth.

Unlike Frequency Division Multiplexing (FDM), OFDM uses orthogonal overlapped sub-channels and it is therefore more bandwidth efficient (*Figure 16*). Orthogonality prevents interference between sub-carriers, indicating that there is a precise mathematical relationship between carriers' frequencies. As in FDM, a Guard Interval is defined to maintain orthogonality over dispersive channels.

Symbol rate is a multiple of *1/T*, where *T* is a symbol period, and reduced by a factor of *N* (*N*: number of sub-channels). Hence, the symbol internal in an OFDM system is $N*T_s$, where T_s is the symbol interval in a single-carrier system. By selecting N to be sufficiently large, ISI is minimized and each sub-band seems to have a fixed frequency respond.

Robustness against ISI and frequency selective fading due to cyclic prefix (CP) and spectral efficiency because of orthogonality are some of the major advantages that OFDM offers. Furthermore, OFDM achieves higher data rates with lower bit rates than conventional communication systems. However part of the price one pays for the low-complexity equalization in CP-OFDM is the high Peak-to-Average-Power-Ratio (PAPR), complexity due to generation of orthogonal sub-carriers and power inefficiency as increases Peak-to Average-Power-Ratio (PARP).

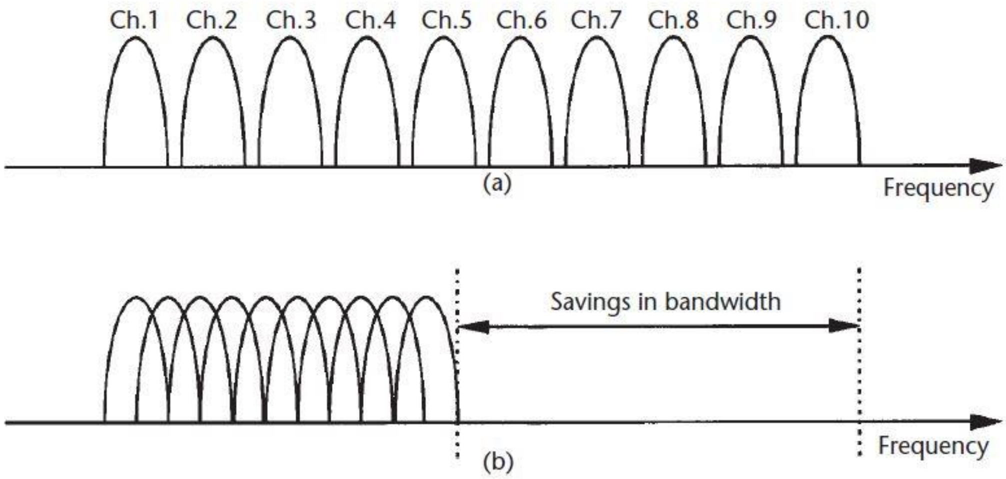

Figure 16 : (a) Regular multi-carrier technique (FDM), and (b) Orthogonal multi-carrier technique (OFDM) [26].

OFDM is a well-known; thoroughly studied and widely applied multicarrier system. It is widely used in wireless communications, and it is the primary technology in IEEE 802.11. It was firstly introduced in 802.11a, providing higher data rates than its predecessors. The used frequency range is split up into 64 subcarriers, from which only 48 are used for data and 4 are pilot carriers. Each one of the subcarriers is modulated using BPSK, QPSK, 16-QAM, or 64-QAM and carries 1, 2, 4, or 6 bits respectively. The major parameters of the OFDM 802.11a PHY are listed in *Table VI.*

Figure 17 presents a simple block diagram of an OFDM system. Modulated symbols undergo a serial-to-parallel conversion, and then are mapped into an IFFT processor. The output of IFFT is an OFDM symbol, consisting of N_{FFT} samples. After parallel-to-serial conversion, an empty guard interval or cyclic prefix is added in each IFDM symbol.

OFDM is well suited to wideband systems, which suffer from a deep fade or narrow band interference. In addition, it is bandwidth efficient and tolerant of time synchronization errors, making it a promising candidate for high speed networks. Given these advantages, 802.11n physical layer builds upon OFDM technology.

Table VI
OFDM parameters for 802.11a (20 MHz channel).

Data Rates for 20MHz channels (Mbps)	Modulation	Coding Rate	Coded bits per subcarrier	Coded bits per OFDM symbol	Data bits per OFDM symbol
6	BPSK	$1/2$	1	48	24
9	BPSK	$3/4$	1	48	36
12	QPSK	$1/2$	2	96	48
18	QPSK	$3/4$	2	96	72
24	16-QAM	$1/2$	4	192	96
36	16-QAM	$3/4$	4	192	144
48	64-QAM	$2/3$	6	288	192
54	64-QAM	$3/4$	6	288	216

One of the enhancements of 802.11n targeting on increasing the data rates was made in the existing OFDM by supporting shorter guard interval. In 802.11a, guard interval is 800 ns, while in 802.11n an optional guard interval of 400 ns is supported too. Shorter interval between symbols increases the throughput by almost 11%. However, there is a tradeoff between guard intervals (GI) and ISI, as the latter increases for short intervals. Guard interval of 400 ns can be used only between HT-nodes and need to be larger than the channel delay spread to achieve flat varying channel.

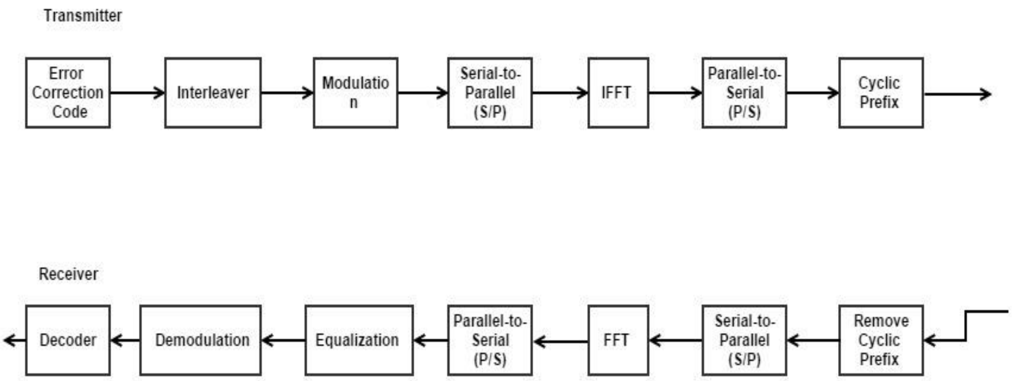

Figure 17 : OFDM block diagram.

2.3.2 Multiple Input Multiple Output (MIMO)

Multiple-Input-Multiple-Output (MIMO) is a technology that uses multiple antennas at both the transmitter and receiver (*Figure 18*). It takes into advantage the multipath propagation, allowing the receive antennas to separate out the signals, coming from different paths. The more obstacles (reflectors, scatterers etc.) exist in an environment, the richer the multipath is and hence the more beneficial it is for the MIMO wireless communication system.

Transmitting or receiving a signal from two or more antennas provides additional degrees of freedom and increases the capacity. The capacity of a MIMO channel is equivalent to the number of transmit/receive antennas When multiple spatial streams are transmitted from different antennas (Spatial Division Multiplexing - SDM), data rate is proportional to min (T_x, R_x, data streams); where T_x is the number of antennas in transmitter, R_x the number of antennas in receiver, and data streams the independent-spatial streams.

Opportunistic communication is another technique that exploits channel fading. When the channel is good, nodes transmit at high rates, otherwise at low rates, providing power gain. Multiple antennas only, at receiver (SIMO) or transmitter (MISO) provide power gain too. On the other hand, MIMO mechanisms can give both power gain and additional degrees of freedom.

In ideal conditions, Shannon capacity of a SISO channel depends on bandwidth (B), signal power (S) and noise power (N).

$$C = B * log_2\left(1 + \frac{S}{N}\right). \quad (14)$$

The channel capacity may be improved by increasing the bandwidth or transmission power, or by reducing the noise power. . However, in wireless systems there are limitations on available spectrum and power. Also, noise relies on many environmental parameters that can change during a transmission. The simple mathematical model that describes a SISO communication channel model is defined by:

$$Y = Hs + w, \quad (15)$$

where *H* is the channel impulse response, *s* the transmitted signal ,*w* the Additive White Gaussian Noise (AWGN) and *Y* is the received signal.

As it was mentioned previously, multiple antennas on a node improve system performance, in terms of reliability or capacity [27]. Diversity techniques intend to receive/ transmit the same signal from multiple antennas, improving system reliability; especially in deep fade channels. On the other hand, when independent data streams are transmitted, simultaneously, from two or more antennas, higher raw throughput is achieved.

There are several ways of providing diversity gain:

- Spatial diversity, multiple antennas that are separated.
- Polarization diversity, combination of multiple waves with different polarization.
- Time diversity, where copies of signal are transmitted in different times.
- Frequency diversity, same signal is transmitted in different frequency bands.
- Angle diversity, multiple antennas on receiver with different directivity.

By repeating the same symbol over L different antennas during L symbol times, we get diversity gain of L but lose on degrees of freedom. In 1998, a simple transmit diversity method was proposed by S. Alamouti [28]. The Alamouti scheme is designed for 1 receive and 2 transmits antennas (MISO) or MIMO 2x2. In a 2x1 system, two OFDM complex symbols are transmitted during 2 symbol times and channel is described as follows [29]:

$$\begin{bmatrix} y[1] \\ y[2]^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} w[1] \\ w[2]^* \end{bmatrix}. \tag{16}$$

Where $y[\,]$ is the received signal, h_i is the channel impulse response, u_i the transmitted signal, and $w[\,]$ the Additive White Gaussian Noise (AWGN).

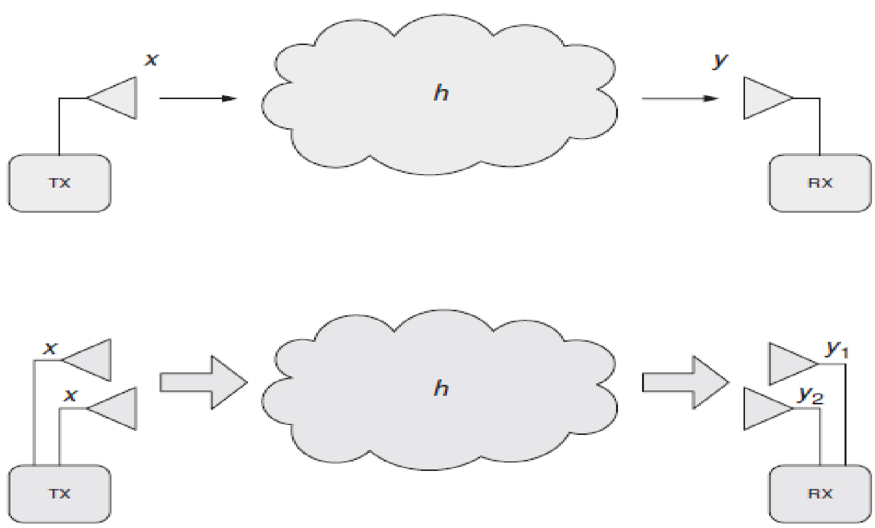

Figure 18 : SISO, and MIMO System [24].

Alamouti's scheme achieves diversity gain of 2 instead of 1 that repetition code provides. Furthermore, it uses half the power to transmit each symbol and utilizes all degrees of freedom in comparison to the repetition code. However, in 2x2 channels, neither repetition code nor Alamouti's scheme utilize all degrees of freedom. Another architecture that is used in transmit antennas is V-BLAST [30], improving network capacity by sending different symbols from each antenna during a symbol time. However, V-BLAST doesn't utilize diversity gain, as Alamouti's scheme does.

So far some of the methods that are used in MIMO and MISO channels have been introduces; taking advantage of the use of multiple antennas in transmitter. On the other side, a receiver should be able to combine all received signals and exploit multipath propagation. In the following section, some of these techniques are briefly described:

- Equal-gain combining, all received signals are weighted with the same factor, irrespective of signal amplitude.
- Maximal-ratio combining, all received signals are weighted according to their SNR.
- Switched combining, where receiver changes to another signal only if the currently signal's SNR drops below a threshold.
- Selection combining, receiver selects antenna with the highest received power.

Often, some of these methods may be combined to achieve better gain. Note that Maximal Ratio Combining (MRC) method in a 1x2 medium achieves better Bit-Error-Rate (BER) than using Alamouti's scheme in a 2x1 channel. Zero Forcing (ZF) equalizer or Minimum Mean Square Error (MMSE) can be applied in order to reduce interference and improve capacity in a MIMO system. In most of the cases, MMSE outperforms Zero Forcing; however MMSR is more complex than ZF.

MIMO technology is one of the new features introduced in Wi-Fi, with 802.11n. This standard supports up to four antennas at the transmitter, the receiver, or at both (4x4 MIMO). At receiver can be used Zero Forcing, MMSE, or other techniques that are compatible with the transmit (tx) signal. Also, it supports space-time block coding methods based on Alamouti's scheme and is compatible with V-BLAST. In addition, beamforming is used in 802.11n, enabling transmitters to send multiple streams over the same channel (eigenbeamforming).

As it is shown on *Table VII* MIMO technology improves data rate by a factor of 4, relatively to other PHY features that were introduced in 802.11n amendment.

Table VII
Improvement of data rates due to MIMO (20MHz channel).

MCS index	Spatial Streams	Coding rate	Modulation	Data rate (Mbps)	
				800ns GI	400ns GI
0	1	$1/2$	BPSK	6.5	7.2
1	1	$1/2$	QPSK	13	14.4
2	1	$3/4$	QPSK	19.5	21.7
3	1	$1/2$	16-QAM	26	28.9
4	1	$3/4$	16-QAM	39	43.3
5	1	$2/3$	64-QAM	52	57.8
6	1	$3/4$	64-QAM	58.5	65
7	1	$5/6$	64-QAM	65	72.2
. . .					
31	4	$5/6$	64-QAM	260	288.8

2.3.3 Channel Bonding

Transmission over wider channels is allowed in 802.11n. Doubling channel bandwidth to 40 MHz, increases raw data rate. However, the use of such channels makes the system more susceptible to interference and decreases transmission range. Due to that, channel bonding leads to degradation in network performance, especially in legacy SISO 802.11a/b/g systems. Moreover, at 2.4 GHz band doubling bandwidth causes more harm than benefits, as only 3 non-overlapping channels exist in this band.

An extensive study of channel bonding in WLANs at 5 GHz band was presented by L. Deek et al. [31]. Many experiments have been conducted in different environments to understand which parameters influence the network performance when channel bonding is used. It has been shown that RSSI, MCS, strength of interfering transmissions, and physical rates of links in CS range affect system performance, especially in a 40 MHz channel. Transmitter must have knowledge of this information before it increases the channel width, in order to maximize the throughput. Furthermore, the experiments demonstrate that it is more advantageous for a node to compete with an interferer who transmits at 40 MHz, due to the use of higher data rates, improving fairness among nodes. Nonetheless, frame aggregation was deactivated in most of the experiments, leading to lower rates. Also, different observations might be derived when MAC aggregation is used.

In *Table VIII* we present the maximum raw data rates that can be achieved in 802.11n WiFi, when using all the above PHY layer enhancements.

Table VIII
Raw rates in 802.11n.

MCS index	Spatial Streams	Modulation	Coding rate	Data Rate (Mbps)			
				20 MHz channel		40 MHz channel	
				800ns GI	400ns GI	800ns GI	400ns GI
0	1	BPSK	$1/2$	6.5	7.2	13.5	15
1	1	QPSK	$1/2$	13	14.4	27	30
2	1	QPSK	$3/4$	19.5	21.7	40.5	45
3	1	16-QAM	$1/2$	26	28.9	54	60
4	1	16-QAM	$3/4$	39	43.3	81	90
5	1	64-QAM	$2/3$	52	57.8	108	120
6	1	64-QAM	$3/4$	58.5	65	121.5	135
7	1	64-QAM	$5/6$	65	72.2	135	150
⋮							
31	4	64-QAM	$5/6$	260	288.8	540	600

Chapter 3

Software Tool

3.1 Network Simulator 3 (NS-3)

NS-3 simulator is a discrete-event network simulator for networking research and education. Its development began in 2006, but it was released in June 2008. Since its first release, ns-3 has been extended and it has been actively developed, as new features and modules are continuously added. .

The core of ns-3 and all models are written in C++, and users can work with C++ and/or Python software development tools. Even though both ns-3 and ns-2 are written in C++, ns-3 doesn't support ns-2 APIs. It's totally a new simulator with a new architecture, despite that some models have been adopted from ns-2.

One major advantage of this simulator is that it uses parallel and distributed simulation technology, utilizing the memory and allowing simulations of large networks. Ns-3 consists of many modules, some of the most important are:

- Ad hoc On-demand Distance Vector (AODV routing protocol)
- Applications
- CSMA
- Flow-Monitor
- Internet
- LTE
- Mobility
- Network
- Optimized Link State Protocol (OLSR routing protocol)
- Propagation
- Spectrum
- WiFi
- WiMAX

Each one of the available modules includes many classes; some of those are mandatory for a simulation, while others are used to give specific features on the system. Below, some of the available classes are briefly presented:

- Node (network module), provides methods for managing representations of computing devices in simulations.
- Application, starts/stops at specific times in a simulation and is installed in a node. An application generates traffic with various rates and protocol types. Packets are sent through sockets to other destinations. Two subclasses are the UdpEchoClientApplication and UdpEchoServerApplication that used for a server/ client application using UDP protocol.

- Channel, models a medium such as wire (CsmaChannel), wireless (WifiChannel) etc. that is needed to transmit data.
- NetDevice, a Network Interface Card (NIC) installed in a node, in order to communicate nodes via a channel. CsmaNetDevice, WifiNetDevice, PointToPointNetDevice have different features and are used on different systems.

This thesis focuses on the IEEE 802.11 standard which is implements using the WiFi module. Its architecture is presented in the following section.

3.2 WiFi module

The Wifi module implements the IEEE 802.11 standard, attempting to provide an accurate MAC and PHY layer of that standard. A WifiNetDevice is equipped with a wireless NIC, based on 802.11, and consists of four “layers”; PHY layer, MAC-low, MAC-high, and Rate Control algorithms.

PHY layer models: PHY layer implements a single model of class WifiPhy, known as YansWifiPhy. This class is based on YANS (Yet Another Network Simulator) [32], a new network simulator that M. Lacage and T. Henderson presented. NS-3 PHY layer supports IEEE 802.11a/b/g and, currently, some features of 802.11n.

MAC-low layer: Generally, MAC-low takes care of RTS/CTS/data/Ack transactions. DCF and EDCAF functions are, also, implemented in that layer (MAC middle sub-layer), regarding to Non-QoS and QoS (802.11e) amendments in WLANs. Furthermore, two classes; DcaTxop and EdcaTxopN handle packet queues, fragmentation or packet retransmissions when QoS is disabled or supported, respectively. So far, only one aggregation scheme is supported (A-MSDU) and it is handled by EdcaTxopN.

MAC-high layer: This sub-layer is responsible for high level MAC management functions like probing, Association/ De-association and beacon generation. There are currently three MAC-high models; ApWifiMac, StaWifiMac, and AdhocWifiMac. The former two represent an AP and Non-AP nodes, respectively. The latter is the simplest of these models, and is used in an IBSS network (ad hoc network). All the three models share a common parent, RegularWifiMac, which encapsulates all MAC functionality.

Rate Control algorithms: There are two main categories of rate control algorithms, these with constant rate and these with adaptive rates. ConstantRate belongs to first category, while AarfWifiManager [33], CaraWifiManager [34], MinstrelWifiManager which was ported from MadWifi by Felix Fietkau, and many more.

Figure 19 depicts the architecture of a WifiNetDevice. To create that module, we have to create and configure five features:

- WifiChannel
- WifiPhy
- WifiMac
- WifiDevice
- Mobility

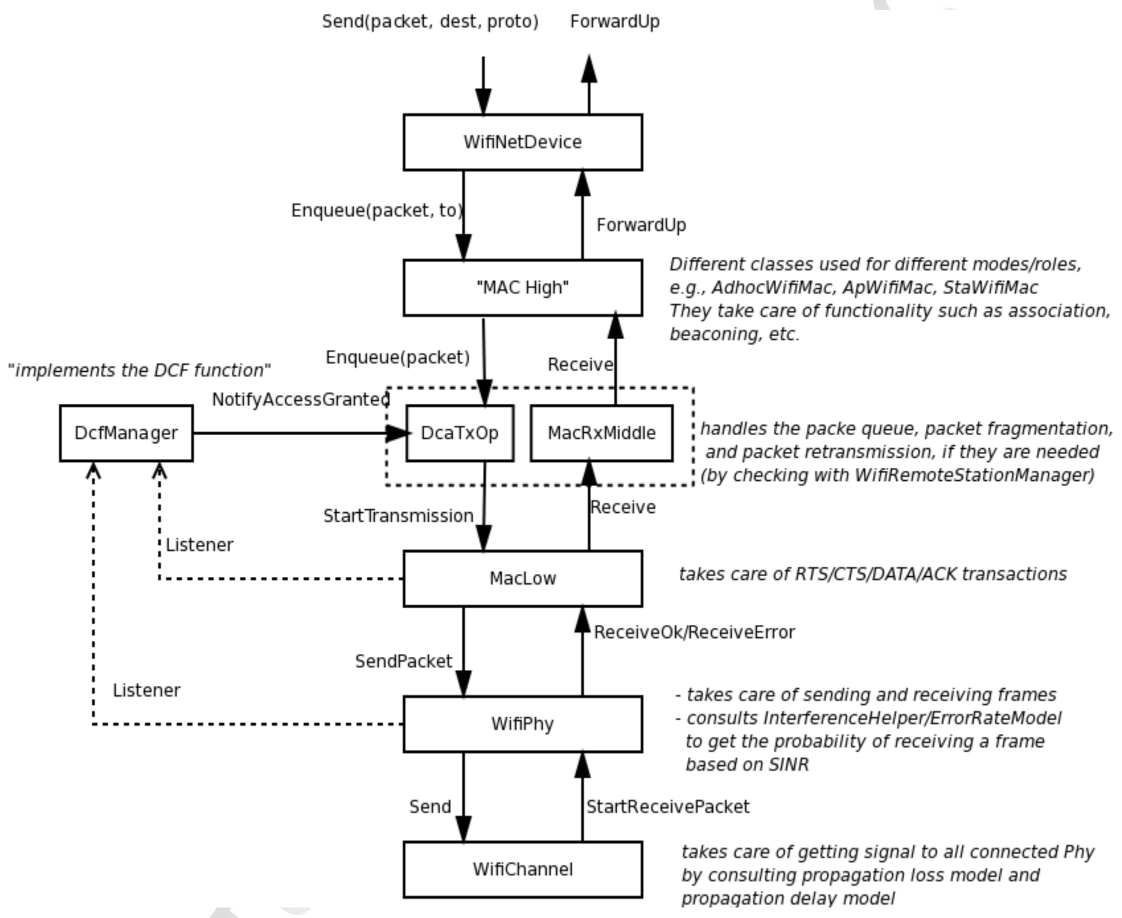

Figure 19 : WifiNetDevice architecture [35].

Despite the great development and the fact that it supports many new MAC and PHY features, ns-3 doesn't support all of 802.11n enhancements. In 2009, University of Florence (LART lab) has begun work on 802.11n model for ns-3. Through the last years many efforts have been made to fully support 11n, so far only few of these enhancements have added. A-MSDU scheme, Block Ack mechanism, HT mode, new coding schemes and preambles have already finished and supported on NS-3. On the other hand, MAC low aggregation (A-MPDU), TXOP[2], channel bonding and MIMO technology haven't been implemented yet. This work relies upon MAC features only, where several patches were provided in order to make them compatible with ns-3.

[2] Although TXOP was introduced in ns-3.5, is not fully supported.

Chapter 4

Enhancements

4.1 Fragmentation mechanism

Each network packet in NS-3 is consisted of four elements; a byte buffer, packet tags, byte tags, and metadata. Some of these are used in every packet, while others are optional and disabled by default.

Byte buffer stores all headers and tags that are added to a packet in a serialized way. These serialized headers are expected to match that of real network packets bit for bit, which means that the content of this buffer is alike to a real packet. On the other hand, tags are used by users to store data in packets that cannot be found in real packets. Two categories exist; byte tags and packet tags. The former is used to tag a set of bytes, while the latter to tag a packet itself. Metadata describes the type of headers and trailers that are attached in packets. That feature is optional and can be used only if a user enables it by calling Packet::EnableMetadata ().

Since every packet consists of a buffer of real bytes, it's easy to fragment it into multiple parts and re-assemble them. Each packet fragment is considered as a new packet, thus a copy of packet and byte tags follows each fragment. Following, we present a depth analysis of how fragmentation works in ns-3.

As it is shown in *Figure 19*, fragmentation occurs in a sub-layer of MAC low (MacRxMiddle) and is handled by WifiRemoteStationManager class. This class holds a list of attributes for every station and is accessible through the following paths Config::Set and Config::Connect. In other words, we can set or modify a set of attributes, like fragmentation threshold. By default fragmentation threshold is set to 2346 bytes, and only a user can change it by just typing

```
Config::SetDefault (“ns3::WifiRemoteStationManager::FragmentationThreshold”,
                    StringValue(“1500”));
```

in the script. The fragmentation threshold was set to 1500 bytes, which implies that a packet (including headers and data) bigger than this threshold will be split into multiple fragments. Note that headers and trailers from the initial packet will follow every fragment.

When a packet arrives in MAC from upper layers, it is queued and manipulated by DcfManager class. Once, node grants access to channel, the queued packet is being forwarded to EdcaTxopN or DcaTxopN class, regarding the status of QoS element. Function NotifyAccessGranted is responsible to check a set of parameters for the current packet, including fragmentation.

If packet's length is larger than fragmentation threshold, data will split into fragments. WifiRemoteStationManager is the class that handles all this procedure, which is consisted of the following steps:

- Calculate total number of fragments.
- Compute fragment size based on a specific threshold, except last one.
- Calculate the size of next fragment, if exists.
- Last fragment is consisted of the remaining bytes of the packet, and is tagged with a flag, revising receiver that this is the last fragment.

In case of a fail transmission, only the corrupted fragment is retransmitted.

In *Figure 20* the Fragmentation process is depicted as it was initially presented in ns-3.

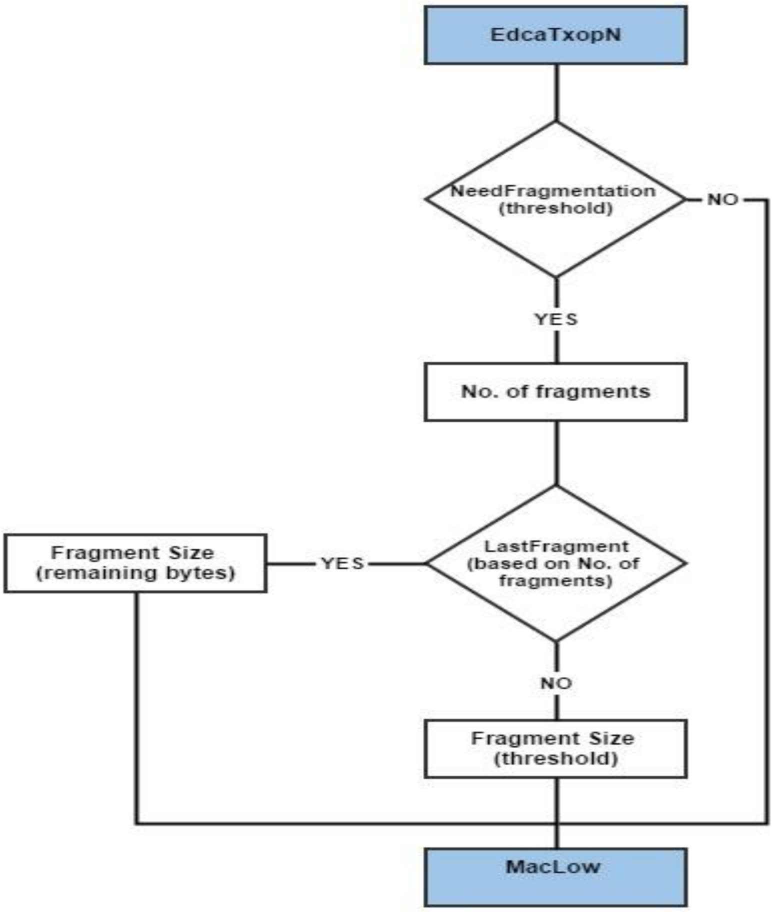

Figure 20 : Fragmentation procedure, as was initially implemented in ns3.

4.2 Fragmentation Enhancements

Fragmentation scheme is based on a fixed threshold, configured by user. This threshold cannot be changed throughout the simulations and it remains the same for all nodes. A major issue is derived from this, fairness among high-rate and low-rate stations. Initially, we defined an adaptive threshold based on transmission opportunity time (TxOP), depending on ACs that 802.11e presented.

Three classes are modified to implement an adaptive threshold; EdcaTxopN, WifiRemoteStationManager, and OnoffApplication. The former two are responsible for fragmentation procedure, while the latter is modified to support packets from different ACs. In the following paragraphs, these enhancements are discussed that are made in the above three classes, starting from higher layers.

Traffic is generated on Application layer, thus an attribute “AccessClass” in OnoffApplication class was added. Furthermore, a packet tag should be used in every packet to indicate the traffic Id (TID) that this packet belongs to. Hence, since a packet is created, a QosTag is attached on it (OnoffApplication::SendPacket), just before is sent to lower layers. It is important to note that the packet tag follows the packet and not each fragment.

Minor changes are done in EdcaTxopN, too. Due to adaptive way that the threshold is being calculated, an additional function is made for the next fragment size. This function, GetNextFragmentSize, calculates the size of next fragment. Size of next fragment might be different from the current, due to lower or higher data rates.

As it has already been mentioned previously, fragmentation process is mainly handled by WifiRemoteStationManager class, thus most of developments are performed here. Four new attributes are introduces, representing the four Access Categories.

```
.AddAttribute ("TxopVi", "Txop time (in milliSeconds) for packets that have been tagged with  AC_VI"
                "This value will not have any effect on some rate control algorithms.",
                TimeValue (Seconds (0.003008)),
                 MakeTimeAccessor (&WifiRemoteStationManager::m_txopVI),
                 MakeTimeChecker ())
```

Each one of these holds a TXOP that can be configured by the user, using a command alike to the one we used to set the fragmentation threshold.

DoSetTxopThreshold is a new function that converts TXOP time to bytes and estimates fragmentation threshold. TXOP is bounded below by zero and above by 32.76ms. Also, there is a limit on the size of fragment, as it cannot be smaller than 256 bytes. In addition, a condition checks threshold if it is even number, as it should be. A zero TXOP means that the packet will be transmitted as it is or it will be split based on fixed fragmentation.

Furthermore, function NeedFragmentation has modified, to support both TXOP and fixed fragmentation. Also, major changes are performed in GetFragmentSize, GetNextFragmentSize, GetFragmentOffset, and IsLastFragment functions in order to support 4 QoS queues. A retransmitted fragment might have different size from its initial, due to variety channel conditions. In other words, our approach is to implement a fully adaptive threshold. Next we present the effects of TXOP in fragmentation procedure, as it was implemented in ns3.

A simple infrastructure is mainly simulated- based on 802.11g system using Log Distance Propagation Loss Model, where loss in reference distance of 1m, at the 2.4GHz band using Friis eq. is 40.047 dB. This project's network contains two stations; an AP and a client-node (STA) transmitting packets. STA generates UDP traffic with a rate depending on Traffic category (approx. 24 Mbps) and packets with size of 2264 bytes. On the other hand, AP- server starts 0.5 seconds, at least, before client's. STA was relocated in different fixed positions and the process was repeated for each case and for three rate control algorithms with duration of 30 seconds per simulation. Since there is a warm up period of the NS3, everything is discarded during this time period (approx. 10seconds). Note that fixed fragmentation is disabled.

Furthermore, to provide accurate measurements of our tests we calculate:

- Average fragment size on PHY layer, where headers and tails are included.
- Average transmission delay of each packet based on *Table IX*.

Table IX
Transmission time of a 2264 bytes data packet (or its fragments).

DataRate (Mbps)	PacketSize (bytes)	BE_Txop: 0ms *No fragmentation*	BK_Txop: 0ms *No fragmentation*	VI_Txop: 3ms	VO_Txop: 1.5
1	2330	18.64	18.64	*3*	*2.048*
2	2330	9.32	9.32	*3*	*1.5*
5.5	2330	3.389	3.389	*3*	*1.5*
6	2330	3.106	3.106	*3*	*1.5*
9	2330	2.071	2.071	2.071	*1.5*
11	2330	1.694	1.694	1.694	*1.5*
12	2330	1.553	1.553	1.553	*1.5*
18	2330	1.035	1.035	1.035	1.035
24	2330	0.777	0.777	0.777	0.777
36	2330	0.517	0.517	0.517	0.517
48	2330	0.388	0.388	0.388	0.388
54	2330	0.345	0.345	0.345	0.345

Table IX shows the transmission time for each fragment size, regarding Access Category. It is important to note that transmission delay of the last fragment is not computed based on that table, as it consists of the remaining bytes of a packet. Note that transmission time for voice traffic, when raw rate is 1 Mbps, exceeds TXOP limit, due to minimum fragmentation threshold of 256 bytes.

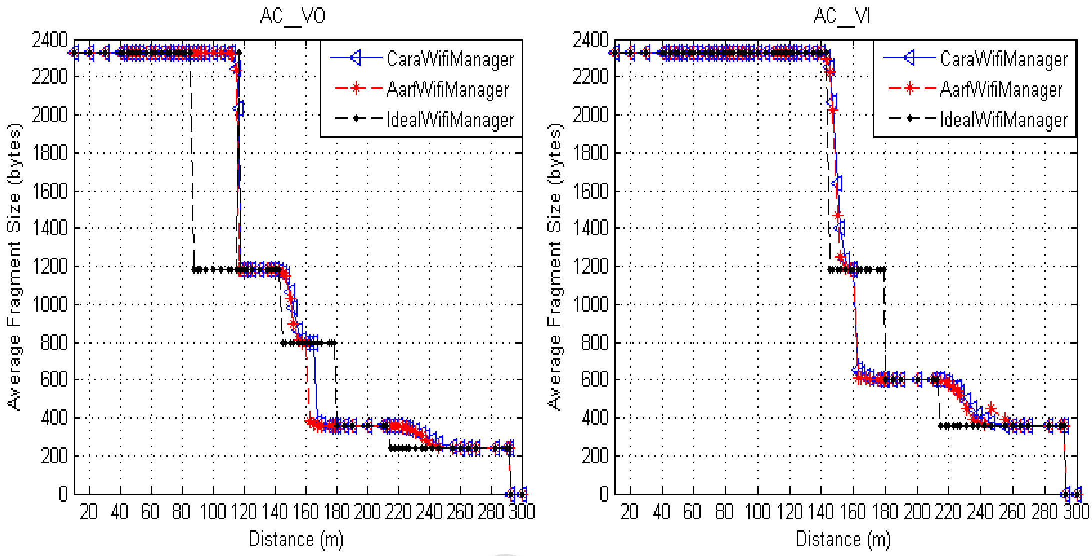

Figure 21 : Average fragment size for voice and video traffic.

Figure 21 shows the average fragment size per distance for three different adaptive rate control algorithms. Despite Aarf and Cara, Ideal algorithm is based, only on SNR. As node moves away from AP, data are transmitted in lower rates, thus smaller packets fit in a TXOP period.

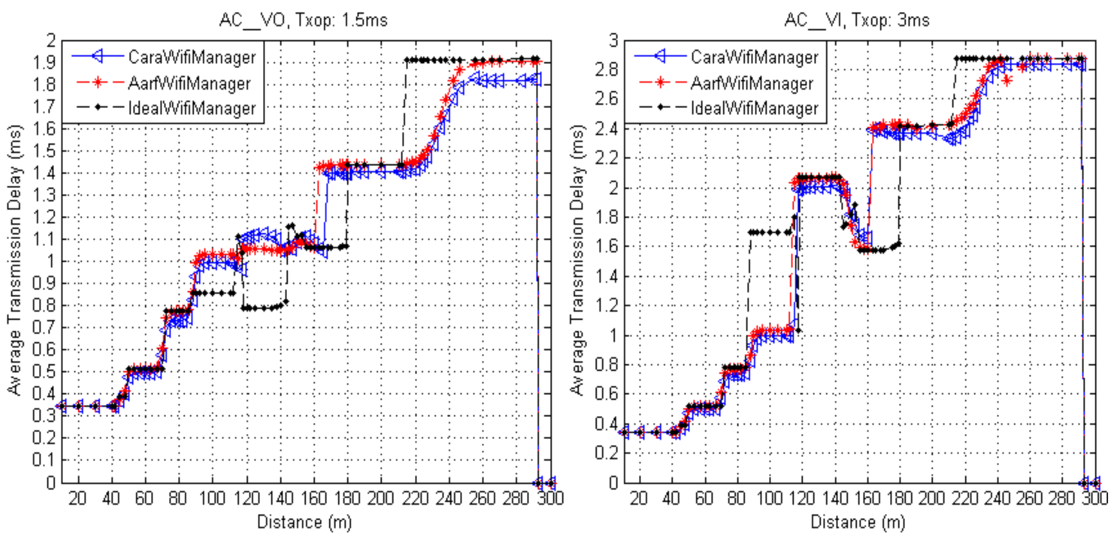

Figure 22 : Average transmission time for voice and video traffic.

Even though TXOP for voice traffic is set at 1.5ms, it can be seen in *Figure 22* that transmission time exceeds the limit. Fragmentation happens when data rate falls below 18 and 9 Mbps for voice and video traffic, respectively. *Figure 23* illustrates throughput for these TIDs for different positions of the node. Maximal throughput is achieved between 5 and 70 for both Ideal and Aarf rate control algorithms. On the other hand, Cara reaches its maximal throughput between 5 and 60 meters. Furthermore, throughput is limited by traffic generation rate that is 24 Mbps (approx.).

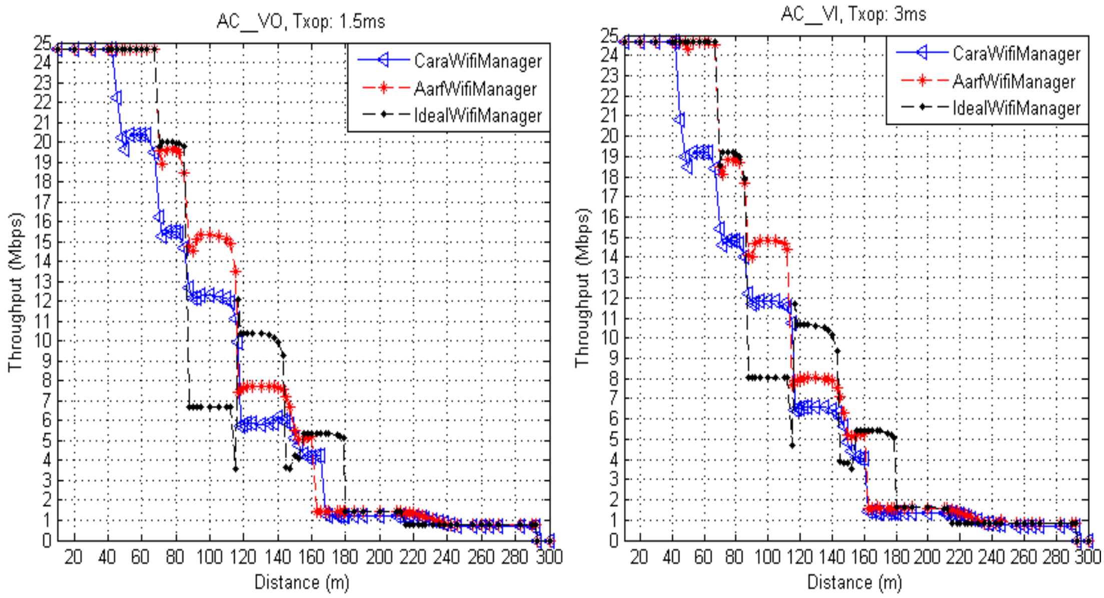

Figure 23 : System throughput for voice and video traffic, respectively.

So far a one-to-one topology is simulated with only one application installed on node, to evaluate performance of our implementation. Next set of figures present results for a one-to-one topology with three different client applications installed on the node.

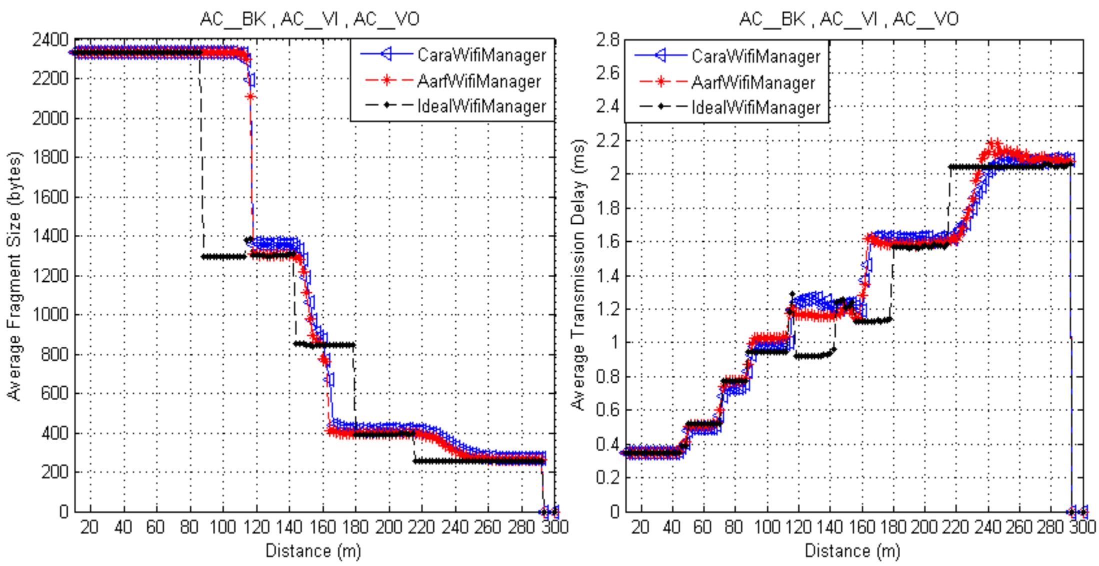

Figure 24 : Average frame size and tx time, when three applications are installed.

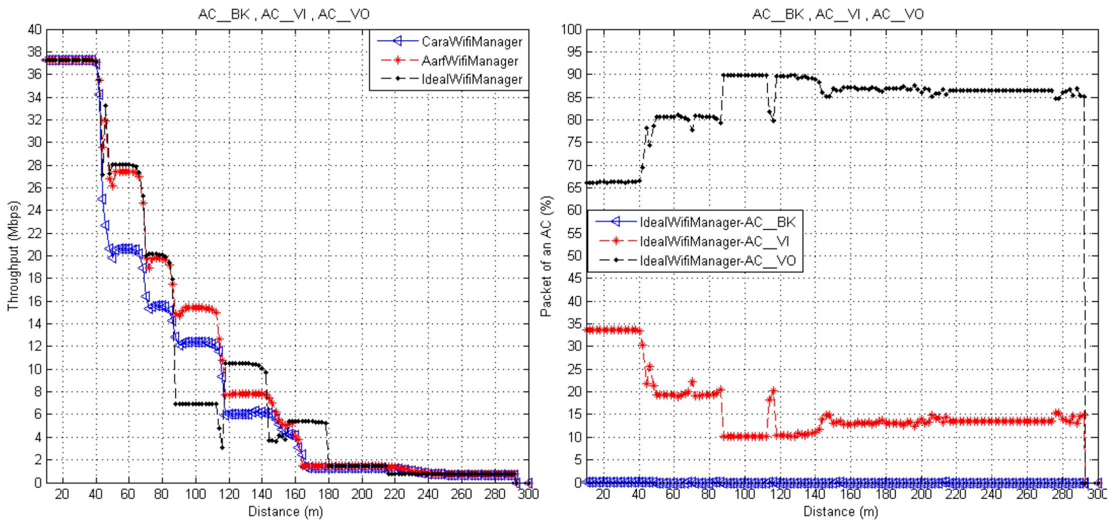

Figure 25 : System throughput and percentage of received packets of each AC.

In *Figure 24*, and *Figure 25* the system performance can be seen in terms of average fragment size, transmission time, and throughput as node is located in various positions. Furthermore, last sub-figure in *Figure 25*, shows the priority of Voice traffic over others two. Voice traffic has great effects in system's performance against background and video flows, even more for lower raw rates.

4.3 Aggregation Enhancements

So far, only one aggregation scheme is supported in NS3; MAC high aggregation mechanism, known as A-MSDU. Next figures show performance of this aggregation scheme for different ACs, keeping the same simulation parameters as previously. Maximum aggregation thresholds that were used during simulations are shown in *Table X*
Simulation parameters for A-MSDU scheme..

Table X
Simulation parameters for A-MSDU scheme.

Access Category	TXOP (ms)	A-MSDU threshold (bytes)
AC_BK	0	3839
AC_BE	0	3839
AC_VI	3.008	7935
AC_VO	1.504	7935

For background and best-effort traffic, threshold is 3839 bytes, because TXOP is zero, meaning that aggregation is based only on that threshold. Since transmission time is limited to 32ms, propagation delays over that time limit may occur for large frames when lower data rates are used.

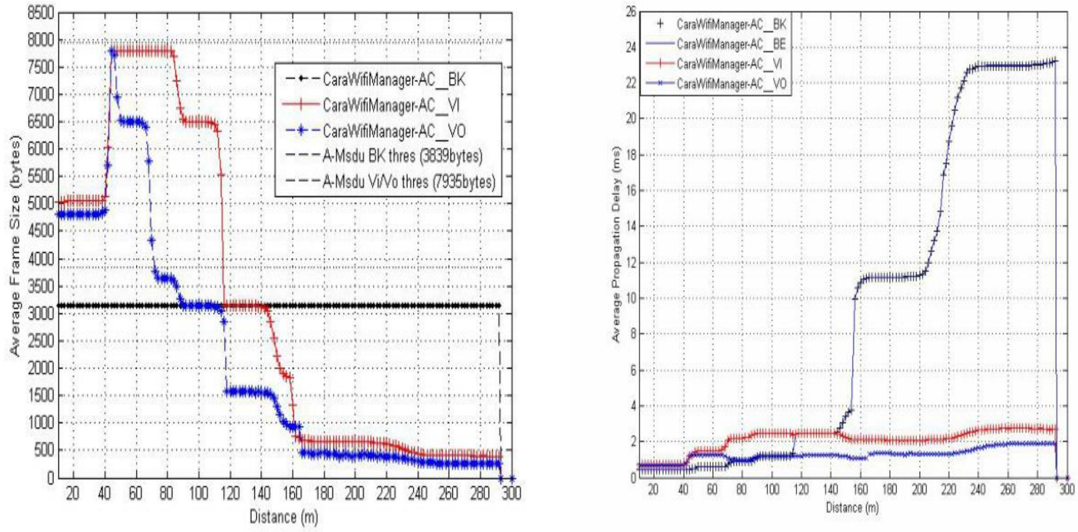

Figure 26 : Transmitted frame size and propagation delay (transmission time) per distance in a simple One-to-One topology.

Number of voice concatenated packets is less than video's, due to its double TXOP period. Close to AP, voice and video performance are alike. Whereas, the size of background packets remains fixed; despite distance between AP and STA. *Figure 26* illustrates this phenomenon. Furthermore, fragmentation occurs for both voice and video traffic after a distance of 140 meters. Note that we generate packets with size of 1500 bytes.

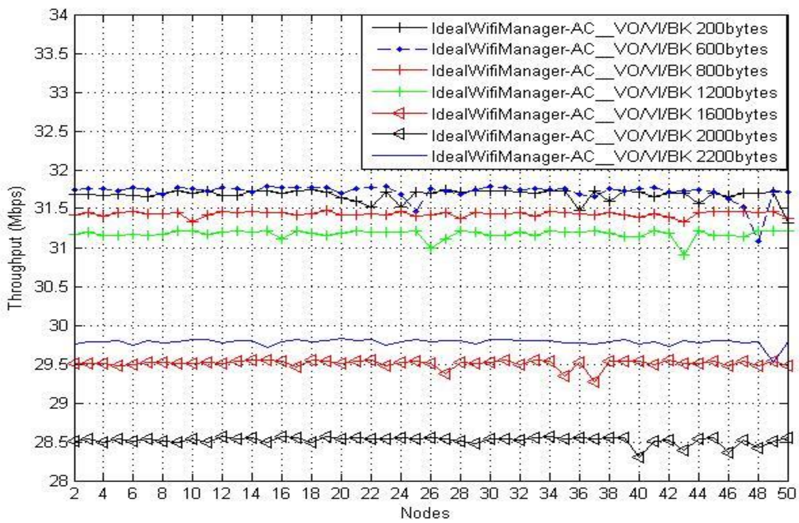

Figure 27 : Raw rate when three applications are installed in each node and A-MSDU is enabled.

On the other hand, transmission time stays below TXOP limit for both voice and video traffic, while for the background flow increases rapidly when channel conditions become worst. As a result, channel is being occupied for long time periods, leading to fairness issues between users and different TIDs. TXOP is a very crucial feature and we believe that it should be used for background and best-effort traffics to avoid these fairness issues.

Generally, aggregations scheme improve MAC and PHY rates by combining multiple packets into one large frame. A-MSDU threshold is limited to 8kB, thus the smaller packets are aggregated, the higher throughput is achieved (*Figure 27*).

Next a similar network topology is simulated, and it is presented in [16]. The authors proposed a frame adaptation algorithm for A-MSDU scheme, based on channel conditions and a feedback mechanism. Additionally, aggregation thresholds exceeded their initial values according to IEEE 802.11 standard. In contrast with that, we used the original concatenation mechanism based on TxOP values. Two values were used; 0.5ms and 1.5ms, leading to a threshold of 4Kbytes and 8Kbytes (maximum A-MSDU size according to 802.11n specs), respectively. Furthermore, data rates of 65 Mbps are used and an open space of 50m x 50m, where AP is located at the center of that area. There are *N* wireless nodes in the network, moving with a speed of 5m/s and pause time of 5s in each position. All nodes are saturated with CBR traffic of 65 Mbps, and packet size is 100 bytes. Only two applications installed in each node, Best-effort and Voice, due to their different TxOP values. The performance of burst mode was evaluated and is presented in *Figure 28*.

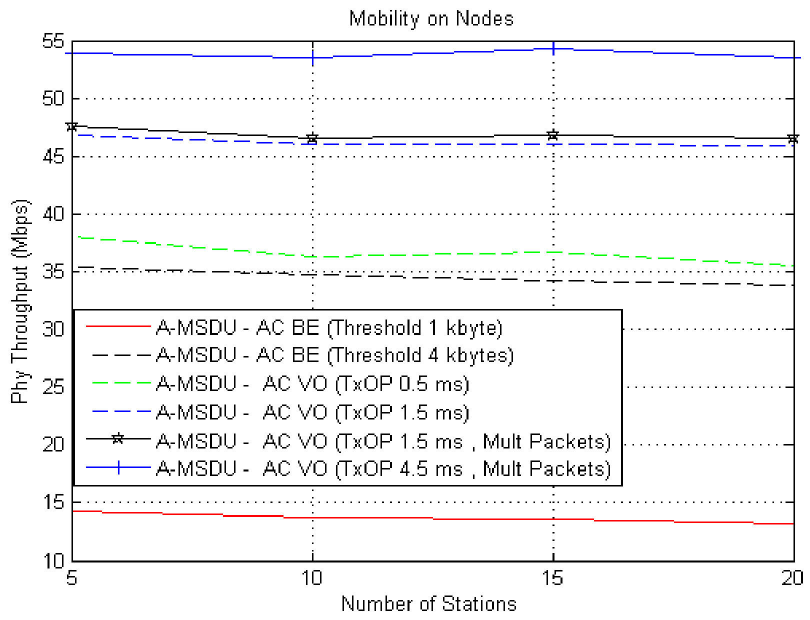

Figure 28 : PHY throughput per mobile node, when A-MSDU is used.

Considering *Figure 28*, EDCA parameters have great effects on system's performance, even if aggregation threshold is the same for all Access classes. A smaller contention window and AIFS increase throughput by 2-3 Mbps. Furthermore, burst mode increases PHY throughput by ~1 Mbps.

Following the second aggregation scheme is presented that is performed in MAC-low. 802.11n features were implemented by Ghada Badawy, including A-MPDU mechanism. Although, most of these features were published and added in NS3, there were some major omissions; especially in A-MPDU. We developed and contributed several patches to overcome these failures and added the third aggregation scheme; known as Two-Level aggregation. In addition, aggregation thresholds are adaptive, as all of these based on TXOP. Enhancements are also done to correct:

- RTS/CTS mechanism
- Acknowledgement timeout
- Number of A-MPDU frames
- Queueing procedure in MAC-low
- Rate control algorithms (Aarf) for new 802.11n data rates
- Burst mode during a TxOP period

Even though, 802.11n coding rates, data rates, short Guard Interval (GI), Greenfield and mixed capabilities were presented in ns-3.18 version (August 2014), A-MPDU, MIMO, and channel bonding are not currently supported. MAC low aggregation scheme is scheduled to be presented in v3-22 (ns-3.22), while the latter on a later version.

Table XI
IEEE 802.11n PHY/MAC parameters used in simulations.

Parameters	**Value**
Traffic rate	65 Mbps
Traffic type	UDP
TxOP	AC_BE: 0ms AC_BK: 0ms AC_VI: 3ms AC_VO: 1.504ms
CW_{min}, CW_{max}	default values (*Table IV*)
A-MSDU threshold	see *Table X*
A-MPDU threshold	10 Kbytes
Propagation Model	Log Distance Path Loss, $PL_0 = 40.047$ dB
PHY rate	65 Mbps

To evaluate performance of A-MPDU and Two-level scheme, various simulations were implemented using the parameters of 802.11n MAC/ PHY as listed in *Table XI*. In addition, the following assumptions were made: a node is considered to be fully buffered only when one application is installed, otherwise it is not considered to be fully buffered. In other words, traffic for an AC is generated with lower rate of raw rate (*Table XII*).

Table XII
Traffic rates when one application or multi-application mode is used.

Access Category	One application	Multi-application
AC_BE	65 Mbps	24 Mbps
AC_BK	65 Mbps	24 Mbps
AC_VI	65 Mbps	8 Mbps
AC_VO	65 Mbps	1 Mbps

In *Figure 29* the average aggregated frame size is illustrated for all aggregation schemes and MAC throughput for various packet sizes and TxOPs. It can be observed that A-MPDU mechanism concatenates more frames than A-MSDU. However it still doesn't fully utilize these aggregation schemes.

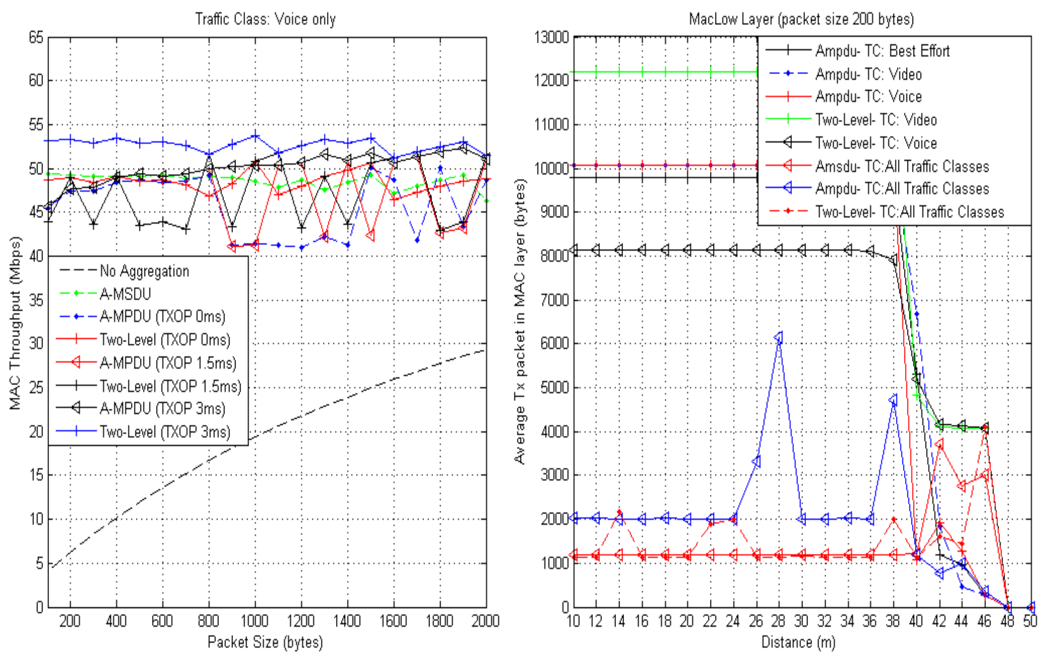

Figure 29 : MAC performance for different aggregation schemes.

A-MSDU's superiority over A-MPDU's can be observed for small packets, due to the fact that it carrier loess overhead over MAC low aggregation scheme [23]. When non-aggregation is used, throughput increases to packet size, achieving its maximal value for a 2 Kbytes packet. On the other hand, packet size doesn't have great effects when aggregation mechanisms are used. However, performance stabilizes as TxOP increases. Two-level scheme outperforms over others, when the same parameters are used for all of them.

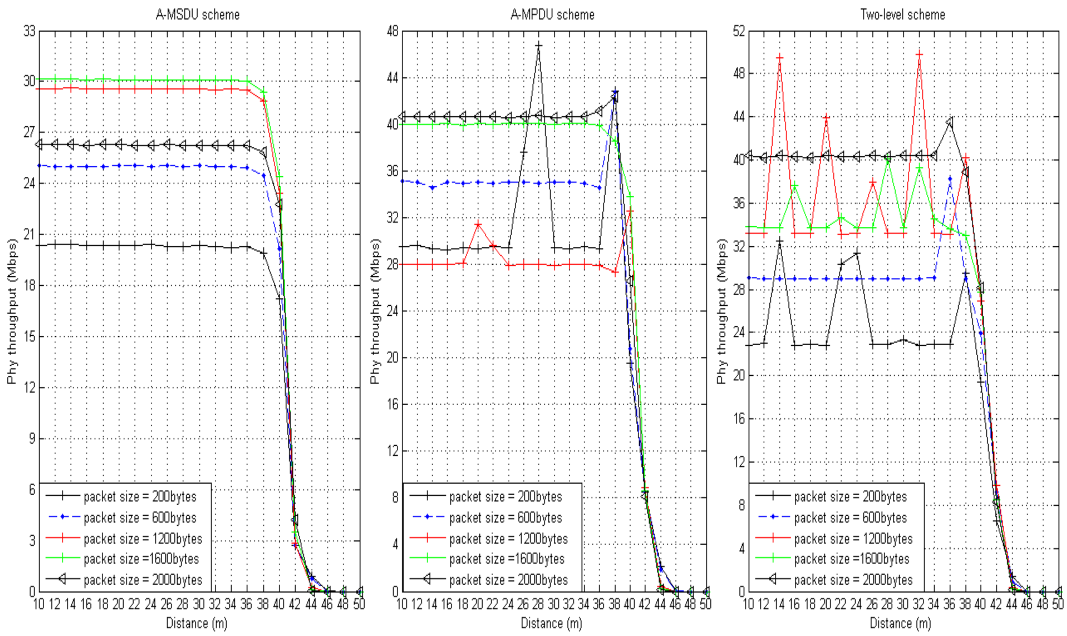

Figure 30 : Performance comparison of aggregation schemes, when all ACs are installed in one node.

The system's throughput is presented for aggregation mechanisms, when all ACs are installed on the STA. Performance in unsaturated conditions (*Figure 30*) is worse than fully buffered conditions; due to lower traffic rates of each AC (see Table XII). In contrast to fully buffered conditions (see Table XI) where throughput ranges 47 – 51 Mbps for A-MSDU and 51 – 52 Mbps for A-MPDU, here, varies from 19 – 49 Mbps. In most of cases A-MPDU and Two-level mechanisms outperform over A-MSDU, increasing throughput by 7- 12 Mbps. Note variance of performance in A-MPDU and Two-level schemes due to large threshold and unsaturated conditions. However, performance can be improved more by setting threshold to its maximal value (64Kbytes) and supporting burst mode during a TxOP period.

MAC performance is resented when multiple packets can be sent during a TxOP period (*Figure 31*). The voice traffic is generated and default values of EDCA parameters are used, which implies that TxOP is set to 1.504ms. Furthermore, the overhead was calculated which is added in data packets, including management frames, headers, and retransmitted data packets. As it can be observed, the aggregation mechanisms increase the number of redundant bytes in comparison to non-aggregation scheme. However, interframe spaces between packets are reduced by concatenating them into one larger frame, leading to better throughput.

It is important to note that A-MSDU mechanism concatenates packets with one MAC header, while in A-MPDU every frame has its own MAC header. This contradicts the results depicted in *Figure 31*. However, the results lie in the fact that retransmitted bytes and aggregation thresholds that are used for the above schemes are different.

Here, the retransmission and size of the aggregated packets need to be considered.

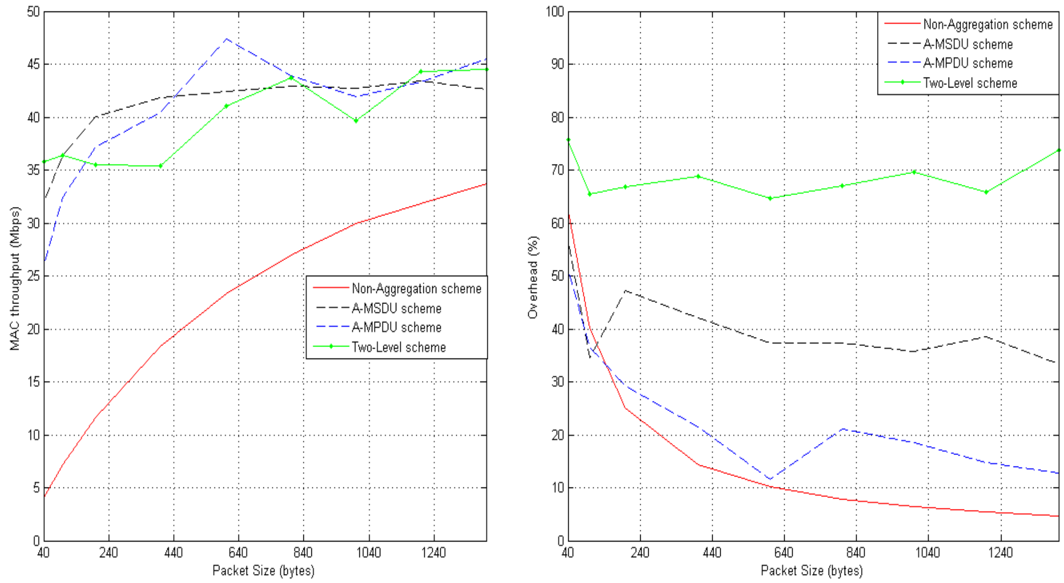

Figure 31 : MAC throughput and overhead for aggregation schemes.

4.4 Impact of EDCA parameters

In this section the impact of EDCA parameters is shown in the system performance. For simplicity reasons, the less cumbersome scheme is adopted, which is the non-aggregation one. We use one AP and one mobile station, under fully buffered conditions.

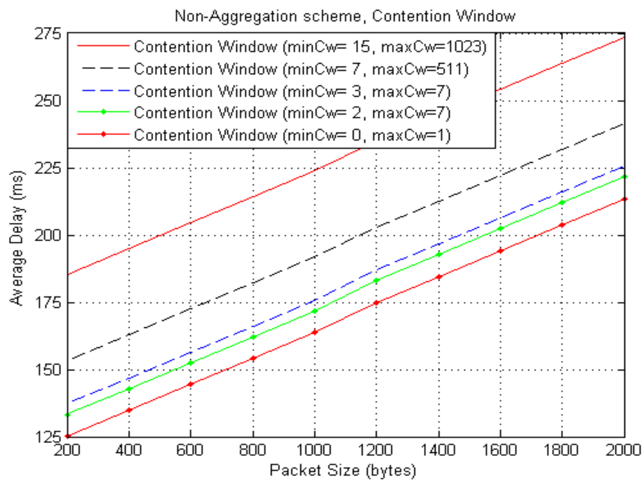

Figure 32 : Contention Window effects on system performance.

Figure 32 illustrates the system performance under different Contention Window boundaries. Performance of a SISO system over error-free channel is affected only be the lower bound (CW_{min}). By diminishing these limits, average delay decreases too, while throughput slightly improves. In dense networks, where multiple nodes contend for the medium, collisions might occur. Backoff will be increased for the collided nodes up to its maximum value (CW_{max}). Under these circumstances, the upper bound should be selected carefully.

As CW_{min} decreases by half, packet delay drops. This reduction in packet delay improves system's throughput by 2-3 Mbps. The highest delay is achieved when the maximum values of Contention window are used, similar to the Background traffic. On the other hand by using the initial CW limits of Voice traffic, delay falls almost by 40ms. This figure makes the prioritization of Voice over Background class clear. Note that for the delay the queuing and transmission time of a packet is assumed.

Contention Window reduces packet delay, but it doesn't improve system's goodput. Hence, a test case where TxOP is modified is simulated. *Figure 33* shows the system's performance of the specific case (*Table XIII*).

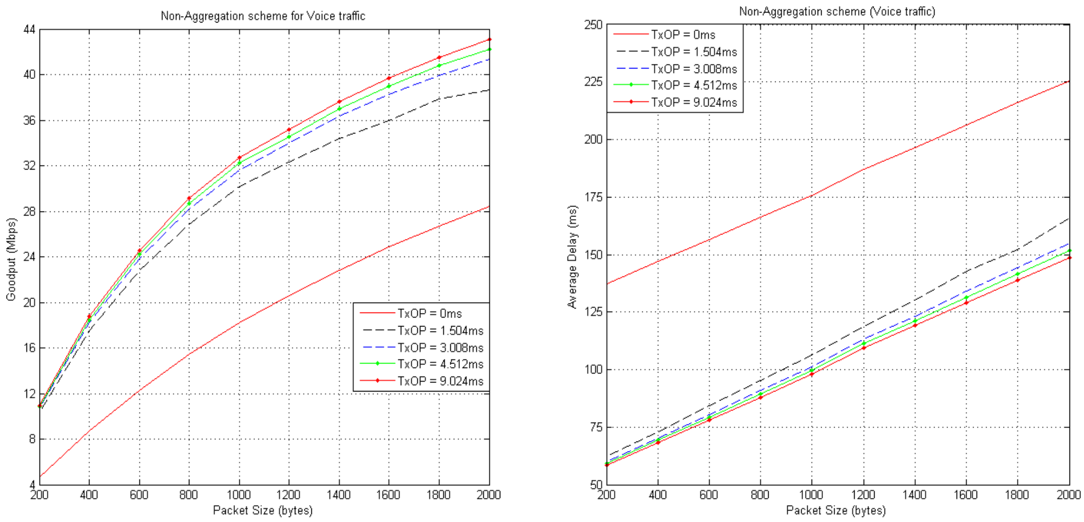

Figure 33 : TxOP effects on system performance.

In the contrary to the CW, TxOP parameter has important effects on the system's performance, in terms of goodput and packet delay. It can be observed that by just using TxOP's default value (AC_VO), goodput is increased rapidly/dramatically, while the packet delay drops by 70ms approximately. Throughput seems to be stable around 42 Mbps, for large packets and TxOPs over 3.008ms. During a TxOP period, multiple data frames are transmitted without RTS/CTS to be interjected between frames.

Thus, redundancy bytes are reduced improving throughput and minimizing delays.

Figure 34 presents the system's performance under different values of AIFSN together with the maximum achievable goodput. Note that the default backoff algorithm was used, as in a collision-free environment backoff algorithms don't affect network performance.

Table XIII
EDCA parameters that are used in Figure 33.

Curve	AIFSN	CW_{min}	CW_{max}	Backoff alg.	TxOP(ms)
Sim A – Aifs (1)	1	0	1	Default	0
Sim B – Aifs (0)	0	0	1	Default	0
Sim C – TxOP	2	3	7	Default	9.024
Sim Optimal	0	0	1	Default	9.024

So far we have presented how performance can be improved by modifying three of EDCA parameters, CW bounds, AIFSN and TxOP period. The former two reduce packet delay, while the latter greatly affects system's performance. Goodput and delay are improved and reduced respectively, by a factor of 1.5. However, we assumed that transmitter is fully buffered and has always packets to send, exploiting the entire TxOP period. In real networks, where nodes generate traffic with lower rates (unsaturated conditions) increasing TxOP may have no effects in performance.

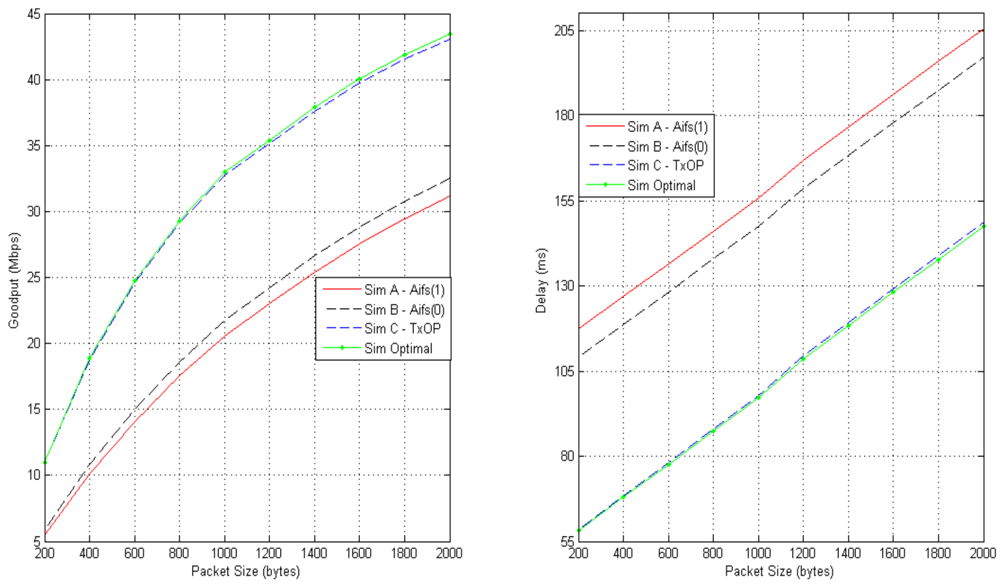

Figure 34 : System performance with various AIFSN and TxOP values.

Presumed upon the above advantages, we configure MAC aggregation (*Figure 35*) and compare its performance with this in *Figure 31*. The values of EDCA parameters that we use are listed in *Table XIV*.

Table XIV
EDCA parameters that we used in Figure 35.

Curve	AIFSN	CW_{min}	CW_{max}	Backoff alg.	TxOP(ms)
A-MSDU	2	3	7	Default	1.504
A-MPDU	2	3	7	Default	1.504
A-MSDU sim	1	1	3	(CW + 1, CW_{max})	3.008
A-MPDU sim	1	1	3	(CW + 1, CW_{max})	3.008

A-MSDU and Two-Level schemes take advantage of these new values, improving their performance up to 10%. However, A-MPDU has worse throughput than its original scheme, when small packets are aggregated. This flows from the additional dummy bytes attached in small frames, as it was explained before. Neither TxOP nor others EDCA parameters seem to fix this drawback of that mechanism. On the contrary, for large packets modified A-MPDU scheme outperforms over its original. However, most of wireless traffic is based on packets smaller than 400 bytes, e.g. VoIP frames are 40bytes according to G.711 codec.

Based on that assumption, we simulate a network where each node generates voice, video, and background traffic with the following rates and frame sizes:

- VoIP traffic: 64kbps and packet size of 40bytes.
- Video traffic: 250kbps and packet size of 512bytes.
- Background traffic: 10kbps and packet size of 256bytes.

In addition, we set TxOP of Background traffic to 0.25ms and reduce by half all EDCA parameters in order to reduce average delay.

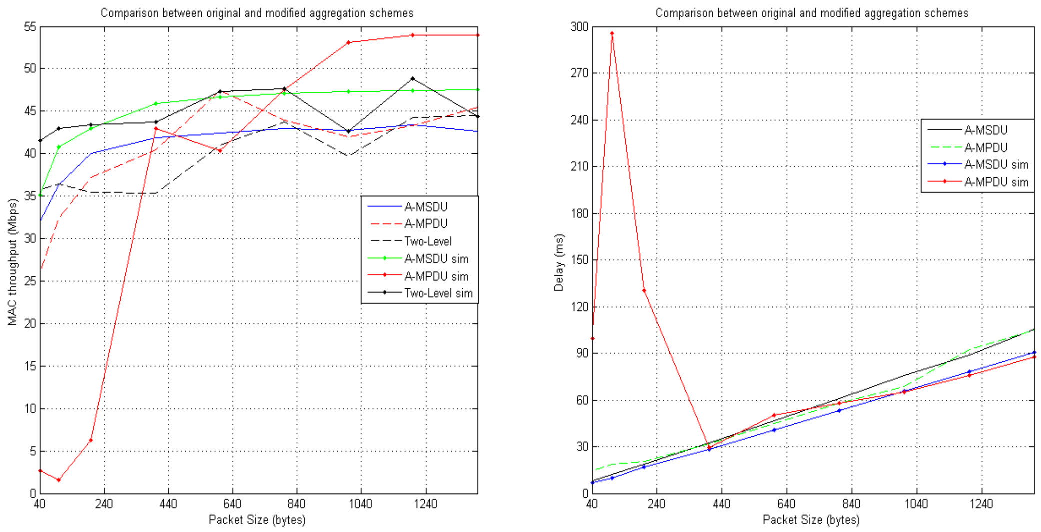

Figure 35 : Comparison of original and modified aggregation schemes.

We can see that for A-MPDU and Two-Level aggregations, average packet delay is increased rapidly to number of users. In contrast, this is not occurred for A-MSDU. The simulation results (*Figure 36*) show that delay is affected by the BER, the aggregated packet size, and the number of competing nodes. However, the former two schemes achieve better throughput than the latter. Therefore, there is a trade-off between maximizing throughput and minimizing packet delay. One way to control delay is by adjusting the aggregated packet size.

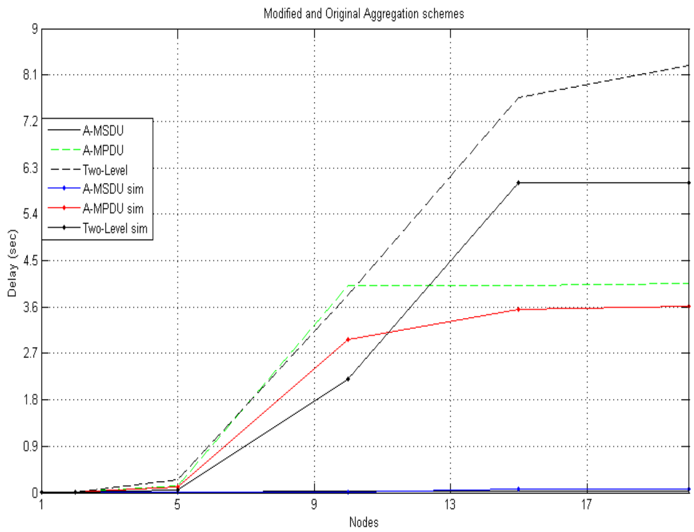

Figure 36 : Average packet delay in a multi-user network.

Chapter 5

Conclusions and Future work

In this research we analysed the performance of 802.11n WLAN under various environments. We focused on MAC layer, as aggregation schemes seem a promising technology that can enhance MAC throughput by reducing overhead. All aggregation schemes are tested for different thresholds based on TxOP, as this feature is implemented too.

Simulation results show the major impact of TxOP in fragmentation and aggregation procedures. Fragmentation threshold based on TxOP value, gives more freedom on a station. Under heavy load or when BER is high and a node operates in lower data rates, it should transmit smaller frames to occupy the medium for a period no longer than TxOP. In other words, this mechanism works in an adaptive way, constraining packet delays not to exceed of TxOP. Aggregation schemes improve MAC throughput, almost by a factor of 2. A-MSDU outperforms over A-MPDU for small packets and under unsaturated conditions. However, we can overcome this issue by combining these two aggregation mechanism, known as Two-Level aggregation scheme. A new adaptive mechanism should be approached that decides which scheme to use. Its decision should be based on queue size, current conditions of the medium and the number of competing nodes.

Regarding EDCA parameters, outcome results show that TxOP has great effects in system's performance. A TxOP of 1.504ms gives better results over a TxOP of 0ms. In terms of performance, throughput is almost doubled and packet delay is dropped by half. For TxOP over 4.512ms system's performance is, slightly improved. On the other hand, by reducing AIFSN and Contention Window boundaries, average packet delay decreases, too. Unlike that, these parameters hardly affect MAC throughput.

In dense networks under unsaturated conditions A-MSDU outperforms over the others, even if we modify EDCA parameters for A-MPDU and Two-Level schemes. However, by reducing Contention Window, AIFSN, and backoff algorithm (CW_{min}+1, CW_{max}) only A-MSDU has worse throughput and average delay than its original EDCA parameters. A trade-off between throughput and packet delay was derived from our simulations. One more thing that should be mentioned is the performance improvement by using TxOP for Background and/or Best Effort traffic. These Access Classes are delay-tolerant; however fairness issues between ACs occur when a TxOP is not used for these categories.

Although, this research covers many test cases, there still remain scenarios which are not discussed, as 802.11n performance in multi-cell and error-prone environments.

References

[1] H. Wu, S. Cheng, Y.Peng, K. Long, and J. Ma, "IEEE 802.11 distributed coordination function (DCF): analysis and enhancement," in Proc. of IEEE Int. Conference on Communications (ICC), New York, NY, USA, April 28 – May 2 2002, vol. 1, pp. 605 – 609.

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," in Proc. of the conference on Communications architectures, protocols and applications. 1994, pp. 212 – 225, ACM Press.

[3] J. Deng, P. K. Varshney, and Z. J. Haas. "A new backoff algorithm for the IEEE 802.11 distributed coordination function." In Communication Networks and Distributed Systems Modeling and Simulation (CNDS '04), 2004.

[4] H. Minooei and H. Nojumi. "Performance evaluation of a new backoff method for IEEE 802.11", Comput. Commun., vol. 30, no. 18, pp.3698 - 3704 2007.

[5] J. Bianchi, "Performance Analysis of IEEE 802.11 Distributed Coordination Function", IEEE J. Select. Areas Commun., Vol. 18, No. 3, March 2000, pp. 535–547.

[6] Taijun Li, Tiebin Tang, Cheng Chang, "A new backoff algorithm for the IEEE 802.11 distributed coordination function", Proc. FSKD 09, PP 455-459, 2009.

[7] A. Koepsel, J.-P. Ebert and A. Wolisz, "A Performance Comparison of Point and Distributed Coordination Function of an IEEE 802.11 WLAN in the Presence of Real-Time Requirements," Proc of MoMuC2000, Tokyo.

[8] Z.Chen and A.Khokhar, "Improved MAC protocols for DCF and PCF modes over fading channels in wireless LANs", in Proc. Wireless Communications Network Conf:, vol. 2, pp. 1297-1302, 2003.

[9] IEEE Std. 802.11e. 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 8: Medium Access Control (MAC) Quality of Service Enhancements, November 2005.

[10] IEEE Std. 802.11™ – 2012, 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, March 2012.

[11] J. Majkowski, and F. C. Palacio, “Dynamic TXOP configuration for Qos enhancement in IEEE 802.11e wireless LAN,” Software in Telecommunications and Computer Networks, 2006. SoftCOM 2006. International Conference on, pp. 66-70, Sept. 29 2006-Oct. 1 2006.

[12] A. Andreadis, and R. Zambon, “QoS Enhancement with Dynamic TXOP Allocation in IEEE 802.11e”, Personal, Indoor and Mobile Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on, pp. 1-5, 3-7 Sept. 2007.

[13] Y. Xiao, “IEEE 802.11n: Enhancements for Higher Throughput in Wireless LANs,” IEEE Wireless Comm. Magazine, vol. 12, no. 6, pp. 82-91, Dec. 2005.

[14] D. Skordoulis, Q. Ni, H.H.Chen, A.P. Stephen s, C. Liu, A. Jamalipour, "IEEE 802 .1 In MAC frame aggregation mechanism for next-generation high throughput WLAN s," IEEE Wireless Commun., February 2008.

[15] J. Deng, and M. Davis, “An Adaptive Packet Aggregation Algorithm for Wireless Networks”, Wireless Communications & Signal Processing (WCSP), 2013 International Conference on, pp. 1-6, 24-26 Oct. 2013.

[16] Y. Lin, and V. W. S. Wong, “Frame Aggregation and Optimal Frame Adaptation for IEEE 802.11n WLANs”, Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE, pp. 1-6, Nov. 27 2006-Dec. 1 2006.

[17] N. Hajlaoui, I. Jabri, and M. B. Jemaa, “Analytical Study of Frame Aggregation in Error-prone Channels”, Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, pp. 237-242, 1-5 July 2013.

[18] N. Hajlaoui, and I. Jabri, “On the Performance of IEEE 802.11n Protocol”, Wireless and Mobile Networking Conference (WMNC), 2012 5th Joint IFIP, pp. 64-69, 19-21 Sept. 2012.

[19] N. Hajlaoui, I. Jabri, M. Taieb, and M. Benjemaa, “A Frame Aggregation Scheduler for QoS-sensitive applications in IEEE 802.11n WLANs”, Communications and Information Technology (ICCIT), 2012 International Conference on, pp. 221-226, 26-28 June 2012.

[20] M. G. Sarret, J. S. Ashta, and P. Mogensen, “A multi-QoS Aggregation Mechanism for Improved Fairness in WLAN”, Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th, p. 1-5, 2-5 Sept. 2013.

[21] M. Adnan, and E. C. Park, “Assuring per-station fairness in multi-rate WLANs: A hybrid approach of contention window control and frame aggregation,” Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on, pp. 282-287, 2-5 July 2013.

[22] J. Kolap, S. Krishnan, and N. Shaha, “Comparison of Frame Aggregation Mechanism in 802.11n WLAN,” Communication, Information & Computing Technology (ICCICT), 2012 International Conference on, pp. 1-6, 19-20 Oct. 2012.

[23] Y. Kim, E. Monroy, O. Lee, K. J. Park, and S. Choi, “Adaptive Two-Level Frame Aggregation in IEEE 802.11n WLAN,” Communications (APCC), 2012 18th Asia-Pacific Conference on, pp. 658-663, 15-17 Oct. 2012.

[24] E. Perahia, and R. Stacey (2013). Next Generation Wireless LANs 802.11n and 802.11ac. 2nd ed. Cambridge: Cambridge University Press.

[25] B. Saltzberg, “Performance of an Efficient Parallel Data Transmission System,” Communication Technology, IEEE Transactions on, Vol. COM-15, No.6, pp. 805-811, Dec. 1967.

[26] R. Prasad (2004). OFDM for Wireless Communications Systems. Artech House Universal Personal Communications Series.

[27] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas", *Wireless Pers. Commun.*, vol. 6, pp.311 -335 1998.

[28] S. M. Alamouti, "A simple transmitter diversity scheme for wireless communications", *IEEE J. Select. Areas Commun.*, vol. 16, pp.1451 -1458, 1998.

[29] D.Tse, and P. Viswanath (2004), Fundamentals of Wireless Communication, Cambridge University Press, New York.

[30] P. W. Wolniasky, G. J. Foschini, G. D. Golden, and R. Valenzuela, “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel,” Signals, Systems, and Electronics, 1998. ISSSE 98. 1998 URSI International Symposium on, pp. 295-300, 28 Sept - 2 Oct 1998.

[31] L. Deek, E.G. Villegas, E. Belding, S.J. Lee, and K. Almeroth, “The Impact of Channel Bonding on 802.11n Network

Management,”CoNEXT ‘11 Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies, 2011.

[32] M. Lacage, and T. R. Henderson, “Yet Another Network Simulator,” In WNS2 ’06: Proceeding from the 2006 Workshop on ns-2: the IP network simulator, New York, NY, USA, October 2006, ACM.

[33] M. Lacage, M. H. Manshaei, and T. Turletti, “IEEE 802.11 Rate Adaptation: A Practical Approach,” Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, October 04-06, 2004, Venice, Italy.

[34] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-aware Rate Adaptation for IEEE 802.11 WLANs. IEEE INFOCOM, 2006.

[35] The ns-3 network simulator. http://www.nsnam.org.