Wireless Audio Networking

Modifying the IEEE 802.11 standard to handle multi-channel, real-time wireless audio networks

A thesis submitted for the degree of Doctor of Philosophy

by

Christos Chousidis

College of Engineering,
Design and Physical Sciences
Brunel University
August 2014

Abstract

Audio networking is a rapidly increasing field which introduces new exiting possibilities for the professional audio industry. When well established, it will drastically change the way live sound systems will be designed, built and used. Today's networks have enough bandwidth that enables them to transfer hundreds of high quality audio channels, replacing analogue cables and intricate installations of conventional analogue audio systems. Currently there are many systems in the market that distribute audio over networks for live music and studio applications, but this technology is not yet widespread. The reasons that audio networks are not as popular as it was expected are mainly the lack of interoperability between different vendors and still, the need of a wired network infrastructure. Therefore, the development of a wireless digital audio networking system based on the existing widespread wireless technology is a major research challenge. However, the IEEE 802.11 standard, which is the primary wireless networking technology today, appears to be unable to handle this type of application despite the large bandwidth available. Apart from the well-known drawbacks of interference and security, encountered in all wireless data transmission systems, the way that IEEE 802.11 arbitrates the wireless channel access causes significantly high collision rate, low throughput and long overall delay.

The aim of this research was to identify the causes that impede this technology to support real time wireless audio networks and to propose possible solutions. Initially the standard was tested thoroughly using a data traffic model which emulates a multi-channel real time audio environment. Broadcasting was found to be the optimal communication method, in order to satisfy the intolerance of live audio, when it comes to delay. The results were analysed and the drawback was identified in the hereditary weakness of the IEEE 802.11 standard to manage broadcasting, from multiple sources in the same network. To resolve this, a series of modifications was proposed for the Medium Access Control algorithm of the standard. First, the extended use of the "CTS-to-Self" control message was introduced in order to act as a protection mechanism in broadcasting, similar to the RTC/CTS protection mechanism, already used in unicast transmission. Then, an alternative "random backoff" method was proposed taking into account the characteristics of live audio wireless networks. For this method a novel "Exclusive Backoff Number Allocation" (EBNA) algorithm was designed aiming to minimize collisions. The results showed that significant improvement in throughput can be achieved using the above modifications but further improvement was needed, when it comes to delay, in order to reach the internationally accepted standards for real time audio delivery. Thus, a traffic adaptive version of the EBNA algorithm was designed. This algorithm monitors the traffic in the network, calculates the probability of collision and accordingly switches between classic IEEE 802.11 MAC and EBNA which is applied only between active stations, rather than to all stations in the network. All amendments were designed to operate as an alternative mode of the existing technology rather as an independent proprietary system. For this reason interoperability with classic IEEE 802.11 was also tested and analysed at the last part of this research. The results showed that the IEEE 802.11 standard, suitably modified, is able to support multiple broadcasting transmission and therefore it can be the platform upon which, the future wireless audio networks will be developed.

Contents

List of figures

List of tables

Abbreviations

3G	Third Generation of mobile telecommunications
A/D	Analogue to Digital
AAC	Advance Audio Coding
AAC-LD	Advance Audio Coding
ACK	Acknowledgment
ADAT	Alesis Digital Audio Tape
AES	Audio Engineering Society
AoE	Audio over Ethernet
AoIP	Audio over Internet Protocol
AP	Access Point
ATM	Asynchronous Transfer Mode
BSS	Basic Service Set
CD	Compact Disk
CFP	Consecutive Freeze Process
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Sent
CTS-to-self	Clear To Sent-to-self
CTSTT	CTS-to-self Transfer Time
CW	Contention Window
D/A	Digital to Analogue
DCF	Distributed Coordination Function
DES	Discreet Event Simulation
DIFS	Distributed Coordination Function Inter-Frame Space
DSSS	Direct Sequence Spread Spectrum
EBNA	Exclusive Backoff Number Allocation
EBU	Europeans Broadcast Union
EDCA	Enhanced Distributed Channel Access
EL	Event List
ERP	Extended Rate Physical
FHSS	Frequency Hopping Spread Spectrum
FM	Frequency Modulation
FSM	Finite State Machine

GPS	Global Positioning System
HCF	Hybrid Coordination Function
HDTV	High Definition Television
H-EBNA	Hybrid Exclusive Backoff Number Allocation
Hi-Fi	High Fidelity
HR	High Rate
HT	High Throughput
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers.
IP	Internet Protocol
ISDN	Integrated Services for Digital Network
ITU-T	International Telecommunication Union - Telecommunication
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MADI	Multichannel Audio Digital Interface
MIDI	Musical Instruments Digital Interface
MOS	Mean Opinion Score
MPEG	Motion Picture Experts Group
MTU	Maximum Transmission Unit
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
PA	Public Address
PCF	Point Coordination Function
PCM	Pulse Code Modulation
PDF	Probability Density Function
PESQ	Perceptual Evaluation of Speech Quality
PHY	Physical
PI	Packetization Interval
PTT	Packet Transfer Time
QoS	Quality of Service
RTCP	Real Time Control Protocol
RTP	Real Time Protocol

RTS	Request To Send
SIFS	Short Inter-Frame Space
SIP	Session Initiation Protocol
STA	Station
STID	Station ID
TC-NAS	Technical Committee on Network Audio Systems
TDIF	Tascam Digital Interface
TDM	Time Division Multiplexing
TXOP	Transmission Opportunities
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPA	Wireless Application Protocol
WPAN	Wireless Personal Aria Network
WSTA	Wireless Station

Acknowledgment

Firstly I would like to express my gratitude to my supervisor Dr Rajagopal Nilavalan for his constant support and guidance throughout the past 3 years. His understanding and encouragement were crucial for the outcome of this research. I would also like to thank my initial supervisor Dr Marios Hadjinicolaou who so suddenly passed away during the second year of my studies and I would like to dedicate this work to his memory.

I would like to thank Prof Tassos Karayiannis for his support and his valuable advices during all this period of my research. In addition, I would like to thank Prof John Cosmas, Prof Hamed Al-Raweshidy and Prof Marios Angelidis for their encouragement and their comments regarding my research. Also, I would like to thank Dr Harry Agius and Dr Arthur Money for their friendly help and support.

I would also like to express my thanks to my friend Dimosthenis Trichakis and Giannis Lianakis for encouraging me and sharing their knowledge during the first steps of my research, and to all my friends and my family for their support.

Finally I would like to express my gratitude to my beloved wife Ioana Pisica who inspired me, supports me and helped me during the course of this research. Without her endless love, patience and encouragement this work would not have been possible.

Author's Declaration

The work described in this thesis has not been previously submitted for a degree in this or any other university and unless otherwise referenced it is the author's own work.

Chapter 1

Introduction

1.1 Overview

The following section will outline the motivation behind the research presented in this thesis. A brief analysis of the relevant industry as it stands today is provided in order to outline the importance of the research and the problems intended to solve. The overall objectives of this research project are presented within the context of improving the development of professional audio networks by enabling the use of wireless data networking technology, which provides a plethora of advantages such as mobility, ease of network configuration and low cost. The contributions to knowledge are also described in this section and the publications resulting from this research are briefly analyzed. Finally, the structure on which this thesis is based is described.

1.2 Motivation

The industry of audio sound and music technology consists of a wide range of diverse technologies intending to cover the needs in music recording and production and live performance. However these needs are complex and not clearly defined, and also depend on numerous parameters which are dynamic and vary as the technology evolves. Within the music technology and audio sound industry coexists nowadays classic technologies like analogue audio electronics and loudspeaker design with the state of the art technologies of digital signal processing and computing. A vast number of devices are available the market in order to meet the needs of production, sound processing and editing, recording, amplification and reproduction of musical sound at various levels of functionality, quality and cost. One of the most constant and crucial issues of concern in sound engineering is undoubtedly the efficient communication among all the above mentioned devices. As the audio systems become increasingly complex, the interconnection of the devices from which they are composed and the overall management of audio sources and signals become more complicated and problematic. The management of classic analogue sound system is based on a central control philosophy. This makes them inefficient especially when the number of sources and interconnected devices increases. This problem became more visible with the advent of digital sound. The sound systems industry was among the first who adopted the digital technology and produced digital devices. However the interconnection between these devices remained for almost two decades in the analogue domain. The causes of this phenomenon are discussed in detail in chapter two. Several attempts to use digital communication protocol for point-to-point interconnections between devices, did not manage to change the philosophy and the principle under which the sound systems are designed.

However, the evolution of data networks and the improvement of their performance regarding speed, security and reliability gave a new perspective into the sound system design. This novel way of data transfer based not in point-to-point interconnections but using a common medium where all devices have a regulated access, became very attractive among music technology vendors. In addition, the introduction of new protocols the years that followed, that allows the distribution and the synchronization of media data, over packet networks made this challenge more achievable. However the disability of the primary local area networks, to support real-time media delivery but also the lack of coordination between different research attempts, led to the development of a number of proprietary systems that are not able to interoperate between each other.

International standardization bodies like EBU and AES realized this problem and recently introduced directions in the form of standards, for the design of audio networking products in order to interoperate between each other and thus to broaden the market in this sector. The research described in this thesis was motivated from these movements. Our effort however was to investigate the possibility and the conditions under which the existing wireless networking technologies can be used for the development of wireless audio networking systems.

A careful overview of the history of network's evolution shows that the biggest growth in terms of users was always related to the migration of the networks into the wireless domain. This has an additional meaning regarding professional applications where the freedom and mobility offered by the wireless technology is significantly important.

However, current wireless networks where always considered unable to meet the requirements of real-time audio delivery and thus where always excluded from the discussion. The actual motivation of this research is to identify the causes according to which wireless networks cannot meet the criteria of real time audio delivery and further to propose solutions in order to alleviate this problem.

1.3 Scope of the thesis

The aim of the research presented in this thesis is to lay the foundation for the effective use of the wireless data network technology in professional real-time audio networking applications. However, a significant effort was made so that the proposed solutions operate as an alternative mode rather than as an independent protocol, allowing in this way the coexistence and interoperability of audio networked devices with regular data wireless networks under common infrastructures.

The main objectives are:

- Investigate existing wireless networking technologies in order to identify the most suitable technology and communication technique that can be used for the development of wireless audio networks.
- Define an efficient and widely accepted audio traffic model that can be used as a standard model for audio networking implementations and also define the specifications of such a network regarding especially the critical parameters of throughput and delay.
- Propose appropriate modifications that will allow the wireless network to meet the above mentioned parameters.
- Investigate and validate the ability of the proposed systems to interoperate and coexist within regular networks and shearing infrastructures.

1.4 Contribution to knowledge

The various chapters of the thesis aim to highlight the five contributions made towards the development of a wireless audio networking system based on existing wireless data networking technologies. Those are:

i. *Definition of a generic audio data traffic model.*

Based on published research which shows that the distribution of tempi within the global musical anthology is not uniform but it follows a normal distribution curve with the majority of tempi to be identified around the tempo of 120 bpm, a generic audio data traffic model is defined in this thesis. This model emulates efficiently the way that data production derived from the music performance. This proposed model can be used as a standard data traffic model for audio networking, allowing the comparative study between different researches attempts.

ii. *Design a collision protection mechanism for broadcasting in IEEE 802.11 ad-hoc networks, based on the extended use of CTS-to-self control message. This work presented in the 10th international conference on Wireless Communications and Mobile Computing (IWCMC 2014), held in Cyprus on August, 2014.*

Broadcasting is the most efficient way to deliver delay-sensitive data over an IEEE 802.11 ad-hoc network. However broadcasting in ad-hoc network provides no delivery guaranty. In addition, in saturated networks, broadcasting causes significantly large number of collisions. In this part of our research an extended use of the CTS-to-self control message is proposed. CTS-to-self is a regular CTS message with a destination address the address of the sender. It is originally used by the protocol as a protection mechanism when legacy technologies coexist with ERP and HT technologies. It is modified here in order to prevent collision in broadcasting by distributing channel reservation information.

iii. *Implementation of an Exclusive Backoff Number Allocation algorithm (EBNA) that eliminates collisions in broadcasting over IEEI 802.11 ad-hoc networks. This work was published in the Journal of Audio Engineering Society, in the Special Issue on Audio Networking (April, 2013).*

The majority of lost data in broadcasting over IEEE 802.11 standard are caused by collisions. Collisions are happening when two or more STAs transmit data simultaneously. The IEEE 802.11 standard implements a probabilistic collision protection technique within its medium access algorithm (MAC) called random backoff. According to this technique, each STA prior to every transmission is requested to additionally defer for a period randomly defined by a number which is selected from a set of integers called contention window (CW). However in the case of broadcasting, this CW is extremely small and when the data production and the number of STAs in the network increases the probability of two or more STA to choose the same number increases and therefore the number of collision too. In this research a novel random backoff algorithm is replacing the original one. According to this proposed algorithm, each STA is assigned a unique pair of integers. All pairs within the network are equally waited. Prior to each transmission, a STA defer for an additional time frame choosing randomly one of the two numbers from its own unique pair. Using this technique we totally avoid collision while maintaining fairness in the long run. However, this causes a linear increase of CW proportional to the number of STAs in the network which causes an increase in the overall delay.

iv. *Implementation of a Hybrid Exclusive Backoff Number Allocation algorithm (H-EBNA) that eliminates collision in broadcasting over IEEE 802.11 ad-hoc networks improving also overall transmission delay.*

The Exclusive Backoff Number Allocation algorithm mentioned in contribution (*iv*), guaranteed maximum throughput with a satisfactorily overall delay for streaming media application. However, real-time interactive applications such as professional audio networks require lower overall delay. For this reason an advanced version of the above algorithm was designed and implemented in this research. This Hybrid Exclusive Backoff Number Allocation algorithm monitors the traffic in the wireless network by keeping statistics for STAs activity. When the probability of collision is low the algorithm uses the classic IEEE 802.11 MAC which, due to its small CW, causes lower delay in broadcasting. When the probability of collision increases the algorithm use for the medium access the Exclusive Backoff Number Allocation concept implemented however only to the active STAs in the network. This results a lower increase of the CW and thus a lower overall broadcasting delay.

v. *Study for the interoperability of EBNA and H-EBNA algorithms with the conventional IEEE 802.11 wireless networks. This work presented in the Science and Information (SAI 2013) conference, held in London on October 2013 and it was selected for publication as an extended paper in the International Journal of Advanced Computer Science and Applications, (December, 2013).*

In this final contribution, the ability of the proposed EBNA and H-EBNA systems to operate in conjunction with conventional IEEE 802.11 devices within the same wireless network, and the effect of this coexistence is investigated. It is important to note that both proposed systems are using by default the broadcasting protection mechanism proposed in contribution (*i*). An audio network in implemented here within a regular wireless data network and both proposed modified medium access methods are implemented and compared with the classic IEEE 802.11 MAC. The results show that the proposed algorithms are able to interoperate with classic IEEE 802.11 networks and also the overall performance of the network increases by the use of the proposed systems.

1.5 Publications arising from this research

The work detailed in this thesis has resulted in number of refereed publications as follows:

C Chousidis, R. Nilavalan, A Floros, Enhancement of IEEE 802.11 in Handling Multiple Broadcasting Audio Data in Wireless ad-hoc Networks, Journal of the Audio Engineering Society, 61 (4), 165-173, 2013

C Chousidis, R. Nilavalan, Modifying the IEEE 802.11 MAC to improve performance of multiple broadcasting of multimedia data in wireless ad-hoc networks, International Journal of Advanced Computer Science and Applications (IJACSA), 2013

C. Chousidis, R. Nilavalan, "Improving multiple broadcasting of multimedia data traffic in wireless ad-hoc networks", Science and Information (SAI) Conference, 822 – 828, London, UK, 2013

C Chousidis, R. Nilavalan, L Lipan, "Expanding the use of CTS-to-Self mechanism for reliable broadcasting on IEEE 802.11 networks", 10th IEEE International Wireless Communication & Mobile Computing Conference (IWCMC), Nicosia, Cyprus, 2014

In addition, two more publication resulting from unpublished material from contributions (*i*) and (*iv*) are in preparation as follows:

C. Chousidis, R. Nilavalan, "A traffic adaptive MAC algorithm for reliable media broadcasting over IEEE 802.11 networks".

C. Chousidis, R. Nilavalan, "The 120 bpm traffic model: An audio traffic model for real-time, multi-broadcasting audio networking studies".

1.6 Thesis structure

The thesis spans in seven chapters which are covering the research works carried out in achieving the aim and objectives described earlier in section.

Chapter 1 gives a brief introduction into the motivations, aim and objectives behind this research and briefly highlights the contributions made to the development of wireless audio networking systems.

In Chapter 2 the basic background regarding audio networking is given. In this chapter the existing commercial systems and standards are examined and their limitations are analyzed in order to define novel research directions. The importance of wireless audio networks is also analyzed and the resultant research challenges are analytically discussed. In addition, the characteristics and the demands of future wireless audio networks are clearly defined. Finally a detailed literature review in both, wireless audio networking and reliable broadcasting in wireless Ad-Hoc networks is provided.

Chapter 3 describes the design and development of a novel collision protection mechanism for broadcasting in wireless ad-hoc networks. Initially, the background in collision protection in IEEE 802.11 networks is given. Then, a mathematical model that gives the probability of collision in a saturated wireless network is defined. Based on the above, the proposed novel protection mechanism for broadcasting is described. Finally, the Simulation characteristics and results are presented and analyzed.

Chapter 4 introduces the use of the exclusive backoff number allocation algorithm (EBNA). Initially the operation of the algorithm is described and the advantages of the algorithm, when it is used for wireless audio networking, are analyzed. In order to simulate test and compare audio networking applications, a generic audio data traffic model, based on tempo “120” is also proposed in this chapter. Finally, the simulation characteristics and the implementation of the EBNA algorithm are described and the simulation results are presented and analyzed.

Chapter 5 covers the operation of the advance version of EBNA named Hybrid-EBNA. This is a traffic adoptive algorithm that implements the EBNA concept only when it is needed and thus reduces overall delay. At the beginning of this chapter a detailed analysis of the algorithm is presented. Then, the implementation of the algorithm in C++ and in OPNET is provided. Finally, the simulation results are presented and detailed discussed.

In Chapter 6 a study for the coexistence of the EBNA and H-EBNA algorithms with the conventional IEEE 802.11 networks is presented. Initially, the objective of this study regarding interoperability between the proposed channel access methods and the classic IEEE 802.11 MAC is analyzed. Then, the physical and the simulation characteristics of the test network are described in detail. Finally, the simulation results are presented and analyzed.

Finally, Chapter 8 summarizes and concludes this research work and provides insight into future development of wireless audio networking systems.

Chapter 2

Background in Audio Networking:

(Introduction to Sound Systems, Fundamental of Audio Networks, Wireless Audio Networking and Literature Review)

2.1 Introduction

With the advent of digital audio, the need for a reliable and flexible distribution method within professional sound systems becomes increasingly intense. In today's conventional analogue systems, digital audio has to be converted into analogue in order to be distributed and processed. This causes a significant reduction in quality and fidelity of sound. The use of data packet networking as a primary technology in distributing digital audio in professional systems is been implemented during the last decade. Also, several standards have been introduced from international standardization bodies. In this chapter the existing commercial systems, and standards are examined and their limitations are analysed in order to define novel research directions. The use of wireless audio systems is proposed and the resultant research challenges are analytically discussed. In addition, a literature review in the recent research within all related areas is also provided.

2.2 Fundamentals of Conventional Analogue Audio Systems

Most sound systems fall into one of the three following basic functional classes: sound reinforcement, studio recording and sound reproduction systems [1]. In sound reproduction systems, the operation of the system is limited to play back pre-recorded audio signals. Typical examples of such applications are night clubs, dancing halls etc. Since the program material is recorded and mixed in the studio, any manipulation of the signal is usually limited to level control and basic frequency equalization. This operation may be handled by a small control units and the complexity of installation is usually low.

In sound reinforcement, editing, mixing and amplification of the live sound sources is applied in order for them to reach a large audience. In studio recording systems a similar operation is taking place but in this case the audio sources are independently directed to the recording devices in order to be available for future editing and mixing. In addition, the audience is limited to a significantly small group. Reinforcement and studio applications vary in complexity from relatively simple setups like conference rooms, theatres and home studios, to large-scale installations. Their complexity increases, when they also have to facilitate the monitoring needs of the performers. Thus, complicated audio loops have to be created and additional processing and control units have to be added. In order to understand the operation of a sound system, we can divide it into a set of subsystems. The most common categories of these subsystems are described below.

i. **Sound Sources:** This is the most visible part of a sound system and it is consisted mainly from the instrument and the vocals. The instruments can be "acoustic" or "electronic". No matter if they produce analogue or digital sound, in most conventional audio systems the signal is sent in an analogue balanced form in order to avoid addition of noise [1]. The signals have to travel considerably long distances, considering their level, in order to reach the Mixing and Control unit (fig 2.1). The number of the sound sources is significantly big and far exceeds the number of the musicians as many instruments need more than one audio line (multiple microphones, stereo signals, etc.). The signals are usually traveling through bulky multicore cables, always facing the problem of interference and crosstalk. As long as the system spreads along the entire premises and all electronic devices are grounded, if no proper grounding installation exists, the long cables are forming "ground loops" which act as antennas for the harmonics of the 50 Hz which are added eventually in the audio signal.

ii. **Mixing and Control Units:** This part of a sound system is usually consisted from one device. It is usually referred as "Mixing Console". This receives the independent incoming signals, adjusts their levels, processes their frequency spectrum and mixes them into several groups in order to send them to the rest parts of the sound system as it shown in figures 1 and 2. If it is a digital console, an A/D converter operate in each input and a D/A converter in each output. Digital communication is also available in some models mainly between console and audio processing units but in most cases the interconnections are kept in analogue form. There are several differences between studio and live performance mixing consoles but the basic idea of connecting and operating is similar.

iii. **Audio Processing Units:** This is a group of devices which is usually located next to the mixing console. The number of devices varies depending on the system and in advanced sound systems can be significantly large. There are several types of devices used in modern system for audio processing, some of them are, Reverb Emulators, Equalizers, Noise Gates, Limiters, Auto-tune Processors, Harmonizers, etc. The interconnection uses a send-return philosophy as it is shown in figures 1 and 2. Each unit can serve one single signal or a specific mix of signals.

iv. **Main PA System:** This is the public address (PA) system which is usually consisted from the amplifiers and the loudspeakers. It can be a simple system with one stereo amplifier and two speakers or a sophisticated system with a group of amplifiers, arrays of loudspeakers and also dedicate low frequency speakers. In all cases a main stereo mix is connected to this subsystem (fig 2.1).

v. **Monitoring Devices:** This is a group of devices which is addressed to the performers. In the case of live music, it consists from ground speakers located in front of each musician or singer, or from in-ear headphones which can be wired or wireless. In the case of a recording studio installation, it consist of close type headphones and one or more of “near field” monitor loudspeakers addressed to the mixing engineer (fig 2.2).

vi. **Recording Unit:** It is found mainly in studio installations but can also be found in live music installations when live recording is needed. Is usually referred as multi-channel or multi-track recorder. There are several technologies and systems used today starting from analogue recording systems, digital tape recording systems (ADAT, DTRS), hard disk recorders and software recorders with or without their dedicated hardware (Pro Tools, Qubase, Logic Audio, Nuendo etc.). In all implementations the interconnection idea is again the same. All incoming sources are directed into the recording units and then return in to the mixing console. Thus, in the mixing console both the direct and the play back signal can be available (fig 2.2).

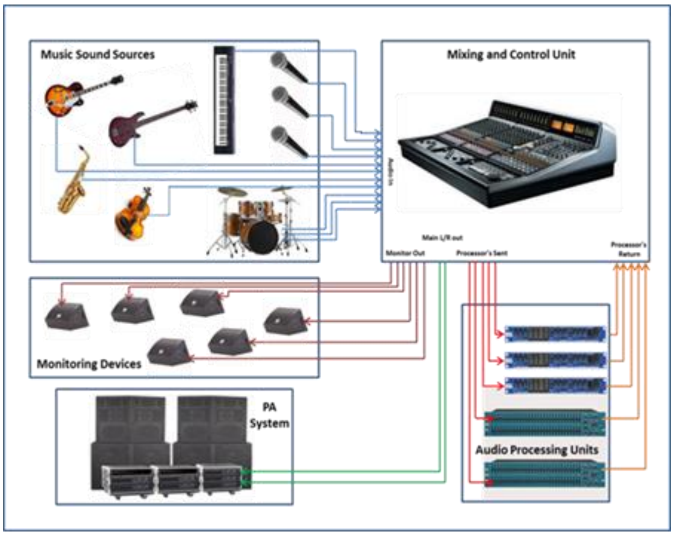

Fig 2.1: Live Concert sound system example

The signals created during the performance are directed to the console. There, initially a level and spectrum adjustment is taking place. In case of live concert, the signals in groups or individually are sent to the processors section and after being processed they are returning back to the console. Then, various groups are created and directed either to the audience through the PA subsystem or to the performers through the monitoring subsystem.

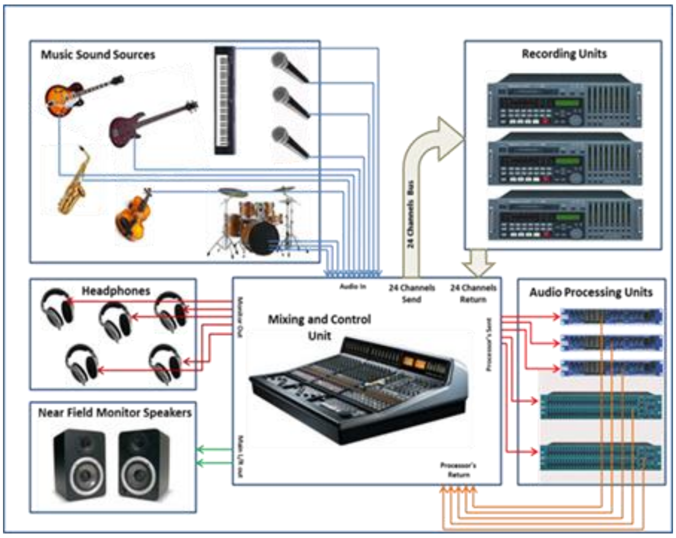

Fig 2.2: Recording Studio sound system example

In case of recording studio, the signals are directed through the console to the recording units and also in different groups and combinations, to the performer's headphones and the main monitoring speakers. Here, during the recording process, play back and live signals can be combined, grouped, processed and monitored, simultaneously according to our needs.

The above is a generic approach for the design and implementation of sound systems. Alternative design techniques apply in all subsystems in order to resolve specific problems and cover specific needs, because every system is unique and have its own demands. The conclusion is that all the signals created by the sound sources have to be distributed in different parts of the system, directly or in various groups, processed or unprocessed. This distribution is based on a centralized control which makes the design and the maintenance of the system complicated, as it is directly affected by various parameters like the number of musicians, the site of installation, the style of music etc. Changes in topology are always problematic and time consuming and the identification and repair of faults is difficult. In addition, the analog signals are exposed to all kind of interference and crosstalk. Numerous gain adjustments and A/D and D/A conversions reduce the audio quality adding noise and harmonic distortion. These are the reasons that a decade now there is an intense effort for the sound systems, to migrate in a more efficient and functional technology and design. It is now evident that the emerging technology in sound systems design will be based on the digital audio, which will be transferred within the system using one or more types of packet networking technologies. As it is shown from the above, devices in a sound system are forming a sort of network, exchanging information and sharing resources. Today's networking technologies are offering a fast, easy, efficient, and inexpensive background on which reliable digital audio interconnection can be implemented.

2.3 Introduction to Audio Networking

The term "Audio Networks" is introduced to describe the use of a network infrastructure in the distribution of real time digital audio. Audio Networks are not dealing with IP telephony or VoIP applications. They are mainly intended to professional audio applications like stage and studio sound and also large scale sound reinforcement systems like stadiums, airports, convention centres, radio and television stations etc. Thus, the main objectives of these networks are audio fidelity and low latency. The most common networking infrastructure used by audio networks is Ethernet-based networks. For that reason the term "Audio over Ethernet" (AoE) is also used. In the case of a non-proprietary network layer [2], where IP routing protocol is used, Audio Networks are also described by the term "Audio over IP" (AoIP) [3]. Audio networks are designed to alleviate installation, maintenance and administrating problems facing the conventional analogue systems and to ensure a reliable and widely accepted way of digital audio distribution. However, Audio Networks it not something new. Is been more than fifteen years, since the idea of using data networks for high fidelity audio distribution started to appear in practical implementations from different vendors, and also various standards to be proposed from international organizations. As it is usually happening with all new technologies that have commercial interest, the lack of common practises and standards in audio networking has resulted many proprietary or under royalties systems to appear, without the ability to interoperate between each other. Just a month before this text was written, the Technical Committee on Network Audio Systems (TC-NAS) of the Audio Engineering Society (AES) released a standard with the title "*Audio-over-IP interoperability*" in order to define audio network's specifications and to give development directions to manufacturers.

A brief description of the most important existing systems and protocols in the field of Audio Networking is given below.

2.4 Existing Networking Audio Systems in the market

2.4.1 Overview

The advantage of distributing audio using Local Area Networks (LANs) aroused the interest of the commercial audio system manufacturers. Thus, a series of protocols and products have been designed to cover this field. Some of them are totally closed systems that have to be purchased by a specific vendor. Other protocols are open to designers and implementers, by paying royalties.

Protocols can be broadly categorized into three layers or generations: Layer 1, Layer 2 and Layer 3 [4]. Layer 1 protocols use Ethernet wiring and signalling components but they are not using the Ethernet frame structure. Layer 1 protocols often use their own media access control (MAC) algorithm, rather than the one is used by Ethernet. Layer 2 protocols encapsulate audio data in standard Ethernet packets. Most of them can make use of standard Ethernet hubs and switches, though some other require the network, or at least a “virtual LAN” to be dedicated to the audio distribution application. Layer 3 protocols encapsulate audio data in standard IP packets (usually UDP/IP or RTP/UDP/IP). Use of the IP protocol improves interoperability with standard computing platforms and in some cases, improves scalability of the audio distribution system. The most well-known systems existing today are described below:

2.4.2 EtherSound by Digigram

EtherSound is an open (under licence) standard for networking digital audio. It is available to audio manufacturers via a licensing program, depending on the level of application. It is compliant with the IEEE 802.3 Ethernet standards in the physical layer but its MAC implement a token-like algorithm excluding any kind of regular LAN operation. It can only support two way communications only when connected in a daisy-chain topology. This can be implemented in different configurations like regular daisy-chain, combined daisy-chain and star, and redundant ring configuration. EtherSound provides significantly low latency which is also stable and deterministic [5]. As an example, it can deliver up to 64 channels of 48 kHz, 48 bit uncompressed audio with a latency of 0.125 msec. The basic technological characteristics are listed in table 2.1 [6].

Specifications	EtherSound
Bandwidth requirements:	100 Mbps dedicated Ethernet network
Audio format:	24 bit PCM
Sampling Frequency:	44.1 to 96 kHz
Network architecture:	Daisy-Chain, Star or any combination of both
Audio clock:	Isochronal transmission. All devices are synchronized from the audio clock of the first device on the network
Channel number:	up to 64 in each direction

Table 2.1: EtherSound main technical characteristics

Manufacturers who wish to integrate EtherSound connectivity to their devices must license the technology from Digigram.

2.4.3 CobraNet by Cirrus Logic

CobraNet is a combination of software and hardware products, designed to deliver in real time uncompressed, multi-channel digital audio over a standard Ethernet network. The protocol works with its specific hardware which encodes and decodes the CobraNet signal. It also has embedded A/D and D/A converters, so any regular analogue source can be connected directly. The system transfer data using data link layer packets and not the TCP/IP packets. This makes data transmitted with lower delay but it cannot travel through routers. Consequently, the networked audio system is limited within a LAN. CobraNet network cards come in several varieties, some of which can support more audio channels than others. The entire network is synchronized to a single CobraNet device known as ‘the conductor’ while the other devices are called performers. CobraNet is organized in into ‘channels’ and ‘bundles’. A typical signal can contain up to 4 bundles of audio. Each bundle can contain up to 8 audio channels of 48 kHz 20 bit for a total capacity of 64 channels. A dedicated network is needed for such an application. Manufacturers who wish to integrate CobraNet connectivity into their devices must license the technology from Cirrus Logic [7] [8].

2.4.4 Aviom A-NET Pro16 & Pro64

Aviom A-Net is a completely proprietary audio distribution and networking technology. It is based on the physical layer of Ethernet but it uses a totally dedicated software and hardware [9]. The production company supports two versions of A-Net and offers two product lines based on those technologies, the Pro16 and the Pro64 series. Pro16 offers a monitor mixing system which is well accepted among musicians and sound engineers. Proprietary personal network mixers can be networked with a distributor. This allows each performer to create a customized monitor mix on stage or in the studio. Some of the most important technical characteristics of the system are listed in table 2.2

Specifications	Pro16	Pro64
Maximum number of channels:	64, using AN-16SBR System Bridge 32, using A-Net Expansion jacks	64 in Auto Mode 64x64 in Manual Mode
Sample rates:	48kHz	44.1/48kHz, 88.2/96kHz, 176.4/192kHz
Resolution:	24 bits	24 bits
Supported audio formats:	Analogue, Digital, from compatible consoles	Digital, from compatible consoles,AES3 digital
Max. Cat-5e cable length	500ft/150m	400ft/120m
Connection topologies:	Daisy chain, star, or combination	Daisy chain, star, or combination
Latency:	<800μs, analogue input to analogue output	<800μs, analogue input to analogue output

Table 2.2: Aviom main technical characteristics

2.4.5 Audiorail

This system is also based on Ethernet network but the format of the data and the way it is encapsulated and transferred is incompatible with it. A Time Division Multiplexing (TDM) technique is used in data transmission, instead. Latency is estimated under 10 μsec [10]. Therefore, it is a totally proprietary system having also its own hardware. It is implemented in a daisy-chain topology and it is actually a digital multichannel audio distribution system instead of multichannel audio networking system. Table 2.3 shows the basic characteristics of this system.

Specifications	Audiorail
Maximum number of channels:	22
Sample rates:	48kHz (22 channels), 96kHz (16channels)
Connection topologies:	Daisy chain
Max. Cat-5e cable length	100m, (multi-mode fibre 2Km)
Latency:	4.4 μsec (plus 0.25μsec per hop)

Table 2.3: Aviom main technical characteristics

2.4.6 MaGIC by Gibson

MaGIC comes from Media Accelerated Global Information Carrier. It provides up to 32 channels of 32 bits bidirectional high fidelity media distribution with sampling rate up to 192 kHz. It is an open architecture standard that allows products from different vendors to communicate. It is based on the 100 Mbit Ethernet, physical. The protocol was originally developed at University of California, at Berkley Centre for New Music and Audio Technologies. The research project was sponsored by Gibson Guitar Company which is not having royalties or any commercial use [11]. It has proprietary timing synchronization mechanism, data transport layer. Data is transmitted between devices at fixed network sample rate in discrete fixed-size frames. Each frame contains a preamble and 188 octet payload. At the default sampling rate of 48 kHz the useful bandwidth is 72.2 Mbit/s. The payload field has two parts: media and control, which are managed independently in order to achieve better real time functionality.

Specifications	MaGIC by Gibson
Maximum number of channels:	32
Sample rates:	192 kHz
Resolution:	32 bit
Connection topologies:	Daisy chain, Star , and a combination of them
Max. Cat-5e cable length	100m
Latency:	10-40 μsec (for a 100baseT physical)

Table 2.4: MaGIC main technical characteristics

Media payload has a size of 1024 bytes and is reserved for low-latency synchronous audio and video data. Control payload has a size of 352 bytes is reserved for MaGIC control messages, MIDI messages, and other protocols. Table 2.4 shows the basic technical characteristics of the protocol.

2.4.7 Dante

Dante is a global networking standard including Internet Protocol (IP) with true IP routing and not only using Ethernet infrastructure. Dante works as an IT network with no limits on layout options. It is a combination of software, hardware and network protocols which deliver uncompressed, multi-channel audio with low latency over a standard Ethernet network. The advantages comparing Dante to first generation audio over Ethernet systems, such as CobraNet and EtherSound is that Dante have the ability to pass through network routers, have higher channel count and automatic configuration [12]. The basic technical characteristics of the system are listed in table 2.5. It is a proprietary system that has the advantage of transferring large number of audio channels with considerably good characteristics and fully IP functionality. It has been adopted from many professional audio equipment brands.

Specifications	Dante
Maximum number of channels:	1024 (per link)
Sample rates:	192 kHz
Resolution:	24 bit
Connection topologies:	Any IP
Routable/Switchable:	Yes
Max. Cat-5e cable length:	100m
Latency:	83.3 μsec (minimum)

Table 2.5: Dante main technical characteristics

2.4.8 HyperMAC & SuperMAC

SuperMAC is a proprietary point-to-point digital audio connection operating over 100Mbir/s Ethernet, based on AES50 protocol. A deterministic protocol is used here that is essentially a time division multiplex (TDM) at the hardware level. It offers a maximum of 48 bidirectional channels with an individual link latency of 62.5 μsec.

HyperMAC on the other hand it is operating over Gigabit Ethernet. It works on the same technical Specifications but offers 192 bidirectional channels with an individual link latency of 41.66 msec.

HyperMAC and SuperMAC are originally developed by Sony Pro-Audio Lab in Oxford and now owned by Klark Teknik, It is available to the audio industry through specialist third party developers (Auvi Tran & ZP Engineering). The actual HyperMAC and SuperMAC core remains the intellectual property of Klark Teknik and is supplied as encrypted ‘black box’ [13]. Table 2.6 shows some of the main technical characteristics.

Specifications	HyperMAC	SuperMAC
Maximum number of channels:	192	48
Sample rates:	96kHz	48kHz/44.1kHz
Resolution:	24 bit	24 bit
Connection topologies:	Point - to - Point	Point - to - Point
Max. Cat-5e cable length:	100m (as specified by IEEE 802.3)	100m
Latency:	41.66 μsec	68.02μsec/62.5μsec

Table 2.6: Dante main technical characteristics

2.4.9 Livewire

Livewire is also a proprietary audio networking system which is fully compatible with IP Ethernet networks. It is created by Axia Audio, a division of Telos Systems. However, is available as an open standard through the Axia's partner program. It uses RTP/IP packets to distribute PCM uncompressed audio. Proprietary hardware is used to encode analogue audio. No dedicated network is needed. Regular traffic like file shearing browsing etc. can coexist with live wire packets. Fixed length small size packets of 72 bytes of audio data are used to minimize buffering time and decrease delay. These streams are called “Livesteams” and they need dedicated hardware. Livewire use an alternative data packetizing technique called “Standard Streams” with significantly large number of audio data bytes encapsulated. This is usually play back audio where delay it is not significantly important.

Livewire is one of the most widespread systems for AoIP, offering a full set of products and support software. It is dominating the market when it comes to networking recording studios and radio broadcasting studios. Table 2.7 shows the main technical characteristics of the system.

Specifications	Livewire
Maximum number of channels:	26@100 Mbps link 260@1000 Mbps link
Sample rates:	48 kHz
Resolution:	24 bit
Connection topologies:	Any IP/Ethernet
Routable/Switchable:	Yes
Max. Cat-5e cable length:	100m
Latency:	0.75 msec (minimum)

Table 2.7: Livewire main technical characteristics

2.5 An overview of digital audio transfer standards

During the last decade a revolution has taken place in the field of digital audio. As the computational power increased and its cost decreased, digital audio replaced gradually the analogue audio signals in all possible areas. This raised consequently the need for reliable digital audio transfer. As it is usually happening with emerging technologies, every vendor tried to design and imposes its own protocol in order to gain a bigger share of the market.

Audio Engineering Society (AES) is global organization dealing with all subjects in the audio engineering sector. Since 1998, in a meeting in the White House with subject "advanced networks for music and audio", their representatives pointed out the importance of networks and the Internet for the future development of professional audio and emphasized the need of proper standardization for audio networking [14] [15]. AES has establish a Technical Committee in Network Audio Systems (TC-NAS) where engineers from all over the world from the academia and industry, are collaborating in order to inform the AES and other organizations on issues related to the committee's focus area, with an emphasis on advanced wide-area networks . AES has proposed a series of standards to cover a number of cases where digital audio transfer is needed. These protocols are listed and briefly analysed below.

2.5.1 AES-3

It is an AES standard for digital audio, digital input-output interfacing Serial transmission format for two-channel linearly-represented digital audio data. It is first issued in early 1985. Final revision released in 2009. It is a Point-to-Point audio data transmission of 2-channel PCM audio with a wide range of sampling frequencies over twisted pair microphone cable (110Ω), using balanced transmission. Some provisions are made in this standard for adapting the balanced terminals to use 75 Ohm coaxial cable and transmission by fibre-optic. The standard is adopted by some commercial applications and it put the base for all AES standards [16].

2.5.2 AES-10 (MADI)

The AES standard for a Serial Multi-Channel Audio Digital Interface (MADI) was first issued in 1991. The final revision was AES-10-2008. It provides the specification for the serial digital transmission of 32, 56 or 64 channel of linearly represented digital audio data. The sampling frequencies could have the range of 23 kHz to 96 kHz and resolution of up to 24 bits per channel. The transmission could be over a single 75Ω coaxial cable or fibre optic cable. The optical fibre cable can provide a range of up to 2km. The protocol specifies an independent master Synchronization Signal [17]. The basic data rate is 100 Mbit/s of data using 4B5B encoding to produce a 125 MHz physical baud rate. This clock is not synchronized to the audio sample rate, and the audio data payload is padded using "JK" sync symbols. The audio payload is identical to the AES3 payload but with more channels. MADI is widely used in the audio industry. Its advantages over other audio digital interface protocols and standards such as AES/EBU (AES3), ADAT (Alesis Digital Audio Tape), TDIF (Tascam Digital Interface) and S/PDIF (Sony/Philips Digital Interconnect Format) are: first, support of a greater number of channels per line which makes it suitable for networking audio and second, the use of coaxial and optical fibre media that enable the transmission of audio signals over 100 meters and up to 2000 meters.

2.5.3 AES-47

Is a standard for digital input-output interfacing transmission of digital audio over asynchronous transfer mode (ATM) networks [18]. It was originally published by AES on 2005. This protocol describes means for the transmission of professional multi-channel audio in linear PCM or AES3 format, across digital networks including metropolitan and WAN, in order to increase performance with regards to latency and jitter. ATM had been chosen as it provided a more efficient service than ISDN and significantly better performance than IP. Because of ATM characteristics (constant length packets and small overhead), a constant transfer rate of audio can be achieved, although a packet transfer method is used. It may be used directly between specialised audio devices or in combination with telecommunication and computer equipment with suitable network interfaces and utilizes the same physical structured cabling used as standard by those networks.

2.5.4 AES-50 (HRMAI)

It is an AES standard for High Resolution Multi-channel Audio Interconnection. It provides a bi-directional, point-to-point connection for up to 48 channels of digital audio in a variety of formats. The link uses a single Category 5 structured-wiring data cable, and is designed for operation in a studio environment [19]. Audio synchronization is maintained by transmitting a 64fs (for example 2.8224 MHz if fs = 44.1 kHz) audio clock signal in parallel with the audio data, utilising the extra signal pairs on a structured wiring data cable. The physical layer is used is Ethernet due to its high bit rate, its robust error performance and its ease in implementation, using standard components.

2.5.5 AES-67 (Standard for audio applications of networks - High-performance streaming audio-over-IP interoperability)

Although many commercial audio networking systems nowadays are using the widely accepted TCP/IP model, interoperability between them is not yet achieved. This is because each implementation uses different techniques when it comes to packetizing of audio data, fragmentation, transmission control, synchronization, quality of service (QoS) and addressing, from a variety of methods offered by the TCP/IP suit. An Audio Engineering Society standards task group called SC-02-12-H has been formed to develop an interoperability standard for high-performance professional digital audio IP networking. This project has been designated AES-X192 and is partially inspired by an EBU initiative called N/ACIP, which published interoperability recommendations for audio over wide-area IP networks. The X192 project committee completed their work in April 2013, and submitted their findings to the AES. The work of the AES subcommittee for an AES standard on audio network, (called X192) was given the official AES number, AES67 and it was released on 11th September 2013, just few months before this thesis was written. This standard aims to propose a common practise in the use of TCP/IP model in audio networking implementations. It precisely defines all methods and techniques that should be used across different layers of the TCP/IP stack in order to achieve high-performance media networking, for professional quality audio (PCM coding, 16 bit, 44.1 kHz and higher) with low latency (less than 10ms) [20].

The standard supports three sampling frequencies: 44.1 kHz, 48 kHz and 96 kHz. It defines the operation for layer 3 (network layer) and layer 4 (transport layer), while interoperability in lower layers is based on Ethernet's LLC, MAC and physical. Fragmentation and reassembly of data packets is not supported. Real-time Transport Protocol (RTP) in conjunction with RTP Control Protocol (RTCP) is been used. Finally UDP/IP transport protocol is used for RTP media and RTCP management packets. QoS issues are also covered by this standard. The standard proposes the classification of the data traffic in three categories which according to their importance are, Clock, Media and Best effort. Clock traffic is a low-frequency transmission (less than 100 packets/sec), which is highly sensitive to latency and specifically to latency variation. This type of traffic is getting the highest priority. Media traffic demands a high bandwidth (over 1 Mbps per audio channel) and a packet size varying from 100 bytes to 1500 bytes (maximum Ethernet MTU size). With UDP in use, lost and delayed packets cannot be retransmitted. For this type of traffic is assigned a high priority. Best-effort traffic constitutes any traffic related to this standard which is not classified as clock or media traffic and therefore it is assigned the lowest priority.

2.5.6 EBU Doc 3326-2007

The European Broadcasting Union (EBU) also realized the need for interoperability between AoIP systems. The EBU is the world's foremost alliance of public service media organizations, with Members in 55 countries in Europe and beyond . EBU has established the N/ACIP project group which worked to create a standard for interoperability named, “Audio Contribution over IP”. This standard was published as EBU Doc 3326-2007. The standards is focused mostly in audio streaming rather than live audio reinforcement or studio applications. Thus, audio compression codecs are in use. Several manufacturers implemented the standard and a plugfest held successfully in February 2008.

The interoperability it is based on the use of RTP over UDP transport protocol and SIP for signalling. As mentioned earlier, the standard it is not addressed to real time applications, therefore audio compression codecs can be used. The mandatory codec formats are, G.711, G.722, MPEG Layer II and linear PCM of uncompressed audio. There are also a set of additional codecs recommended by the standard. Those are, MPEG Layer III, MPEG-4 AAC and MPEG-4 AAC-LD. The standard recommends the avoidance of fragmentation for large packets but it does not forbid it. QoS issues and delay limits in all processes like audio encoding/decoding, packetization, and network latency are also defined [21] [22].

2.6 Drawbacks and problems in the deployment of audio networking

A careful reading of the above discussed in this chapter shows the effort of manufacturers and organizations, to establish and promote audio networking during the last fifteen years. This is because networked audio systems allow much more flexibility than conventional analogue systems. In a networked audio system, any device connected to the network is able to communicate with any other with significant simplicity in connections and routing. However, audio networks raise a number of issues that affecting audio but it is not important for data communication, for which the current networking technology is designed. As it is shown from the existing standards and implementations, it is difficult to define the optimum networking characteristics, as different domains of audio applications have different needs and demands that are often conflicted between each other. Differences between various implementations are raising issues on networking topologies, range of the network, the number of supported channels, audio formats and end-to-end delay. Even in cases where an existing networking system totally meets the needs of an application, the lack of interoperability makes the transition from the conventional to the networked audio system to be viewed with scepticism. Being limited to one specific vendor it was never a choice for sound system designers who are used to combine products from various manufacturers in order to keep a balance between quality and cost. The ability to host in the same device different standards and protocols is technically possible, especially if the physical part of the network remain the same. The main issue in this case is related to licensing. In order for a device to be compatible with many standards and proprietary protocols it needs to apply to different licensing schemes. This consequently increases the overall manufacturing cost of the device. Another issue that holds back sound system designers from migrating to networking solutions is yet the need of a physical network installation. The problem here is that the majority of sound system engineers are not familiar with networking infrastructures. Networking technologies are having not only a different philosophy and terminology but they are dealing with a totally different technology than conventional analogue audio systems.

Concluding, a careful research in the professional audio market shows that audio networking systems hold only a very small share of the total market. This appears to be significantly disproportionate to the general expansion of the IP applications in all technological fields and implies that crucial improvements have to be done in this sector.

2.7 Wireless Audio Networking

Wireless audio networking is characterized as the “holy grail” in the audio networking case [23]. Its importance is related to two main factors; mobility and network infrastructure.

Mobility was always a very important issue in performing arts. Numerous wireless systems for transferring audio are existing in the market using in most cases a point-to-point radio link. In those systems, the audio signal is modulated and transmitted over the link in a digital or analogue form depending on the implementation. For each audio source a separate radio channel is used. This system works well when the number of wireless sources is kept small but still, routing, mixing, processing and resources sharing is done with the signal to be in a wired and analogue form. In addition, when the number of wireless sources increases, the implementation becomes dysfunctional and the interference between channels reduces audio quality. The idea of a wireless audio network is significantly important because it will add one more advantage in the audio networking systems. That is mobility. Musicians, singers and performers generally will be able to move during the performance without the limitations of cables and pre-designed infrastructures enjoying at the same time the advantages of a networked audio system. Roaming techniques can also be used to enhance performance mobility.

Network infrastructure is according to the author’s opinion, one of the most significant obstacles in the evolution of audio networking. Sound systems engineers are not familiar with networking practices and terminology. This is why all audio networking vendors are putting special effort in publishing white papers, technical notes and even books that introduces data networking to sound systems engineers [3].It is evident that a new specialty in the field of sound system engineering is created nowadays. The need to design and implement a network infrastructure in a stage or studio discourages sound engineers as it still creates limitations when it comes to the topology and it is time consuming especially in tour sound implementations. This is a huge challenge for wireless audio networking as it is capable to resolve the above issues and boost the evolution of audio networking in general. This is not an extreme assumption if one considers the impetus that was given to the evolution of network technologies and the internet by the development of wireless networking technologies in LAN and 3G level.

2.8 Literature review on wireless audio networking

The discussion over the subject has started since the mid-2000s without significant research achievements. Current wireless networking technologies at common consent are characterized unreliable for real-time, high quality audio delivery for multiple audio channel applications.

In the white paper released by the AES’s TC-NAS on 2009 with the title “Best Practices in Networked Audio”, it is clearly mentioned that digital audio networking will be benefited from the adoption of wireless networking technologies [24]. However, the existing technologies are considered unable to substantially support such an implementation. For a small physical scale of networking the TC-NAS highlights three main wireless technologies as the most appropriate to carry out a wireless audio network. Those are: Bluetooth, WiFi and WiMAX. Bluetooth is more oriented to compressed audio applications. This limits audio quality and in combination with Bluetooth’s range limitations, makes this technology unsuitable for professional audio applications. WiMAX has all the necessary characteristics like bandwidth and transmission range, to support multi-channel high quality networks but their adoption is not wide by the consumer electronic vendors. That consequently raises the cost of such implementation. The most promising wireless networking technology according to TC-NAS, for professional audio application is the IEEE 802.11 (WiFi). It is widely adopted by the most commercial electronic and computer products, it is reliable and it has the appropriate bandwidth to deliver many uncompressed audio channels. However, audio delivery over WiFi introduces a number of issues mostly related to the transmission range, congestion control mechanism and electromagnetic interference that affecting delay and throughput.

It is clear that TC-NAS's comments are referring to the existing networking technologies. A very much promising wireless networking technology for the implementation of professional audio networks in the future is the IEEE 802.15.3 Wireless Personal Aria Networks standard (WPAN) [25]. The 802.15.3 is a MAC and PHY standard for high-rate (11-55 Mbit/sec) covering at least 10 meters in all directions. An amendment called 802.15.3c was published on September 11, 2009. This amendment specifies a millimeter-wave-based (mmWave) alternative physical layer for the existing 802.15.3. This WPAN operates in the 57-64 GHz unlicensed band and allows high coexistence with all other microwave systems and a very high data rate over 2 Gbit/s. It is able to support applications such as high speed internet, video on demand, HDTV, home theater and generally, real time streaming and wireless data bus and it will be able to replace cables. This is a media oriented wireless networking system which is not yet implemented and thus cannot be used in wireless audio networking systems yet.

The majority of the research efforts during the past years are focused in the use of the IEEE 802.11 standard to support audio delivery from one source to one or multiple destinations. This type of application is mostly addressed to home theatre and Hi-Fi audio delivery. In many cases audio signal is in a compressed form something that is appropriate for streaming but not for real time live music applications. A realistic live musical performance scenario, where many sources are exchanging audio data between each other simultaneously, using a wireless networking infrastructure has not yet been examined. In [26], the authors implement a point-to-point transmission over 802.11b. They are aiming to use this technology to replace conventional FM transmitters that are used for connection between audio sources and consoles, amplifiers etc. The results shows acceptable latency (approximately 12 ms) but this system has a limited field of application and not in any case constitutes a network. Seppo Nikkilä et al in [27] examine the possibility of streaming multichannel high fidelity audio over 802.11 networks. Their research is addressed mainly to home theatre applications (Blu-Ray systems with 8 audio channels, 32 bit resolution and 192 Ks/sec sampling rate). The proposed system uses a centralized MAC based on the Point

Coordination Function (PCF) of the 802.11n standard. They implement and test their system and they prove that the use of IEEE 802.11 standard is a feasible and attractive method for the wireless distribution of high quality multichannel audio data. However, their system is more an audio distribution rather an audio networking system as it supports one audio source and multiple independent receivers. In addition, the use of PCF is considered with skepticism as it has never been implemented by vendors in any of their products. In [28] the authors are investigating the quality of audio transmission over an ad-hoc network using routing protocols. In their research they measure the important parameters that affect audio delivery such as latency, jitter and packet loss and their impact on perceived audio quality. The experiment tests one-hop and two-hop delivery in near, far, static and mobile configurations. It is again a point-to-point audio delivery scenario using compressed audio but the results show a significant finding. The two-hop scenario results an average packet loss of 32% which is unacceptable for audio. This means that a future wireless audio networking implementation must be based on one-hop transmission, excluding the use of routing protocols. An analysis and evaluation of real time audio playback over the 802.11 wireless technology is performed in [29]. The paper introduces a mathematical audio playback analysis and uses simulation to emulate the wireless network behavior. A PCM uncompressed audio packet (16 bit, 24Ks/sec) is transmitted through the simulated network, and then the packet is reconstructed and a distortion analysis is performed comparing in real time the incoming and outcoming audio. The audibility of the distortion introduced is confirmed through listening tests. The results show that audio quality can be within acceptable limits especially when the packet size varies in medium size values (approx. 880 bytes). Kevin Curran in [30] deals with the packet loss recovery in music streaming over bandwidth constrained networks. He implemented an intelligent algorithm that identify the parts and the structure of a song which admittedly they are repeating in well-known patterns and replaces the lost parts, if any, with similar by using extensive buffering techniques. This method can improve the quality of the final audio but it is limited to song streaming. In addition, due to the extensive buffering significant latency is added. The work in [31] examines the possibility of improving streaming performance in 802.11g networks by varying the Packetization Interval (PI). This parameter defines the duration of the audio in each transmitted packet. It is one of the parameters used by RTP protocol which is the dominant protocol in audio streaming applications. The optimum range of PI is defined taking in to account a delay threshold and the number of simultaneously connected recipients. The audio data are transmitted in compressed mode using the Advance Audio Coding (AAC). The level of resulting delay is approximately 400ms, suitable for digital radio streaming but not for real time music applications.

The quality of audio, delivered over wireless LANs is a significant research topic. Most of the research that has been done is in the field of Voice over IP (VoIP), which is the most commercial application. The tests and experiments uses compressed audio rather than uncompressed PCM and they are focused on the human voice spectrum. Measuring techniques of audio quality are based in both subjective and objective methods. The International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) in its P.800 recommendation defines several subjective tests based on human rating [32]. In these tests the mean average of the results gives a Mean Opinion Score (MOS). MOS can also be approximately calculated using objective methods such as the E-Model [33] [34] or the Perceptual Evaluation of Speech Quality (PESQ) [35]. Further discussion and analysis of these methods is beyond the scope of this thesis but we consider them as important tools for the evaluation of the quality of the audio delivered over networks.

Real time scenarios of audio delivery require the use of encryption which consequently adds overhead and affect audio quality. This has been the subject of several research projects. In [36] the authors test an 802.11g WLAN and show that the use of WPA2 decreases throughput by 5%. In [37] the impact in audio delivery by using AES and 3DES tunnel mode encryption is investigated. The 802.11b is used as networking technology and the investigation is based on ITU-T's E-Model. It is shown that AES encryption gives better results when it comes to MOS scaling. Finally in [38] a more integrated study between 802.11b, 802.11g and 802.11n is performed concerning audio quality. Objective and subjective evaluation methods are used for this study. The results firstly verify the ability of E-Model to successfully predict MOS and secondly show that even using the latest 802.11n standard it is not likely to achieve MOS greater than 3.5, when the theoretical maximum is 5, according to the G.107, ITU-T recommendation.

2.9 Characteristics and demands of a Wireless Audio Network

From what we have examined until now in this chapter it is understood that the field of audio networking could be significantly benefited by expanding in the wireless domain. On the other hand, it is clear that the existing technology it is not capable to support this expansion. In order to create a "research path" and to examine which changes, modifications and improvements can be done, we must first clearly define the fundamental characteristics and demands of a future wireless audio network suitable for real-time professional applications and capable of supporting multiple active audio sources.

According to the author's opinion such a network must be based on an existing technology which however should be dynamic and widely accepted by the industry. This technology is admittedly the IEEE 802.11 standard. This is a technology that is constantly evolving and improving keeping its philosophy and characteristics while maintaining backward compatibility.

As it is described in section 2.3, in an audio network any transmitted information must be available to all potential members of the network. Then, each individual STA will have to decide in the application layer if this information is needed or not. The best way to achieve this, in the case of a WLAN is data broadcasting. Unicast or multicast packets will be difficult to be implemented and will not benefit the performance of the network, as each transmission equally occupy the wireless medium, regardless of whether it is addressed to one, many or all STA in the network.

An 802.11 WLAN can be implemented in two main ways, in a centrally controlled configuration using an Access Point (AP), or in an ad-hoc configuration where stations communicate directly between each other. Taking in to account that latency is the most important issue in an audio network, the choice of the AP is not preferable. This is because for broadcasting using an AP, we have first to transmit the data to the AP which subsequently broadcast them to the entire WLAN. This causes for each packet in order to be transmitted to occupy twice the wireless medium and double its overall transmission time. The advantage of the use of an AP is that limits the hidden node problem because usually APs are centrally located in the WLAN and they are accessible by all STAs. In our case this is not an important issue because audio networks are implemented in a confined space which is usually a stage or a studio and therefore STAs can communicate directly without being out of range. Consequently, for ease of implementation and reduction of the delay in transmission, an ad-hoc type of network is preferable.

Another key characteristic in audio networking is the format of audio data. It is shown from both the research attempts and also the industrial implementations analysed earlier in this chapter that real time audio networking systems cannot be reliable using compressed audio. This is mostly because of the delay that is introduced by the compression and decompression algorithms. The suggested audio format is the uncompressed PCM audio with a minimum CD quality of 16 bit resolution and 24 KHz sampling rate.

Finally, an important issue is the number of STAs a wireless audio network must be able to support. IEEE 802.11 standard is able to support unlimited number of STAs in the same Basic Service Set (BSS). However, this is a feature that is not important for an audio network. Audio networks are limited by themselves as they usually contain a finite number of clients. This limitation is caused by various other parameters such as the number of performers, the audio channel of the mixing console, the number of the sound processing devices etc. Examining the existing wired audio networking systems but also the typical values when it comes to the audio lines in a professional audio system, we can safely assume that a wireless audio network able to support approximately sixty (60) STAs will sufficiently covers all potential needs.

Summarizing the above we can argue that a future wireless audio network will be functional and feasible if is based on the IEEE 802.11 wireless technology, it uses broadcasting as audio delivery method, have an ad-hoc networking configuration and if it is able to support at least sixty simultaneously active audio STAs delivering an 16 bit, 24 KHz PCM audio.

2.10 The IEEE 802.11 medium access control mechanisms

In order to investigate the ability of IEEE 802.11 technology to support a wireless audio network, according to the characteristics described in 2.8, and to propose possible modifications we need first to understand how this standard arbitrates the wireless medium. Below, a brief description of the standard is given, highlighting the characteristics that mainly affecting the deployment of a wireless audio network.

2.10.1 General Description

The IEEE 802.11 MAC is mainly designed for wireless unicast communication and for unlimited number of users in the network [39]. In the Distributed Coordination Function (DCF), which is its primary medium arbitration method, a Random Backoff Mechanism in conjunction with Virtual and Physical Carrier Sense mechanism provides a level of protection from collisions. The 802.11 2007 standard introduces an additional protection mechanism using Request-to-Send/Clear-to-Send (RTS/CTS) or Clear-to-Send-to-Self (CTS-to-Self) control frames. The last one is mainly used for Network Allocation Vector (NAV) distribution in mixed-mode environments where different 802.11 technologies coexist. Although RTS/CTS is used to address the hidden node problem, CTS-to-Self is used strictly as a protection mechanism for mixed-mode networks using data rates and modulation methods that legacy 802.11 technologies can understand. NAV is distributed by setting the duration field of the control frame with the time in microseconds required in order for the two parties to complete transmission including ACK. It is clear, however, that there is no MAC-Level recovery mechanism in broadcasting [40] and as mentioned before this could not be a choice in the case of live music audio networking due to significant buffering delays. In live music audio networking, the focus must be on preventing the loss of packets and the collisions instead of recovery and retransmission. NAV distribution is possible in broadcasting, only in mixed mode networks, by using the CTS-to-Self control frame [41]. CTS-to-Self is a standard CTS frame transmitted with a destination address of the transmitting station. The transmitting station cannot hear its own transmission in a half-duplex medium, but all nearby wireless stations (WSTAs) are alerted that a broadcast frame is pending and they can also update their NAVs with the value included in the duration field of the CTS-to-Self frame. As mentioned above, the use of CTS-to-Self is strictly limited in mixed-mode environments and it is using lower data rates that reduce throughput and increase delay. A modification of the 802.11 MAC to use CTS-to-Self as a main NAV distribution method, also using high data rates, will significantly contribute to the performance of the protocol especially in broadcasting. However, the use of CTS-to-Self alone cannot eliminate the collision's occurrence, which is caused by the drawbacks of 802.11 MAC random backoff mechanisms. This mechanism significantly contributes in collision avoidance but cannot totally eliminate them, especially when the number of WSTAs increases and there is also continuous data production, as in live music performance. In heavy data loads, there is a high likelihood that two or more WSTAs will choose the same backoff value. In this case the collision cannot be avoided.

2.10.2 Analysis of the IEEE 802.11 MAC algorithms

IEEE 802.11 MAC Layer is the lowest part of the Link Layer and it is placed between the Physical (PHY) and the Logical Link Control (LLC) sub-layer.MAC architecture is based on two basic coordination functions, Point Coordination Function (PCF) and Distributed Coordination Function (DCF). PCF is a contention free access method that provides polling intervals to allow uncontended transmission opportunities (TXOP) for participating WSTAs. This function will not be further analysed in this thesis for two reasons. First, because it demands the use of an AP which is against the characteristics of a wireless audio network as we defined them in (2.8) and second because it was never implemented in any commercial products. The optional Hybrid Coordination Function (HCF) that is introduced [amendment] to support QoS is also outside of our interest. In a wireless audio network all data are time sensitive and belong to the same category (i.e., audio), so there is no chance to divide them in different access categories and give them different priorities. In this study the fundamental DCF contention-based access mechanism is been used. DCF's timing diagram is illustrated in Figure 2.3 and its function is described as follows. A WSTA with a packet to transmit waits for the channel to become idle. When an idle period equal to DCF Inter-Frame Space (DIFS) is detected, it generates an initial backoff time value. This value indicates the period that the WSTA has to additionally defer before transmitting and it is randomly chosen from a "pool" of integers called Contention Window (CW). The random backoff process is the fundamental mechanism for the implementation of Carrier Sense Multiple Access with Collision Avoidance philosophy (CSMA/CA) used in IEEE 802.11 to prevent collisions. CW increases exponentially for every retransmission. The Short Inter-Frame Space (SIFS) is used for STAs that already gained access to the medium and they are in the process of exchanging control messages. Its use will be analysed later in this thesis.

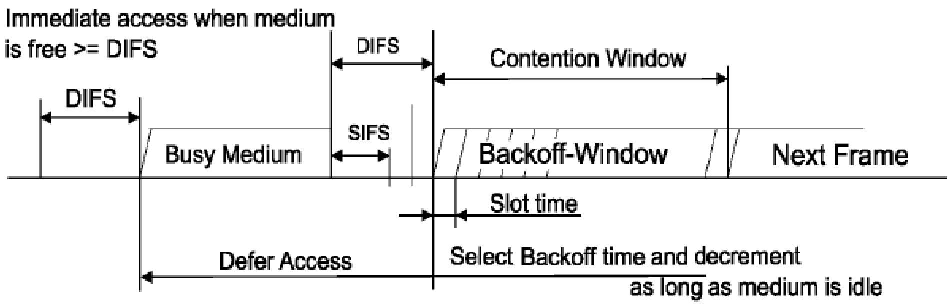

Fig 2.3: IEEE 802.11 basic access method

This process it is taking place independently in each STA. Under low utilization, stations are not forced to wait very long before transmitting their frame. If the utilization of the network is high, the protocol holds stations back for longer periods of time to avoid the probability of multiple stations transmitting at the same time. When we refer to Contention-Based access, random backoff is actually the primary mechanism for contention. The values for the random backoff time are extracted from the following formula:

$$Backoff\_Time = INT\ [CW \times Random(0, 1)] \times aSlotTime \qquad (2.1)$$

Random(0,1) is a pseudo-random number between 0 and 1 drawn from a uniform distribution. *CW* is an integer within the range of values *CWmin* and *CWmax* which are defined by the standard and they are different in different versions. The values of contention window are $CW=2^x-1$. The initial value of x starts from an integer defined by the IEEE 802.11 standard and increases for every unsuccessful retransmission attempt with maximum value equal to 10. For example, for x = 4, CW4 = 2^4-1 = 15, CW5 = 31, CW6 = 63..... CW10

= 1023. The *aSlotTime* duration is the value of the correspondingly named PHY characteristics. The backoff timer is decremented with one slot as long as the channel is idle. When a transmission is detected, the backoff timer freezes and starts to decrease again when the channel is sensed as being idle for a DIFS. When the timer reaches zero the data packet is finally transmitted.

2.10.3 Drawbacks of Random Backoff in Wireless Broadcasting

The IEEE 802.11 standard defines that the CW size exponentially increases for each retransmission attempt of the same packet (fig 2.4). However, as there is no retransmission in broadcasting, the CW size always holds the CWmin value. Under high utilization due to increasing number of WSTA and/or high data production, CWmin appears to be extremely small. In this case we are facing two major problems. The first one is that it is possible for a WSTA that just completed a transmission and has a new packet to send, to choose zero as its initial backoff time and start transmitting immediately after a DIFS. As we can see from equation (1), backoff time is a random outcome based on a uniform distribution but its range increases proportionally with the size of CW.

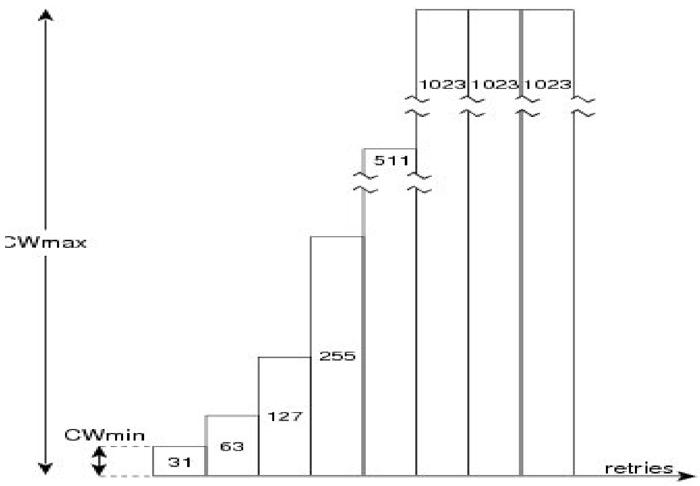

Fig 2.4: Exponentially increases of CW in IEEE 802.11 MAC

This consecutive transmission will give other WSTAs no chance to backoff. This problem is referred as the backoff counter consecutive freeze process (CFP), and was extensively analysed by Xianmin Ma and Xianbo Chen [42]. They show that the solution would be the ability a dynamic CW in broadcasting as it is in unicast transmission. The second and most significant problem in the case of wireless audio broadcasting is that there is a high likelihood for two or more WSTAs to choose concurrently equal backoff values. It is easy to understand that when we have fifty or more WSTAs producing continuous data and they are performing a backoff process using a CW = 15 (as in 802.11g and 802.11n) this is highly possible. In this case a collision is occurring and a data packet is lost as there is no recovery mechanism and no time for retransmission. This appears to be the main reason that 802.11 based wireless networks are not suitable for audio networking although they provide the required specifications.

2.11 Literature review in reliable broadcasting in wireless Ad-Hoc networks

The problem of multiple broadcasting of audio data within an ad-hoc wireless network has not been thoroughly investigated. Most of the research attempts are focusing in achieving reliability due to the lack of acknowledgment mechanism rather than quality of service. It is often assumed that the broadcasting packets are small in size and are used for control purposes or to discover and advertise resources. The methods proposed to increase reliability in wireless ad-hoc networks can be classified as “probabilistic”, where each node rebroadcast a packet with a given or calculated probability and “deterministic” where each node preselect some of its neighbours to rebroadcast the packet. More specific, the broadcasting methods can be divided in the seven following categories [43] [44] [45].

2.11.1 Simple Flooding Method

In Simple Flooding a source which wants to broadcast a packet, disseminates it to its neighbour STAs [46]. Each STA then checks if it received this message before and if it is not, it disseminate it again to its neighbours and so on. This process is repeated until all STAs finally receive the packet. This is a reliable broadcasting method but we can understand that it cause significant congestion problems because the medium is occupied several times for the same packet. It can also cause, under conditions, the broadcast storm problem. This method is characterized by significant delays and it not recommended for media type of data traffic.

2.11.2 Probability Based Method

This method is used in dense network and it is an alternative to the Simple Flooding. In dense networks many nodes share the same coverage area and thus it is possible for the broadcasting packet to reach all destinations without been retransmitted by all STAs. This technique save some network resources and improve delay performance. It uses the flooding philosophy but the STAs are retransmitting under a predefined probability. When the probability of retransmission is 100% this method is equal to Simple Flooding.

2.11.3 Counter-Based Method

This method is based on the research described in [47] which showed that there is an inverse relationship between the times a STA receive the same broadcasting packet and the probability of this STA to reach an additional uncovered area. Thus, when a packet is received for the first time, the STA sets a counter which increases every time the same packet arrives (from a different destination). If the counter reaches a threshold within a pre-set or randomly selected time period (depending on the implementation), it drops the packet, if not the packet is retransmitted to the nearby STAs. This method also reduces the congestion load of the wireless network compering to the initial Simple Flooding method.

2.11.4 Area-Based Method

This is also a method that uses rebroadcasting in order to achieve reliability in broadcasting within an ad-hoc wireless network. The decision for rebroadcasting is taken based on the location of the WSTAs. The idea is that if a broadcasting packet is received from a nearby STA that means that the surrounding area is covered and no retransmission is needed. In a distance-based scheme, each STA that receives a broadcasting packet set a random timer. When the timer expires it compares the distances of the STAs that sent the same packet. If there are STAs within predefined limits it assumes that the area is covered and thus does not rebroadcast the packet. In a location-based scheme, more precise techniques are used in order to define the location of the STAs in the network using GPS for example. Then this information is added to the packet's header. The receiving STA calculates the covered area according to the locations of the broadcasting and/or rebroadcasting STAs and decide whether to rebroadcast or not.

2.11.5 Neighbour Knowledge method

In this method, nodes periodically or dynamically transmit beacon messages to advertise their own existence and also obtain information about the surrounding nodes in their coverage area. Beacons usually contain the addresses of the nodes so a neighbour nodes list can be created. There are several broadcasting schemes that try to reduce redundancy in broadcasting within ad-hoc networks. In the "Self Pruning" scheme each STA maintain a neighbour STAs list. This list is also transmitted with the broadcasting packet. The receiving node compares his list with the transmitting node list and retransmits the packet only if there are additional nodes that could be covered [48]. In the "Dominant Pruning" scheme a two-hop neighbour list is used. Its broadcasting packet contains a list of nodes that are defined as gateways and are responsible for rebroadcasting the packet. The target here is for the packet to reaches all nodes within two hops [49].

2.11.6 Cluster Based methods

In this method nodes in the same network are forming clusters around a “head” node. The characteristic of the head node is that it has all nodes of the cluster within its coverage area. Thus, the head node is the only one which rebroadcast a broadcast packet. To rebroadcast message to nodes in other clusters, gateway nodes are used usually located in the boundaries of the cluster [50].

2.11.7 State of the art in broadcasting in wireless ad-hoc networks

Several variants of the above methods are proposed by researchers to further improve broadcasting in wireless ad-hoc networks. In [51] a reliable Minimum Spanning Tree flooding mechanism is proposed. This technique implements unicast transmission instead using also a link layer acknowledgment and retransmission. This method has comparable reliability with Blind Flooding (or Simple Flooding), limiting the broadcast storm problem. In [52] the authors propose an alternative Dominant Sets-based broadcasting method enhanced by a neighbour elimination scheme. This method has advantages over others when it comes to communication overhead. It uses instead of STAs IDs a “key” parameter which takes in to account the number of neighbour nodes (*degree*) and the surface coordinates of the node. The method increase reliability by reducing rebroadcasting packets and also the maintenance communication cost compared to clustered structure. Recent research attempts are investigating the use of network coding in order to improve transmission efficiency and reduce redundancy. In [53] the authors shows that coding-based probabilistic schemes in outperform non-coding probabilistic schemes regarding broadcasting efficiency. According to this approach, packets transmitted from various sources are grouped into globally unique sets called “generations” in order to be rebroadcasted. However, handling this type of coding in a distributed manner is a difficult task and it also increases decoding delay. In [54], network coding method is applied to deterministic broadcast schemes. Two coding algorithms are proposed and applied to a dominant pruning broadcasting scheme. The first one is a simple XOR-based coding algorithm and the second is a Reed-Solomon based coding algorithm. The results show that a gain up to 60% can be achieved when it comes to retransmission load, comparing to non-coding broadcasting methods.

2.12 Shortcoming of the existing broadcasting methods to handle real-time audio data

All the above methods ensure reliability by rebroadcasting the packets in the wireless network. They also reduce redundancy by applying several techniques that limits the number of rebroadcasting to the necessary. They are aiming to improve throughput and energy efficiency rather providing any kind of quality of service. These methods are suitable for small number of broadcasting packets which are also tolerant when it comes to delay. Real-time audio data broadcasting especially in a multiple broadcasting environment requires a very low delay delivery, giving no chance for any kind of retransmission. Thus, the above described methods are not suitable for wireless audio networks. Wireless audio networks are not suffering from coverage problems because usually are limited in small areas and also have limited number of nodes and low mobility. Their main problem is the collisions, caused in broadcasting, due to the heavy traffic created by real-time audio. This problem becomes more acute due to the drawbacks of the random backoff process as analysed in section 2.6.

To achieve reliable broadcasting over wireless audio Ad-Hoc networks a novel method is proposed in this thesis. This method consists of a number of modifications on the classic IEEE 802.11 standard, taking in to account all the characteristics and demands of audio networks as they are defined in section 2.9. The proposed modifications are implemented, tested and analysed in simulation environment. A brief description of the network simulation platform used in this research is given in the next section.

2.13 Network modelling and simulation

The implementation and evaluation of innovative ideas and practices in the field of network engineering is a complicated task. Modern networks are wide in range, are combining a tremendous variety of technologies and topologies and they are supporting numerous protocols and applications. The deployment of a new protocol or the modification of an existing one is difficult to be implemented in the real world, mainly due to the variety of systems that interact between each other and the range and size of the systems that has to be considered. Mathematical modelling is undoubtedly the most accurate way of research and analysis, but becomes extremely complicated when the system under consideration are becoming large in size and complexity. Network simulation is the dominant alternative method used today for network modelling, from both the industrial and the research community [55]. It is widely accepted as a research practice depending obviously on the accuracy of the simulation tool that is used.

2.14 An overview of network simulation tools

There are several network modelling and simulation tools available today. *QualNet, J-Sim, Prowler, NetSim* from *Cisco, PlanetSim, OPNET-Guru* and others are some of the well-known platforms. Their common characteristic is that they provide modelling capabilities up to a component-level. That means that networks can be designed and tested using only existing (specific or generic) models and commercial published protocols. The network simulation platforms that provide full developing capabilities and allow engineers and researchers to create their own models and protocols and also modify existing ones are, according to author's opinion, OPNET Modeler, OMNeT++ and the NS2 - NS3 family. The NS2-3 is a Linux-based simulator free for academic and research purposes. It is particularly popular among researchers especially those with computer science background. It is a Discreet Event Simulation (DES) platform based on C++. Its success among the scientific community results a significant number of models and code to be available. The disadvantage of NS2-3 is that it has no graphic network design environment. That makes de design of the network a difficult task. Generally, NS is powerful tool which need very good programing skills. It is dominating the research sector mostly because its reliability and the abundance of freely available models and protocols. OMNet++ is also a freeware network simulation software for research purposes. It is a DES platform that uses C and C++ and allows users to access its source code in order to modify existing models and protocols and create new ones. The network design is done by writing code using a specific language called NED, but there is a graphical visualisation environment and a user friendly interface for attributes and statistics setting. It is has also the advantage of running in windows. OMNet has a dynamically increased community of researchers and there are many models available nowadays. OPNET Modeler is an industrial standard network simulator that is available as a commercial product. It is also a DES platform based on C and C++ and allows users to access the source code of all its models and also create new ones. It has an extensive documentation with many tutorials and paradigms covering all areas and also a designing support department. OPNET provides a fully graphical design environment and a vast selection of generic models but also models of commercial network devices from major vendors. It is the dominant network simulation platform in the industry and in government research institutions globally. It is also highly accepted from the academic research community. For this research

OPNET Modeler has been used mostly because its reliability, the wide availability of its models and its convenience in terms of network design. In addition, OPNET provides a "wireless suite" with extra futures for wireless network design and implementation.

2.15 Discrete-event Simulation (DES)

OPNET is a "discrete-event simulation" platform. The concept of modelling a system based on a DES is that the system jumps from a state to its next state triggered by the occurrence of an event. Events are specific activities that occur at a certain time. Each event is scheduled at particular instant in time and the system remains unchanged in the time between events. Figure 2.5 shows the evolution of DES over the simulation time. Each event consists of at least two characteristics, *time* and *type*.

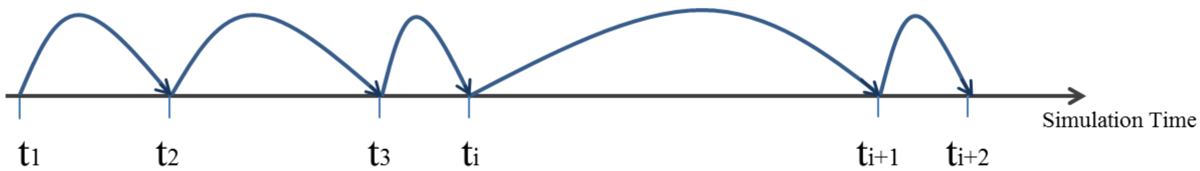

Fig 2.5: The Discrete-event simulation evolution over the time

The *time* defines when this event is scheduled within the simulation time and the *type* defines the kind of this event. All events are recorded in an *event list* (EL) which is dynamic. That means, new events can be created and enter the list as the simulation is executed. The entire simulation shears the same EL, a *simulation time clock* as well as a set of *System State* variable, *Statistical* variables *Event routines* and the *Timing routine*. In order to understand the DES operation the event-scheduling algorithm is described below. For an EL which contains all events ordered according to their occurrence within the simulation time we have:

$$EL = \{ E_1(t_1\text{-}type_1), E_2(t_2\text{-}type_2), E_3(t_3\text{-}type_3), \ldots\ldots, E_i(t_i\text{-}type_i) \} \quad with \quad t_1 \leq t_2 \leq t_3 \leq \ldots\ldots \leq t_i$$

After the simulation starts the next event is executed from the EL. In this case the first event (E_1) scheduled for time (t_1), due to its type ($type_1$) causes the call of the *Event routine 1* which brings the system in a *State 1*. The event routine may change the state variables, update statistics and generate new event notices in the EL. The system remains in this state and if the EL is not empty the *Timing routine* deletes the E_1 from the EL and retrieves the next in order event (E_2) which brings the system in *State 2* and so on [60]. When the EL is finally empty the simulation is terminated. In network DESs the events in most cases are data packets that are created, move through layers, are transmitted, received and destroyed in the network. Figure 2.6 shows a flow diagram that describes the above operation.

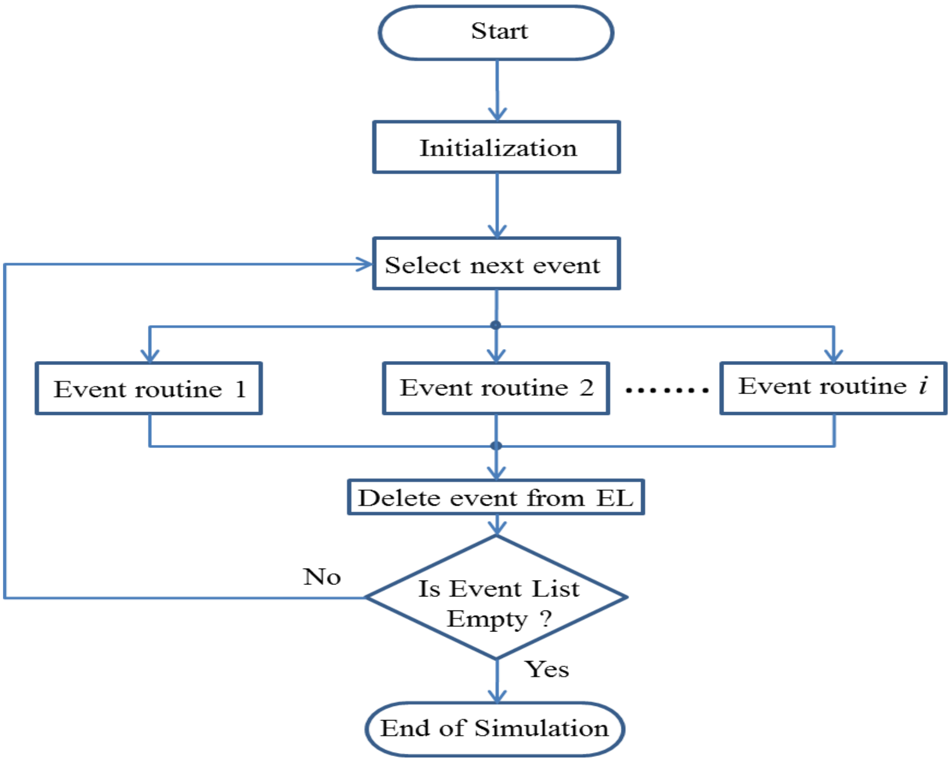

Fig 2.6: Basic Discrete-event simulation algorithm

2.16 OPNET's general characteristics

OPNET is a high level event based network simulation platform. The simulation operates at "packet level" based on the above described DES framework. It provides an extensive collection of generic and commercially available network hardware and protocol models. The simulation is controlled by a centralised "kernel process" which pass control to processes that are executed in the models when required. Kernel process is not accessible from the users but all model's processes are open source and available for modification. Developers can modify existing models and protocols and create new ones using a C-type programing language called Proto-C. This is actually a regular version of C which contains a vast library of OPNET specific functions especially designed for data network applications. OPNET provides an extensive ability to adjust the network operation including traffic generation and application's behaviour. It also has a number of editing abilities like *Probability Density Function* (PDF) and *Packet Format* editors. Network models in OPNET follow a hierarchical structure. They are divided to three main domains, the *Network* domain the *Node* domain and the *Process* domain. Figure 6 provide a graphical representation of OPNET's hierarchical architecture.

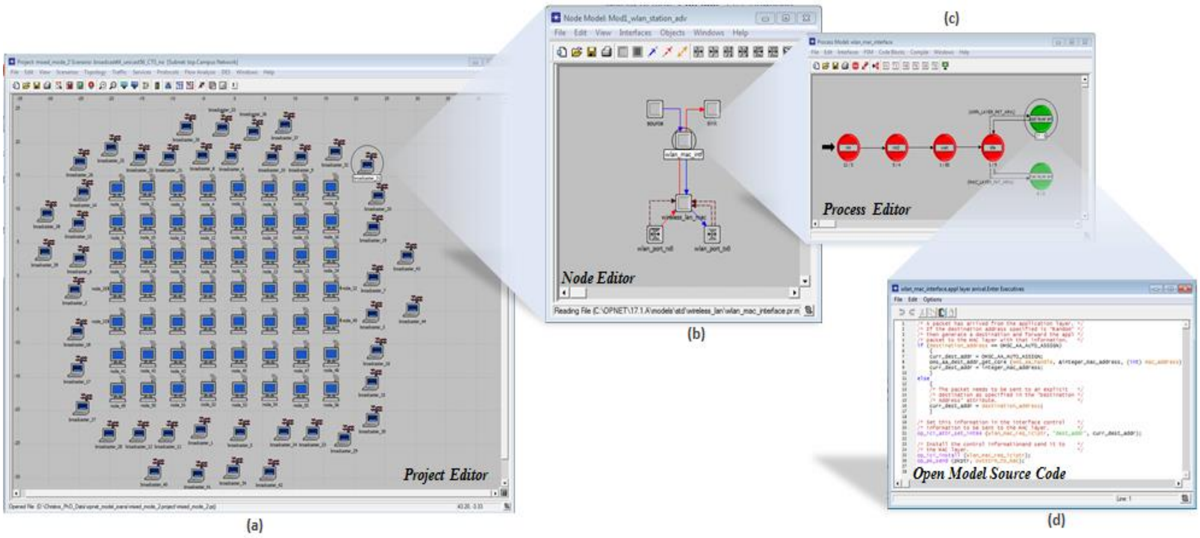

Fig 2.7: OPNET Modeler hierarchical architecture

2.16.1 Network Domain

The network domain is the the section of OPNET where the topology of the network under study is designed (Fig. 2.7-a). It can also be classified in unlimited number of sub-networks. The network is built by graphically adding nodes in the project editor and connecting them between each other with specified data links. In the case of wireless networks (fig. 2.7) the wireless medium (radio channel) is assumed. In addition geographical coordinates and mobile node trajectories are defined in this domain.

2.16.2 Node Domain

The node domain is the section of OPNET where the characteristics and the operation of each individual node are defined (Fig. 2.7-b). Typical nodes include workstations, packet switches, hubs and routers, satellite terminals, remote sensors and other components of a network infrastructure. The operation of each node is also defined using various modules that are combined using a graphic way. Modules are the basic building blocks of node models. Modules include *processors*, *queue*s, *transmitters* and *receivers*. Processors are the primary general purpose building blocks of node models and are fully programmable. Queues offer all the functionality of processors but can also buffer and manage a collection of data packets. Transmitters are the outbound interfaces between objects inside a node and communication links outside it, while receivers are the inbound interfaces. There is a variety of transmitters and receivers available for point-to-point and wireless data communication [61]. Figure 2.8 shows the collection of available modules.

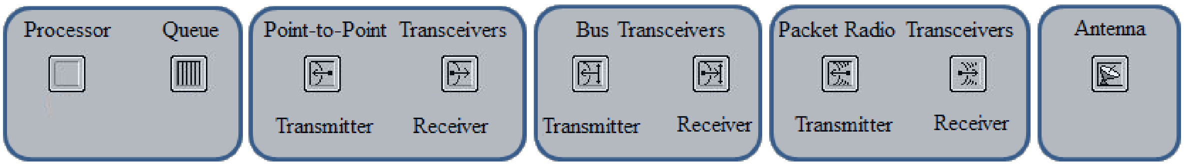

Fig 2.8: Node Domain Modules

Modules are connected between each other using three types of connections; *packet streams, statistic wires* and *logical Tx/Rx associations.* Packet streams carry data packets from a source to a destination module and it is the "physical interconnection" between modules. Statistic wires carry a single data value from a source to a destination module in order to report the occurrence of an event. A logical Tx/Rx association is used to establish an association between a transmitter and a receiver to indicate that they perform a function as a pair.

2.16.3 Process Domain

Process domain is the part of the software used to specify the behaviour of processor and queue modules which are existed in the node domain (Fig 6-c). In OPNET each module is modelled as a *finite state machine* (FSM). A FSM implements the behaviour of a module by determining what action the module can take in response to an event [56].

2.16.4 Finite State Machines

A FSM is a mathematical model of computation to design a sequential logic. FSM are widely used in computer programing but also in engineering in biology and other sciences thanks to their ability to model complicate systems and describe sequential behaviour. State machines are virtual devices that model the behaviour of a system by analysing the states that the system takes and its reaction to all possible events. The machine always remains in a known state. When an event occurs the machine makes a transition to a next state depending on the type of the event. A finite state machine is expressed visually with a state transition diagram [57]. In this diagram a state is represented with a circle and a transition with an arrow. In figure 2.9 we can see the transition diagram of a basic FSM that accepts a binary input.

According to this example, when the machine is activated executes the initialization process and enters the state-1. An input of logical "1" will keep the machine in state-1 while an input of a logical "0" will cause a transition to state-2. Once in state-2 an input of "1 will keep the machine in state-2 and an input of "0" will switch it back to state-1.

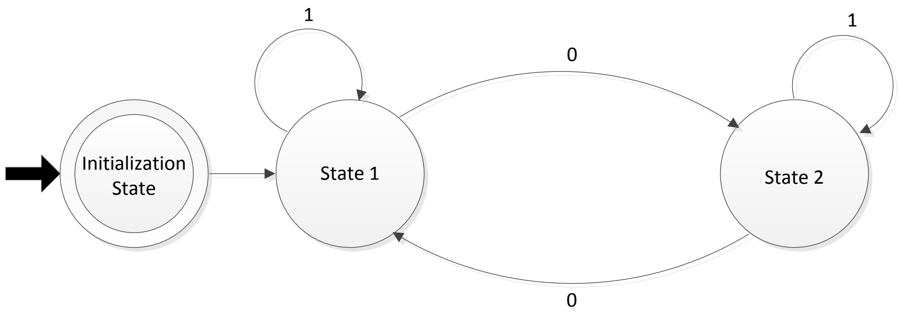

Fig 2.9: Basic FSM accepting binary input

A further analysis in the techniques and applications of FSM is beyond the scope of this thesis. Our interest is focused in the way that OPNET is using FSM to create processes and model the behaviour processor and queue modules.

2.16.5 OPNET process modelling

Inside of each processor and queue module in OPNET a process model is executed. A process model is a FSM that represent the logic and the behaviour of the module using states and transitions. A state is the condition of a module at a given chronological instant within the simulation time. A transition is the change of a state in response to an event. The condition of a module within the states and the transitions are defined in OPNET by programing code. Fragments of C/C++ code can be attached to each part of an FSM. This code, augmented by OPNET-specific functions, is called Proto-C [56]. Each state in OPNET is divided in two main parts. The *enter executive* and the *exit executive.* Code can be attached

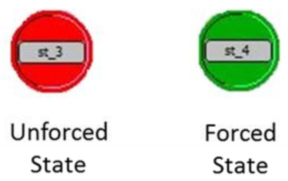

Fig 2.10: Type of states in OPNET Modeler

independently in each part. OPNET uses two types of states; *forced* (green) and *unforced* (red), (fig 2.10). Parts of the code that defines the module's operation can be attached to the transitions.

When the process enters a forced state it executes the part of code in enter executive, then executes the part of code in exit executive (fig 2.11), and transition to the next state. When the process enters an unforced state, after executing the code in enter executive the process model blocks. It stops execution and returns control to the Simulation Kernel. The next time the process model is invoked, execution continue with the exit executive of the unforced process

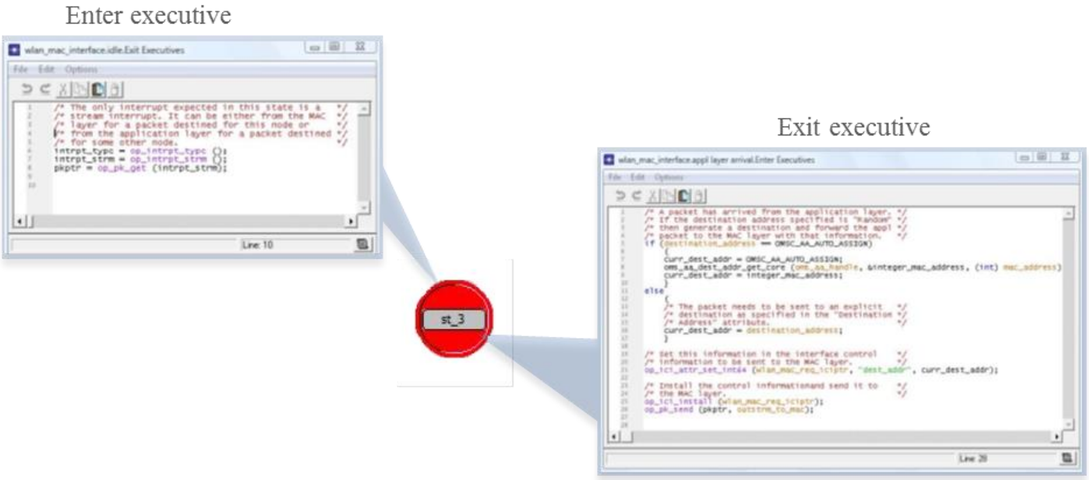

Fig 2.11: Enter and Exit Executives of a OPNET State

2.16.6 OPNET project and project editor

The network domain is designed in the main editor of OPNET which is called *project editor*. Each project can be fragmented in different scenarios. Each scenario represents a particular instant or version of the network with a separate simulation running for it. Each model (node) in the network has a detailed list of attributes that define its operation. The attributes setting process is also taking place in the project editor. Local statistics for each node and global statistics for the entire network are also selected here. Finally in the project editor, we define the simulation characteristics, we implement the debagging procedures and we obtain and analyse the simulation results.

When it comes to modelling of wireless networks, OPNET provides a complete wireless suite that implements all the well-known wireless protocols. It also proved tools for simulating mobility for round, airborne, and satellite systems and modelling antennas.

OPNET is an advanced network modelling and simulation platform. It is an industrial standard tool that requires a solid knowledge of algorithm design, good programming skills and a deep understanding of packet networks technology in order to provide reliable and accurate results.

2.17 Summary

In the first section of this chapter the background for all aspects of this research is provided. Initially, a brief description of conventional audio systems and an introduction to the fundamental of sound engineering is given.

In the next section, a compact but analytic description of the audio networking systems existing today and the most important relevant standards is provided. The advantages of audio networking systems over the conventional analogue audio systems are also thoroughly discussed. The key characteristics of the existing audio networking systems and standards, the technical specification as well as their problems and limitations are analysed in order to understand their importance and to define research challenges.

In the third section of this chapter an introduction to wireless audio systems is performed. Here we are trying to document and support the argument that the development of wireless audio networks will significantly benefit the evolution of audio networking concept. The characteristics, the demands and the limitations of the wireless audio networks are analysed and the research challenges of this sector are clearly defined. A literature review in the resent research in the field is also performed. Afterwards, all the existing wireless technologies are examined in order to define the most appropriate technology to support a wireless audio network for highly demanding professional applications. We conclude that the IEEE 802.11 (Wi-Fi) is the most suitable to operate as networking platform for future wireless audio networks, however the limitation of the standard to handle broadcasting of heavy data traffic in ad-hoc networks must be resolved. At this point, an additional literature review regarding recent research in broadcasting over wireless ad-hoc networks is also performed and most specific research targets are defined.

Finally in the last section, a brief introduction on the fundamentals of network simulation is performed and the operation of OPNET modeler, which is the major simulation tool used in this research, is described.

Chapter 3

Expanding the use of CTS-to-Self mechanism to improve broadcasting on IEEE 802.11 networks

3.1 Introduction

The key point in the design of wireless audio networks is to improve throughput performance in broadcasting, maintaining at the same time the over transmission delay at low levels. Considering that audio networks require a real time data delivery, retransmission and other recovery techniques for lost data packets cannot be applied. The major problem that leads to lost packets in broadcasting over 802.11 wireless ad-hoc networks is collisions. Collisions caused by the inherited problem of the standard to support reliable broadcasting especially in heavy data traffic environment. The main aim of this chapter is to implement and test a protection mechanism for broadcasting, similar to the one provided by the standard for unicast transmission. For this reason the extended use of CTS-to-self message is proposed. CTS-to-self is a control message initially designed to offer protection in networks where latest and legacy 802.11 technologies coexist. In this chapter the MAC algorithm of the 802.11 standard is modified in order to act as a protection mechanism for broadcasting packets. The chapter is organized as follow:

Initially a description of the RTS/CTS protection mechanism, provided by the standard for unicast transmission, is analysed. Then the current use of the CTS-to-self mechanism is described. The probability of collision during broadcasting in WLANs is defined. Subsequently a comparative study between theoretical and simulated model is performed using OPNET. This study is also used to validate the accuracy of OPNET WLAN model. Finally the proposed modified use of CTS-to-self is analytically described. The modifications are implemented and simulated in OPNET and the results are analysed and discussed.

3.2 RTS/CTS protection mechanism

As it was mentioned in 2.10, the main mechanism for collision avoidance in the IEEE 802.11 Distributed Coordination Function (DCF) is based on the combination of the exponentially increased CW in its MAC algorithm with the standard ACK mechanism, provided only for unicast transmission. An additional optional technique for collision avoidance is the use of RTS/CTS (Request-to-Send/Clear-to-Send) control messages. This technique performs a Network Allocation Vector (NAV) distribution and helps to prevent collisions. According to this, a RTS control message is sent prior to every unicast transmission. This message contains the sender and recipient addresses and also a duration field which contains the time in μsec that all other STAs have to defer transmission for this period. The recipient replies with a CTS message after a SIFS time. This message contains only the initial STA address and the duration field, appropriately modified. This modified value represents the remaining time in order for the transmission to be completed. Equations 3.1 and 3.2 shows the way NAV is calculated in both cases. Figure 3.1 shows a timing diagram for the sending and receiving STA as well as for STAs that perform virtual carrier sensing.

$$NAV_{RTS} = 3 \times SIFS + T_{CTS} + T_{DATA} + T_{ACK} \quad (3.1)$$

$$NAV_{CTS} = 2 \times SIFS + T_{DATA} + T_{ACK} \quad (3.2)$$

Where T_{CTS}, T_{ACK}, T_{DATA}, the time required for the transmission of CTS, ACK and data packet respectively [41] [58].

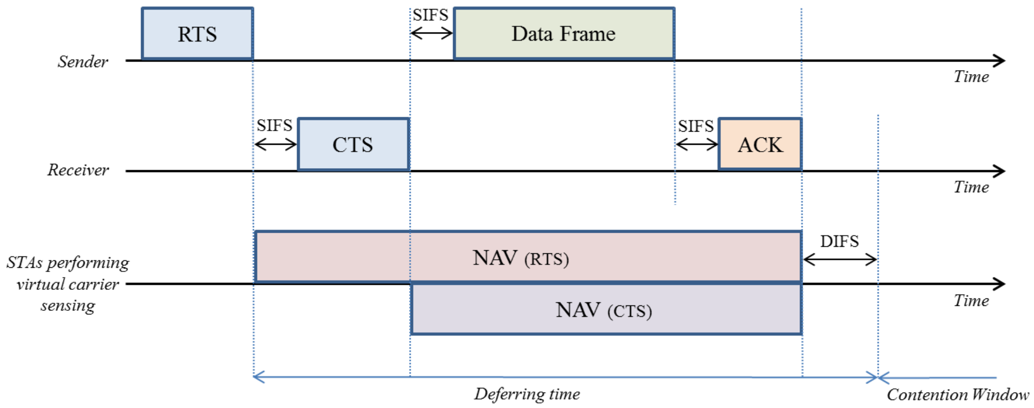

Fig 3.1: IEEE 802.11 basic access method

The CTS transmission plays an additional very important role. It provides an extra protection regarding the so-called "hidden node" problem. As long as WLANs are expanding in wide areas there is a strong possibility two or more nodes in the same BSS not to be able to "hear" each other transmission, but there able both of them to communicate with a third one. In such a case, a simultaneous transmission will cause a collision. In figure 3.2, STAs “A” and “B” are out of range but they are both in range with STA “C”. If “A” wants to transmit to “C” and implement the RTS/CTS technique, STA “C” will replay with a CTS message that will be received also from “B”. NAV information related to "A" transmission will be transferred indirectly to STA "B", who will defer transmission until the transmission of "A” is completed. This technique is even more effective when it is used with an AP which usually is located in a central point of the network.

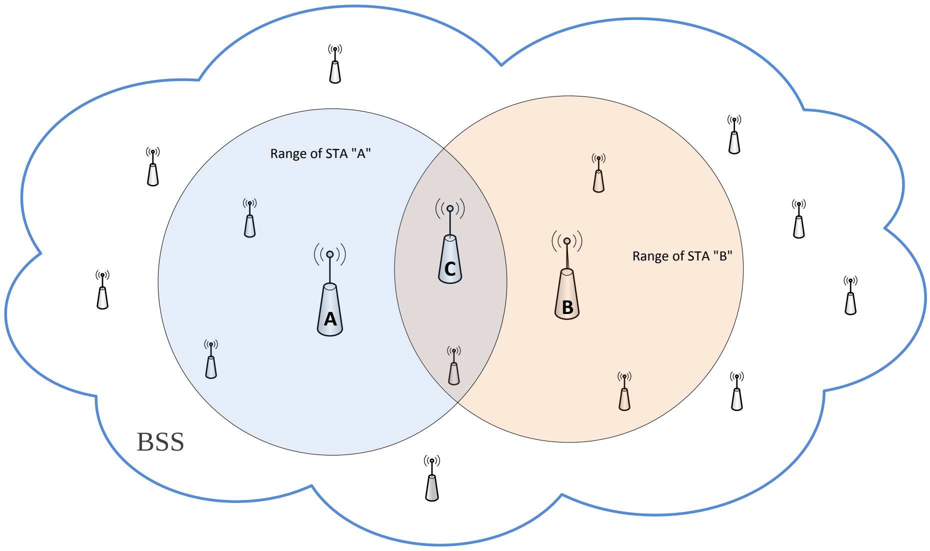

Fig 3.2: The "hidden node" problem

The conclusion of this process is very important. It means that according to the IEEE 802.11 standard, the CTS messages are treated independently and the NAV can be distributed within the WLAN without necessarily the full RTS/CTS messaging. However it is clear at this point that the above described protection mechanism cannot be applied in multicast and broadcast transmissions as long as CTS is always transmitted as a follow up of a RTS message, which in order to be transmitted must have a recipient address in his body.

3.3 The regular use of CTS-to-Self control message in IEEE 802.11

The evolution of the IEEE 802.11 standard included, among others, changes in the physical technology and modulation types. However, there was from the beginning an intense effort to maintain backward compatibility. So far 802.11 have defined seven PHYs (Clauses) as of 802.11n and there is one more recently issued with 802.11ac amendment. Excluding the IR (Infrared) the FHSS (Frequency Hopping Spread Spectrum) PHYs and the 802.11ac, those are:

- Clause 15 [Direct Sequence Spread Spectrum (DSSS) PHY for 2.4 GHz, defined in the original 802.11 specification]
- Clause 17 [Orthogonal Frequency Division Multiplexing (OFDM) PHY for 5 GHz, defined in the 802.11a amendment]
- Clause 18 [High Rate (HR)/DSSS PHY for 2.4 GHz, defined in the 802.11b amendment]
- Clause 19 [Extended Rate Physical (ERP) PHY for 2.4 GHz, defined in the 802.11g amendment]

- Clause 20 [High Throughput (HT) PHY for 2.4 and 5 GHz, defined in the 802.11n amendment]

Each of these PHYs defines the operating band, the modulation type and operating rules that a STA may use. However, devices with incompatible PHYs cannot sense each other as occupying the medium which would clearly lead to a break-down in the CSMA/CA system.

The 802.11-2007 and its amendment in 2012 mandate support for both, Direct Sequence Spread Spectrum (DSSS) and Orthogonal Frequency Division Multiplexing (OFDM) technologies for Clause 19 ERP and Clause 20 HT [59] [60]. The following list gives a detailed description on how the backward compatibility should be implemented.

- STAs implementing the Clause 18 PHY, must also implement the Clause 15 PHY.
- STAs implementing the Clause 19 PHY, must also implement the Clause 18 and 15 PHY.
- STAs implementing the Clause 20 PHY (in 2.4 GHz), must also implement the Clause 19, 18 and 15 PHY.
- STAs implementing the Clause 20 PHY (in 5 Ghz), must also implement the Clause 17 PHY.

To achieve this, the standard uses a technique similar to RTS/CTS. When the presence of a non-ERP or non-HT technology identified in the network (mixed mode environment), ERP and/or HT STAs use for their transmission a protection mechanism based on a specified message called *Clear-to-Send-to-Self* (CTS-to-Self). CTS-to-Self has the structure of a regular CTS message. The only difference is that it sent from a STA to itself. That means that receiver's address field in the message is set to the address of the transmitting STA.

The protection mode can be initiated by an AP or any other STA in the BSS by setting the ERP information element in its beacon frame. Every time such a beacon is transmitted, every ERP or HT STA that hears the beacon will understand that protected mode is set. From now on, when a STA want to send data, it performs NAV distribution by sending a CTS-to-Self frame using one of the mandatory Clause 15 or Clause 18 rates and using one of the mandatory Clause 15 or Clause 18 waveforms that all STAs can understand. This notifies all STAs in the BSS that they must wait for a given period of time until the data and ACK have been transmitted. Then the data and ACK will be sent at faster speed according to the technology and rate used in the network. The CTS-to-self NAV distribution mechanism is lower in network overhead cost than is the RTS/CTS NAV distribution mechanism, but CTS-to-self is less robust against hidden nodes and collisions than RTS/CTS. Figure 3.3 shows the format of a CTS-to-self message.

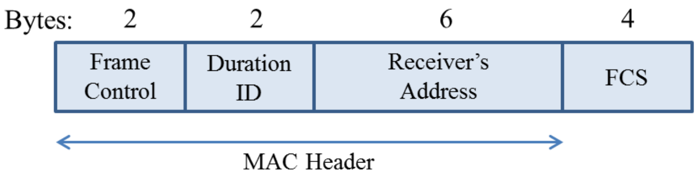

Fig 3.3: CTS-to-Self frame format

The cost the use of a CTS-to-self is more overhead on every ERP and HT transmission. This reduces the benefits of all the 802.11g/n improvements, resulting in significantly lower effective when operating in mixed environments [61]. The techniques used by ERP and HT STAs to obtain and distribute the information of the presence of a legacy technology in the WLAN it is not discussed here as it is beyond the scope of this chapter [62].

Although is not clearly defined in the standard, it is a common practice for wireless systems developers to activate the CTS-to-self protection mechanism always when a broadcast transmission needs to be protected [*OPNET Doc*]. Nevertheless, as we described in 2.10.3, collisions cannot be eliminated. Particularly in a media broadcasting environment, collisions are becoming a major problem especially when the number of broadcasting STAs increases.

3.4 Probability of collision in a multi-broadcasting wireless network

In order to propose a protection mechanism against the collision problem we must first investigate and define the size of the problem in theoretical and practical level. In this section we first use elementary probability theory to calculate the probability of collision in a wireless ad-hoc network. Then we simulate such a network and compare the theoretical and the simulation results. This process has also an additional importance. It is a good practise in order to evaluate our simulation model and the results obtained from it.

3.4.1 Calculating the probability of collision in broadcasting over IEEE 802.11 networks

As it was discussed in 2.10.3, in case of broadcasting in an 802.11 network the size of CW remain constant and it always holds its minimum value (CWmin). In the case of an audio network the data production is continuous. Therefore, we can safely assume that a saturated network where all STAs have data to transmit, constantly attempt to access the medium. When the medium is sensed idle for a DIFS, a random integer backoff time is selected from a range of [0 – *CWmin*-1] with a uniform distribution. The backoff counter is decremented as long as the medium is sensed idle. When the medium is sensed busy, the countdown freezes and resumes decrement when the medium is sensed idle again. A collision occurs when the backoff counter of two or more STAs reach zero simultaneously [63]. Based on the above, the probability p_1 of a STA to transmit in an arbitrary slot is:

$$p_1 = \frac{1}{CWmin} \tag{3.3}$$

According to the principals of the elementary probability theory, the probability of a STA not to transmit in an arbitrary slot p_2 is:

$$p_2 = 1 - \frac{1}{CWmin} \tag{3.4}$$

In broadcasting wireless ad-hoc networks, each attempt from STAs to access the medium is an independent event, as each STA implements the random backoff process autonomously. If the total number of STAs in the network is n, and a STA i, transmitting in an arbitrary slot, the number of remaining STAs in the network is n-1. The probability $p_{2\ (n-1)}$, of no other STAs transmitting in this slot, from (3.4) and according to the *product law for independent events* [64], is:

$$p_{2(n-1)} = p_{2(1)} \times p_{2(2)} \times \cdots \times p_{2(i-1)} \times p_{2(i+1)} \times \cdots \times p_{2(n)}$$

$$p_{\ 2(n-1)} = \left(1 - \frac{1}{CWmin}\right)^{n-1} \tag{3.5}$$

A collision occurs when at least one more STA transmit in this slot. This probability $p_{(collision)}$, considering the (3.5), is:

$$p_{(collision)} = 1 - p_{2(n-1)} = 1 - \left(1 - \frac{1}{CWmin}\right)^{n-1} \tag{3.6}$$

From (3.6) we can see that the parameters affecting the broadcasting probability of collision in a saturated wireless ad-hoc network are the size of CW and the number of STAs in the network [65]. In case of broadcasting CWmin is constant and therefore (3.6) can be solved easily. The graph in figure 3.4 shows the probability of collision as it is calculated using (3.6) for an increasing number of saturated STAs. For this calculation, the value CWmin=15 has been used. This is the value of CWmin, specified in IEEE 802.11g amendment.

We can see from figure 3.4, that in a multi-broadcasting environment where all STAs have data to broadcast (saturated network); the probability of collision is dramatically high. It is approximately 50% for 12 broadcasting STAs and it became almost 100% when the number of STAs exceeds 50. This is actually the key parameter that precludes the use of the IEEE 802.11 technology from being used as a wireless audio networking platform, although there is plenty of bandwidth available.

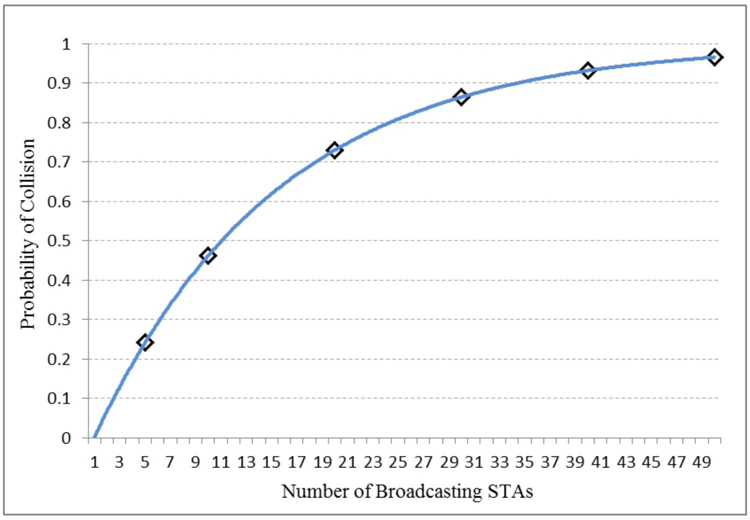

Fig 3.4: Theoretically calculated probability of collision in broadcasting

3.4.2 Measuring the probability of collision

In this section a comparative study between theoretical and measured probability of collision is performed. The purpose of this study is to investigate the effect of the CW size in the number of collisions occurring during broadcasting and also to validate the accuracy of the OPNET's wireless model that will be used in the rest of this research.

OPNET implements its IEEE 802.11 models following precisely the published specifications of the standard. However, predefined parameters cannot be accessed through model's attributes. One such example is the size of CW in the MAC algorithm. When traffic set in broadcasting, the CW in OPNET wireless model remain constant and always holds its minimum size which for 802.11g is 15. The only way to simulate a broadcasting transmission in OPNET with various CW sizes is to modify the code executed inside the model.

For the simulations in this research the *wlan_station_adv* node model of OPNET is been used. The model, in the node level, consist of a *source* processor where the outbound packets are created, a *sink* processor where the inbound packets are destroyed a radio receiver and a radio transmitter, a *wlan_mac_intf* processor and the main *wireless_lan_mac* processor where the MAC algorithm is implemented. Inside the *wireless_lan_mac* processor, in the process level, there is only one state called *spawn* which is executed ones at the beginning of the simulation. This is actually a parent process which reads the attributes that define the physical characteristics of the node, and the surrounding nodes in the network and creates the appropriate MAC child process and it never weak again during the simulation. More details about the insides of the *wlan_station_adv* of OPNET will be given later in this chapter. For this study the standard *wlan_mac* process is used. This process consists of the state diagram and the initialization code mainly included in the *function lock* (FB). Here, a series of functions that define the operation of the process are created and a series of parameters are set. Figure 3.5 shows the part of the code where the CWmin is declared.

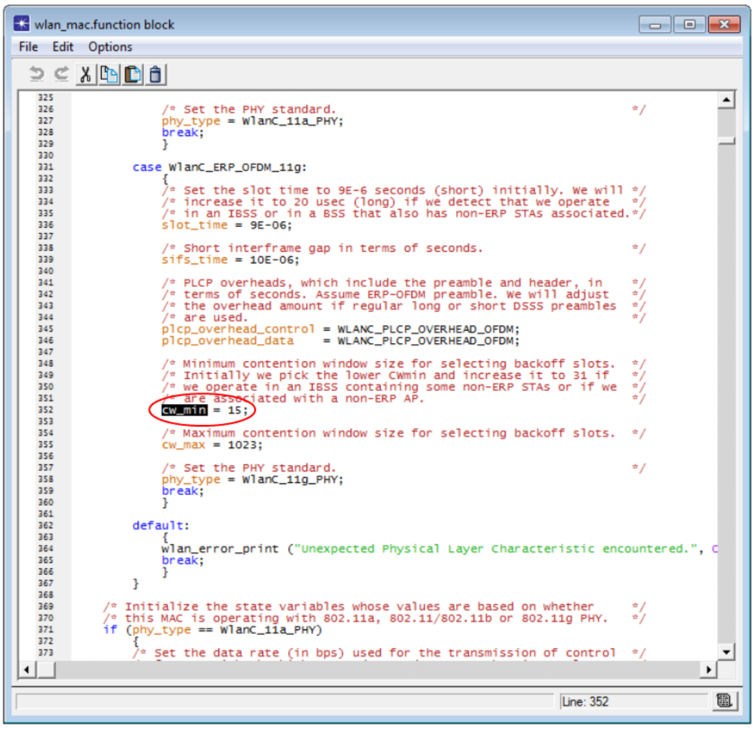

Fig 3.5: CWmin parameter in function block-wlan_mac process

For this study the size of CW window was manually changed with values 15, 31 and 63. OPNET provides a statistic which reports the collision status but there is no collective statistic to measure the number of collisions in each STA. For more accurate measurements a *collision counter* statistic was created. This statistic monitors the changes in the "collision flag" and precisely reports the exact number of collisions encountered in each STA during the simulation (fig 3.6). Adding custom statistic is an important tool in OPNET. Details about the creation of the custom statistics are given in Appendix A.

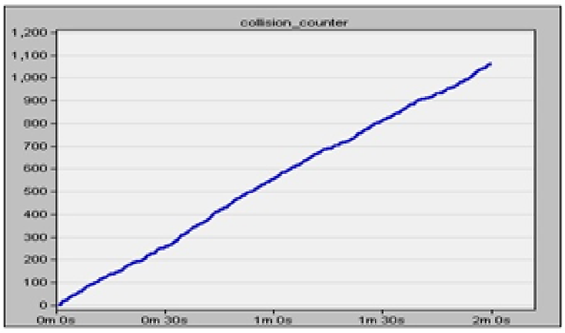

Fig 3.6: Collision Counter Statistic

The simulation parameters are: packet size 1024 bytes, physical characteristics 802.11g-24Mbps, bit rate 400 Mbps. The simulation time was 2 minutes and it was run three times using different seed number.

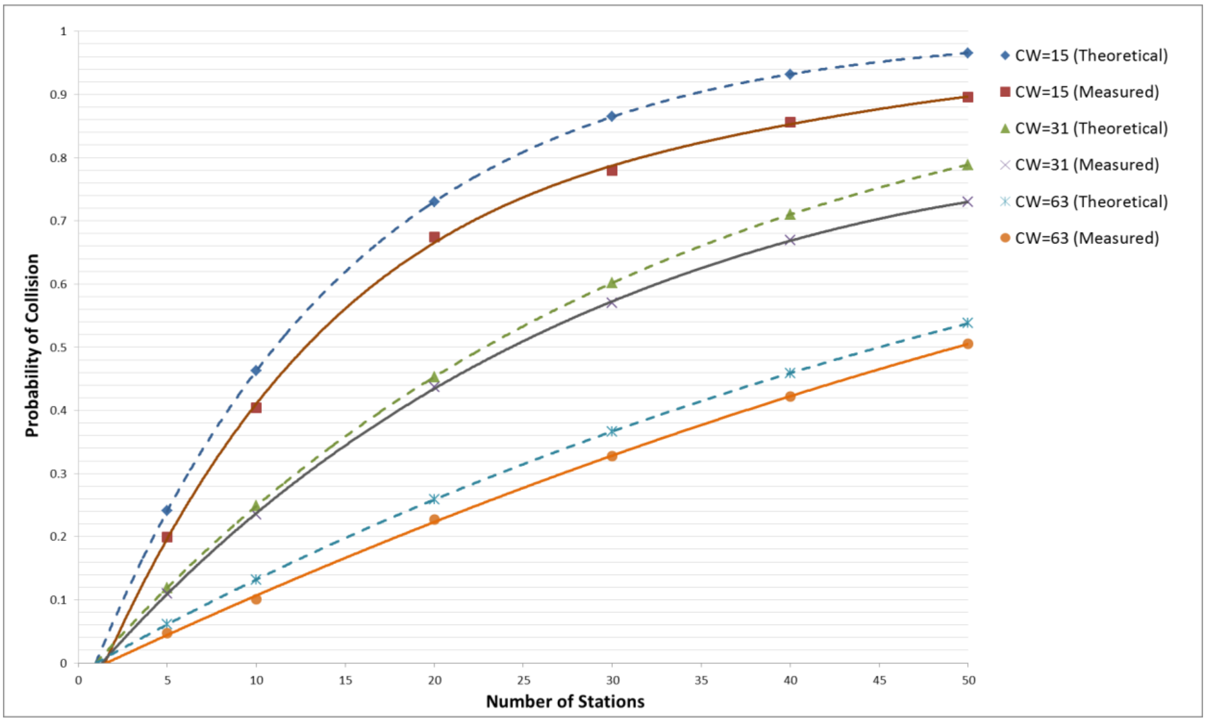

Fig 3.7: Probability of collision (Theoretical and Measured)

The graph in figure 3.7 shows the theoretically calculated probability of collision *(dashed lines)*, for three different CW sizes (15, 31, and 63) and also the number of collision resulting from the simulation of a saturated network where all STAs are broadcasting their data.

The conclusion drawn is that the CW size affects dramatically the number of collisions experiencing in broadcasting environment. Increasing the CW from 15 to 63 we can reduce almost 50% the probability of collision. Another important outcome from this study is the validation of *wlan_station_adv* node model of OPNET. The model appears to operate sufficiently as long as the theoretical and the measured results are matching.

3.5 Proposed modified use of CTS-to-self message

In this chapter, the use of CTS-to-self as a protection mechanism in data broadcasting in IEEE 802.11 networks is proposed. This modification does not demand any additional change in the receiving process. CTS-to-self is treated at the receiving STA as a regular CTS message. All changes have to be done in the transmitting process. More specific, for broadcasting packets the STAs enters the "broadcasting protection mode" where a CTS-to-self message containing NAV information, is transmitted in a similar to the RTS/CTS way. As it is described in 3.3, CTS-to-self frame is transmitted using one of the mandatory Clause 15 or Clause 18 rates and modulation. This low speed transmission reduces the overall performance of the network and adds delay. Therefore, the physical characteristics of the CTS-to-self have to be modified. More specific, the MAC algorithm of IEEE 802.11 standard is reprogramed to transmit a CTS-to-self message prior to each broadcasting packet [66]. The second modification in the MAC is to reprogram the physical characteristics of the CTS-to-self transmission. CTS-to-Self transmission parameter has been modified to always adjust with the bit rate and modulation technique used for data transmission in the wireless network.

According to the proposed amendment, when a STAs has a packet to broadcast waits for the channel to become idle (Fig 3.8). When an idle period equal to DIFS is detected, it additionally defers, performing the random backoff process. When random backoff count down reach zero and the medium is sensed idle for a DIFS period, a CTS-to-Self is transmitted. After a Short Inter-frame Space (SIFS) period the data packet finally is broadcasted. The CTS-to-Self frame contains in its "duration" field the time that all non-transmitting STAs must defer before trying to access the medium.

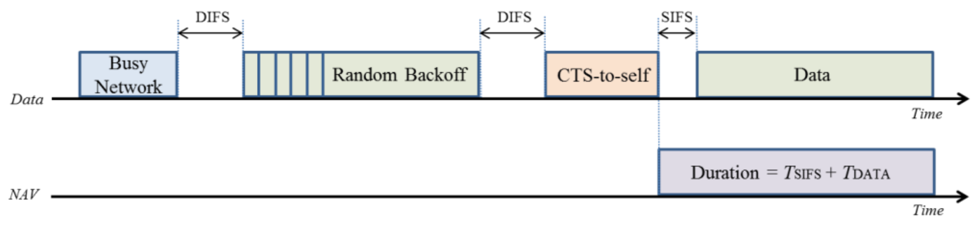

Fig 3.8: Proposed use of CTS-to-Self

3.5.1 Advantages of the proposed modification

The first advantage of the proposed technique is that we achieve NAV distribution in broadcasting. Thus we protect the network from collisions and dropped packets due to the excess of the number of retransmission attempts, allowed by the standard. The second and most important advantage is that we minimize the effect of the collision when this happens. A collision happens when two or more STA randomly select the same backoff number and complete the backoff process simultaneously. In such a case, if classic 802.11 is implemented and the forthcoming packet is a broadcast packet, the inevitable collision that follows will last longer period than if the proposed technique is implemented (Fig 3.9). In the proposed model, the collision will take place between the CTS-to-Self frames instead of data packets and the jam in the network will last significantly less. As long as additional traffic is added in the network due to CTS-to-self, the actual number of collisions is expected to rise. However, those collisions will be mostly between CTS-to-self frames and they are not expected to affect throughput performance.

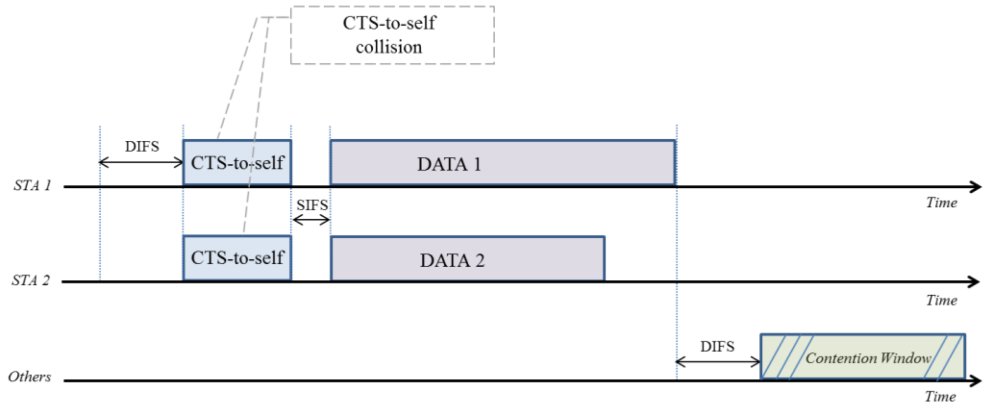

Fig 3.9: CTS-to-Self collision

3.6 Simulation Characteristics and results

The OPNET simulation characteristics of this study are described in this section. The simulation is based on IEEE 802.11g Physical characteristics with a bit rate of 54 Mbps. The wireless STAs are forming an ad-hoc network with its population to increase gradually from 5 to 60 STAs. The packet generator in each STA is set to create a data load of 256 Kbps which is a satisfactory average data load, produced by the most commonly used media compression codecs. The final payload in the wireless medium is found to be 320 Kbps due to the MAC overhead. All STAs in the network work in a saturated condition. That means that they always have packets to transmit. The generated data load remains constant in all simulations but as the number of medium access attempts for each STA depends on the packet size, three different packet sizes (2048, 1024 and 512 bytes) are used for each population increase. This allows us to test the effect of the expanding use of the CTS-to-self protection mechanism in various traffic conditions. The custom "collision counter" statistic mentioned in section 3.4.2, is used also here in order to accurately measure the number of collisions. Table 3.1 contains the settings for all the packet size simulations.

Packet Size	2048 (bytes)	1024(bytes)	512(bytes)
Start Time	Norm. (0.01, 0.0001)	Norm. (0.01, 0.0001)	Norm. (0.01, 0.0001)
On-State	Constant (120) sec	Constant (120) sec	Constant (120) sec
Off-State	Constant (0) sec	Constant (0) sec	Constant (0) sec
Interarrival Time	Constant (0.05) sec	Constant (0.025) sec	Constant (0.0125) sec

Table 3.1: Simulation settings

3.7 Results

The simulation runs for 2 minutes. This is enough time for the network to reach a steady state where accurate measurements can be taken. The statistics collected during the simulation are: Throughput, Overall End to End Delay and the Number of Collisions encountered in each STA. For each increase of the population in the network a separate simulation is performed. In order to ensure accuracy, each simulation runs three times using a different “seed” number. The final results are the average values from the three simulations.

3.7.1 Throughput

Throughput measurement represents the total number of bits (in bits/sec) forwarded from wireless LAN layers to higher layers in all wireless LAN nodes of the network. It is important to note here that when a packet is broadcasted in a wireless network, it will be received from all STAs except the one which transmit it. For a network with *n* STAs and each STA receiving a stream of *A* bits/sec, the total measured throughput in the network will be:

$$Throughput = A1 + A2 + \cdots + An = \sum_{i=1}^{n} Ai \ (bps) \tag{3.7}$$

That explains why throughput gets values much higher than the maximum nominal bit rate of the network which in our case is 54 Mbps.

The graphs in figures 3.10, 3.11 and 3.12 illustrate the throughput performance of the network for different data packet sizes, with and without the use of CTS-to-Self as a protection mechanism. It is clearly shown that using the proposed CTS-to-Self protection technique, a better throughput performance can be achieved. However, the improvement is greater when large packet size is used and it is more visible when the number of simultaneously broadcasting STAs increases (Figures 3.11 & 3.12). This is because when a large packet size is used, for each packet that is protected using the CTS-to-Self mechanism, the amount of data that is successfully transmitted is higher. Also, for the same payload in the network the number of medium access attempts are less, which makes the propose protection technique, more effective. The immersion observed in figure 3.12 is essentially the saturation of the wireless medium caused by the excessive number of medium access attempts.

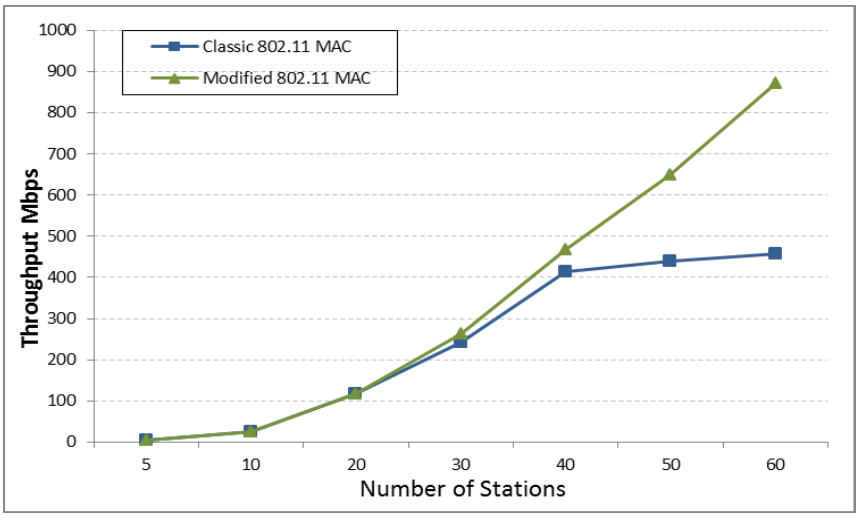

Fig 3.10: Throughput Performance for Packet Size 2048 bytes

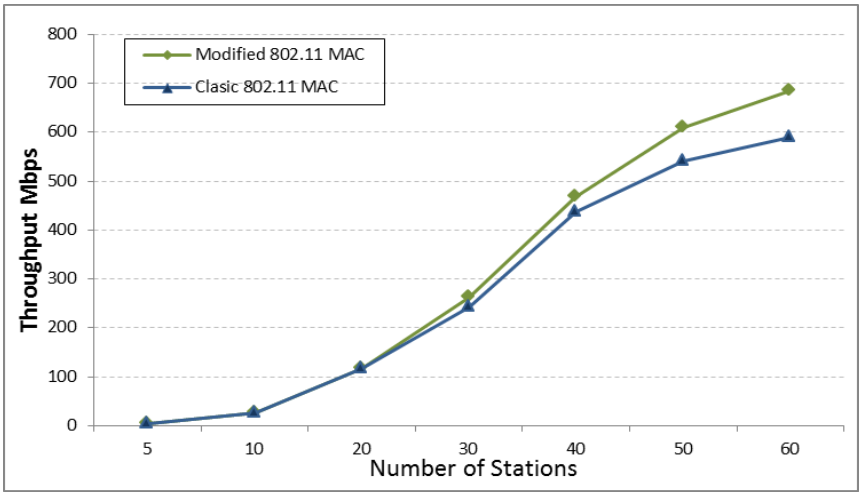

Fig 3.11: Throughput Performance for Packet Size 1024 bytes

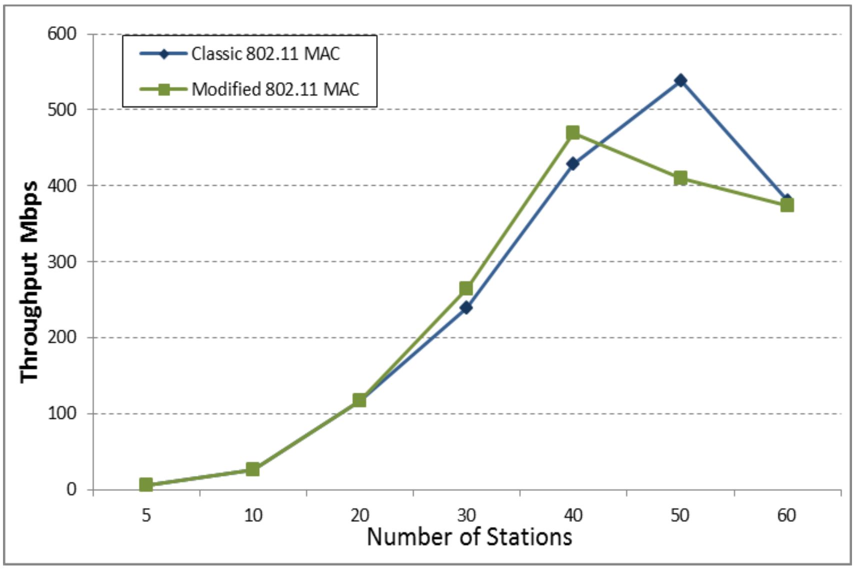

Fig 3.12: Throughput Performance for Packet Size 512 bytes

This excessive number of attempts causes equal number of random backoff processes, which dramatically increase delay and causes packet dropping due to buffer overflow. It is expected to happen first in the modified rather than the classic model as the use of CTS-to-self creates additional traffic which in the case of small packet size became significant.

As a conclusion, we can say that the proposed protection mechanism for broadcasting in wireless networks by using modified CTS-to-Self control messages can increase throughput in the network if is applied in combination with large data packets.

3.7.2 End-to-End Delay

Figures 3.13, 3.14 and 3.15 are illustrating the average end-to-end delay of all packets received by the wireless LAN MACs of all WLAN nodes in the network and forwarded to the higher layer.

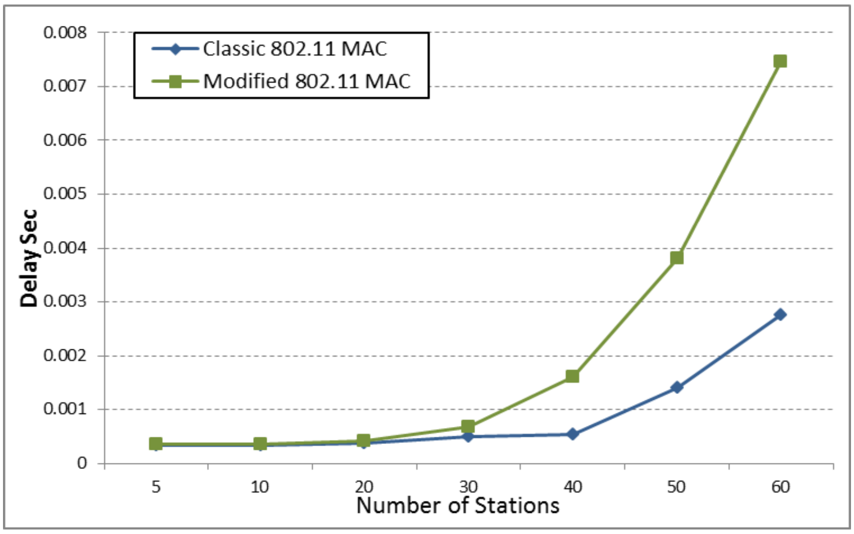

Fig 3.13: End-to-End Delay for Packet Size 2048 bytes

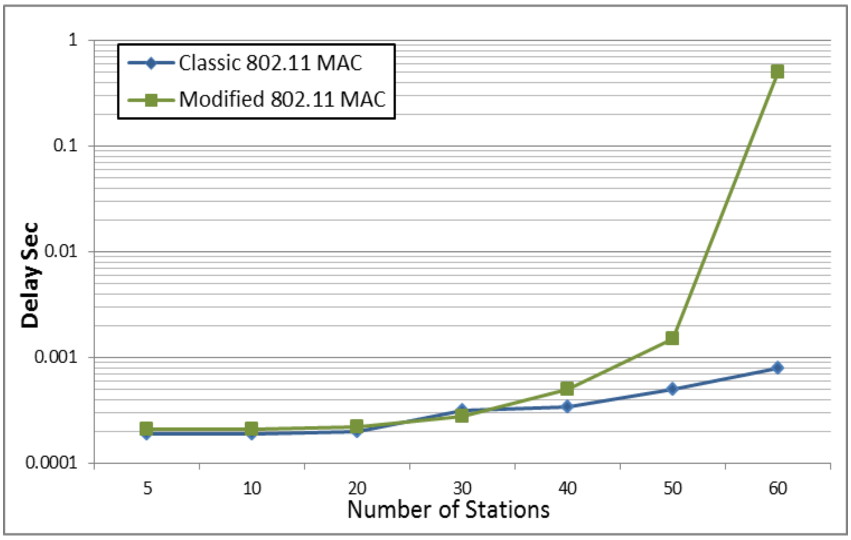

Fig 3.14: End-to-End Delay for Packet Size 1024 bytes

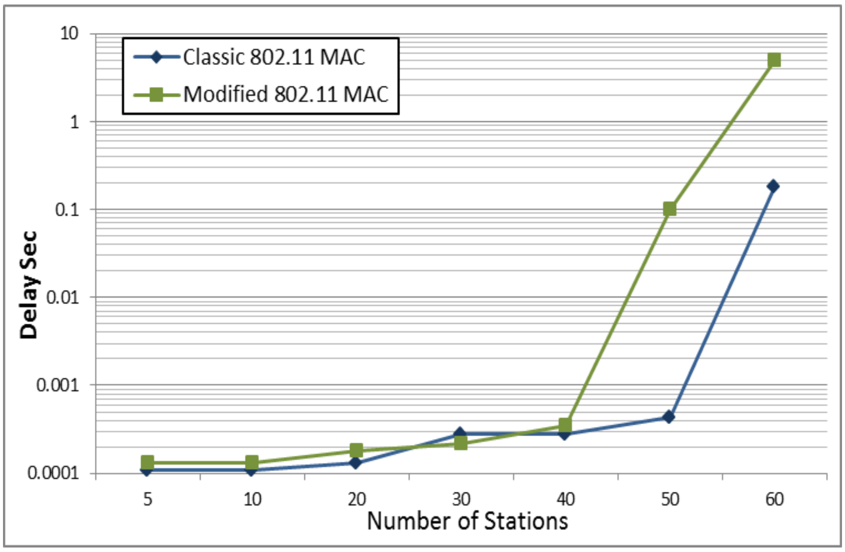

Fig 3.15: End-to-End Delay for Packet Size 512 bytes

It can be seen that the use of large packets gives the best performance when it comes to delay (Fig 3.13). As it is expected, the delay reaches higher values due to the additional CTS-to-Self transactions however remains in acceptable levels even for real time media application. Decreasing packet size, the overall delay increases. In the case of 512 byte packet size are becoming significantly high especially when the number of broadcasting STAs Exceed forty (Fig 3.14). In figures 3.13 and 3.14 a logarithmic scale is used for the time axis in order to illustrate the wide ranges of delay values.

3.7.3 Average Number of Collisions

A particularly interesting finding, resulting from this study is described in this section. It actually proves the claim in 3.5 that the use of a CTS-to-Self control message prior to every broadcasting packet will increase the number of collisions but will also increase throughput. This is, as it is expected, more visible when large packet sizes are used. Having an excessively small CW in broadcasting it is expected that many stations will complete the random backoff at the same time; especially in a busy network. Scheduling a CTS-to-Self before every data transmission, we ensure that most of the collisions will happen between CTS-to-Self frames instead of data packets. For a 2048 bytes packet the duration of a collision lasts approximately 150 times more than a collision between CTS-to-Self frames. Thus the time spent in a collision between CTS-to-self messages affect less the overall performance of the wireless network than a collision between data packets.

In this simulation, the “collision counter” custom statistic described in 3.4.2 is been used. Figures 3.16, 3.17 and 3.18 shows the average number of collision per STA, as the number of STAs increases. In figure 3.16 it is shown that using the modified IEEE 802.11 MAC, for a relatively small increase of collisions in the network a significant increase of throughput is achieved (Fig 3.10). When the packet size decreases, the number of collisions increases (Fig 17).

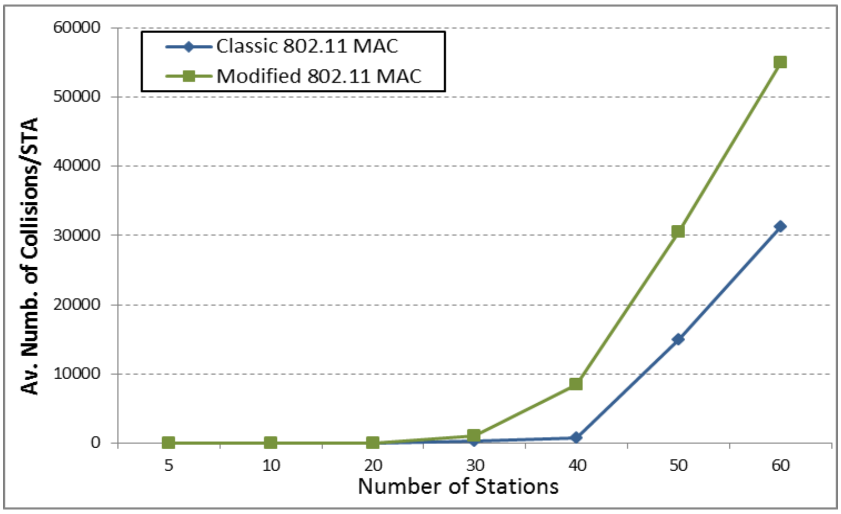

Fig 3.16: Average Number of Collisions per STA for Packet Size 2048 bytes

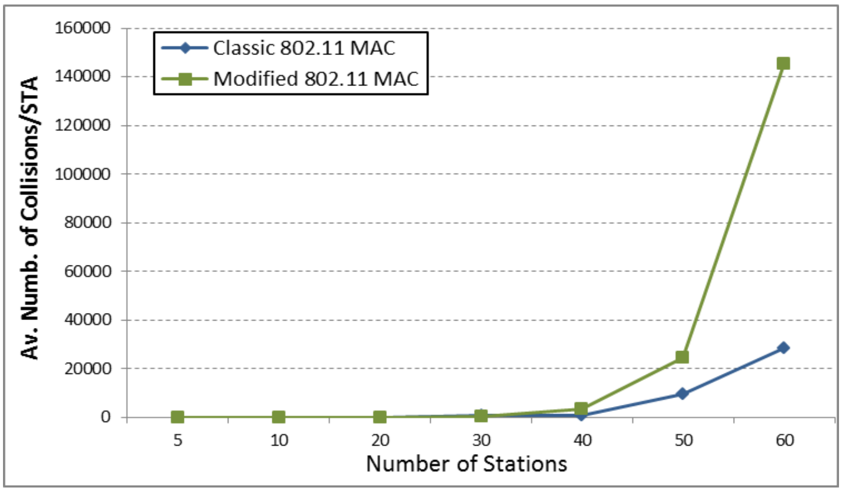

Fig 3.17: Average Number of Collisions per STA for Packet Size 1024 bytes

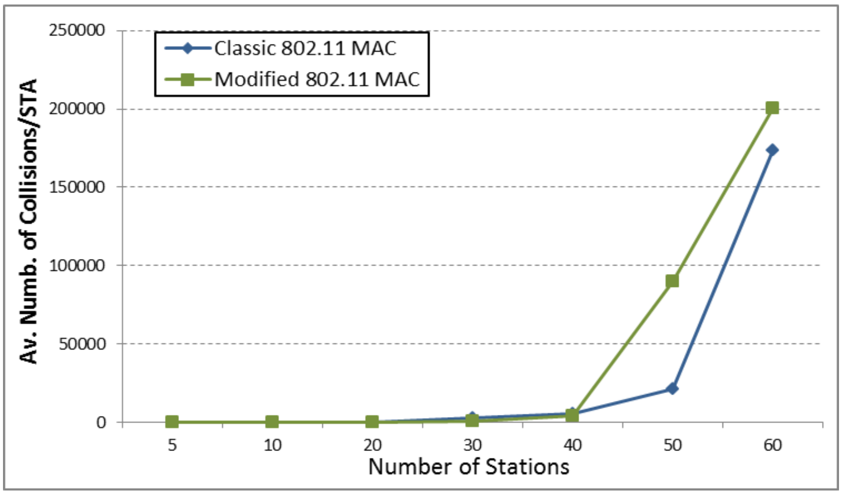

Fig 3.18: Average Number of Collisions per STA for Packet Size 512 bytes

This affects the overall delay and consequently decreases throughput performance. Finally, when the packet size further decreases to 512 bytes, the number of collisions increases dramatically for both the classic 802.11 and the modified MAC (Fig 18).

3.8 Summary

In this chapter we are have been investigating the possibility of improving the performance of broadcasting in a saturated IEEE 802.11 ad-hoc network using the CTS-to-Self protection mechanism. A saturated wireless network can satisfactory represent the scenario according to which, many or all STAs in the network are broadcasting real time audio data. CTS-to-self is a similar to RTS/CTS technique used by the 802.11 standard only for mixed-mode environments where ERP and HT devices coexist with legacy 802.11 technologies. A CTS-to-self has the regular CTS packet format and it is sent by a STA with destination address its own address. This is an alternative way to distribute NAV information in order to avoid collisions.

We extend this idea using this control message in a heavy traffic broadcasting environment where no other protection mechanism can be used. The target is to distribute network allocation information and at the same time to limit the effect of collisions in the network. For this reason the classic 802.11 MAC is appropriately modified and various types of data traffic is applied using different packet sizes.

As it is shown from the simulation results that this technique can significantly improve performance when many saturated STAs are broadcasting in an ad-hoc network. However this improvement can be achieved mainly when large size packets are used. For throughput-sensitive applications a combination of the proposed protection technique with the appropriate packet size can guaranteed reliable broadcasting, adding a small delay is also acceptable for time-sensitive applications. When packet size becomes small the number of packets needed to maintain the same bit rate increases. In this case, the additional traffic caused by the CTS-to-Self messages reduces the performance of the network.

The proposed technique that has been studied in this chapter can improve the performance of broadcasting in a wireless ad-hoc network for small number of broadcasting STAs and type of data that are tolerant with small delays. It can be easily used for audio and video streaming where buffering techniques can be applied to alleviate the delay problem. When it comes to wireless audio networks, the proposed technique can have only an ancillary use. Audio networks require not only lower latency but also higher throughput performance.

In the next chapter our effort focuses on improving overall throughput in broadcasting of audio data over an IEEE 802.11 network.

Chapter 4

Exclusive Backoff Number Allocation algorithm (EBNA)

4.1 Introduction

As it has been observed from the previous chapter, throughput-demanding applications cannot be served using a protection mechanism in the case of multiple broadcasting in a wireless ad-hoc network**.** The modified CTS-to-self mechanism can have an ancillary use but still an effective, broadcasting oriented collision avoidance mechanism, must be developed. To achieve this, the special characteristics of a wireless audio network must be taken into consideration, as analysed in section 2.9. The most crucial of those, at this state of our research, is the finite number of STAs in the network. The idea behind the initial design of the IEEE 802.11 standard was that theoretically, there is no limit in the number of the STAs that constituting the network, and also that this number can be dynamically changed, without affecting network's overall operation. This was based on the assumption that not all STAs have always data to transmit and also that applications such as *ftp* and *http* are, to some extent, delay tolerant. Audio networks are by definition different in both above characteristics. They consist always from a known finite number of STAs, *(instruments, microphones, speakers, consoles, and processing devices)*, which also produce continuous data, following most of the time a repeated pattern. Based on this, we design and implement in this chapter a novel MAC algorithm that is able to handle congestion process in a wireless ad-hoc network with finite number of STAs, causing theoretically zero collisions. This is achieved by linearly adjusting the size of CW according to the variation of the number of STAs in the network and also by allocating to each STA a unique backoff number, following a fair waiting scheme.

The rest of this chapter is organized as follows. Initially, the idea and the operation of the Exclusive Backoff Number Allocation algorithm (EBNA) is described. Then, the live music oriented data traffic model, proposed for this research, is analysed. The implementation of the EBNA algorithm and all the necessary modifications in OPNET are also thoroughly described. Finally, the simulation results are presented and evaluated.

4.2 The Exclusive Backoff Number Allocation algorithm, (EBNA)

As it is discussed in section 2.10.3, collisions in 802.11 MAC occurs when to or more STAs complete their backoff count down during the same time slot and start transmitting simultaneously. This is something that cannot be prevented in DCF. The standard instead tries to minimize the probability of collision using positive ACK for each successfully transmitted packet. If an ACK fails to return after a unicast transmission, the sender assumes that the network is congested and packet was collided. In this case the packet is retransmitted using double CW size. Thus the probability of collision decreases with a corresponding increase of delay. In broadcasting this technique cannot be implemented as there is not a chance for a positive ACK. Especially in real-time media broadcasting, a possible implementation of an ACK mechanism will lead to very long packet delivery delay, not suitable for this type of applications.

It results from the above, that if STAs perform the backoff process using each of them a different unique number there will be no chance for collision. However this simplified hypothesis raises two main issues. First, if the number of STAs varies, the CW cannot remains stable as long as in order to satisfy the above hypothesis the size of CW must be greater than or equal to the number of STAs in the network for any given time. Secondly, if one and only one unique backoff number is allocated to each STA, this will lead to an unfair distribution of waiting times, favouring the STAs with the lower numbers over others with higher numbers.

The EBNA algorithm proposed in this chapter, assigns exclusive backoff numbers to each individual SAT in the network resolving at the same time the above described issues. Before proceeding with the details let us assume a wireless ad-hoc network where during the association process each STA obtain a station ID (STID). In addition, each STA knows the total number of STAs (No_of_STAs) in the network at any given time.

It is important to note that we are not developing the association process here. The aim of this research is to investigate and propose solutions for the fundamental problem of reliable broadcasting of audio data in wireless ad-hoc networks. There are several techniques to obtain a STID and also the population of a wireless network during the association process and distribute these information using beacon messages but their analysis is beyond the scope of this thesis.

For any given time, each STA has available the values of the two variables STID and *No_of_STAs*. The size of CW is being given by:

$$CW = 2 \times No\_of\_STAs \tag{4.1}$$

This CW is divided into two equal groups according to the following relation:

$$group1 \leq \frac{No\_of\_STAs}{2} \tag{4.2}$$

$$group2 > \frac{No\_of\_STAs}{2} \tag{4.3}$$

For each transmission attempt for a variable named *Group*, the algorithm generates a random value between 1 and 2 with equal probability. According to the outcome, the algorithm chooses for this attempt the appropriate group from the CW. If *Group*=1, *group1* is selected, if *Group*=2 *group2* is selected. In the case that *group1* is selected, the number of slots the STA has to backoff is equal to its ID. In the case that *group2*, the value of backoff slots is symmetrically opposite to the first one, as we can see in figure 4.1. The backoff values are allocated to the station according to the following:

$$Group = 1, \qquad Backoff\_Slots = STID$$

$$Group = 2, \qquad Backoff\_slots = [(No\_of\_STA\,) \times 2) - STID] + 1 \qquad (4.4)$$

For example, for a network with 10 STAs the STA with STID=2 will constantly select randomly backoff values between the integers 2 or 19 following a normal distribution. Similarly the STA with STID=6 will select backoff values between 6 and 19. Figure 4.2 shows the EBNA pseudo-code.

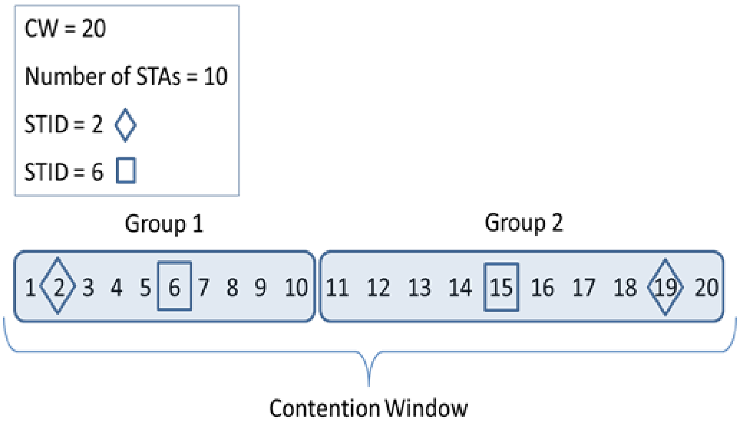

Fig 4.1: The EBNA algorithm (example for STID=2 & STID=6)

```
No_of_STAs=10    % Total number of STAs in the network
STID=2           % The ID of current STA

Group=rand (1, 2)
If Group=1
      Backoff_slots=STID
else
      Backoff_slots=No_of_STAs * 2 – STID+1
end If
```

Fig 4.2: EBNA algorithm pseudo-code

This simple algorithm allocates to each STA in the network a unique pair of backoff numbers. Since the selection of the group is based on a normal distribution and each pear is symmetrically opposite; in the long run, the average of waiting time for all STAs is the same and equal to CW/2. This ensures fairness among all stations within the network.

4.3 The music audio data traffic model

In order to emulate a professional live audio environment it is significantly important to understand the form of live music audio. The form of the audio produced from an instrument or a microphone during the musical performance is totally different from the one we meet in a mixed sound track. Live music performance mainly produces a monophonic audio that is not continuous but contains gaps that sometimes are considerably long. In figure 4.3, track “A” illustrates the waveform of a song's audio, while track “B” shows its corresponding vocal track. It is shown from the waveform in figure 4.3.*B* that this audio source, which in this case is a microphone, generates bursts of data rather than a continuous data stream.

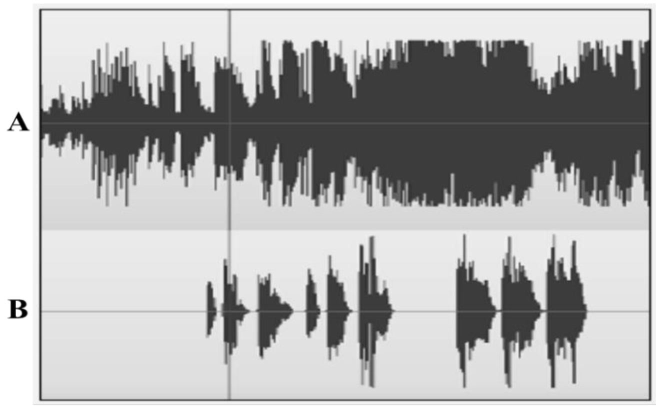

Fig 4.3: Voice track and mixed audio of a song

In order to realistically emulate the behaviour of a multi-broadcasting wireless audio network, a generic traffic model is proposed in this thesis. This traffic model is based in an average music tempo of 120. Before going into details that defend our decision, we must first give the definition of tempo marking. A tempo marking indicates the actual duration of the time values during music execution [67]. The number (120 in our case), indicates the beats per minute or bars per minute (bpm) that has to be executed and thus it defines the speed of execution. This means that a particular note value (for example, a quarter note ♩) is specified as the beat, and that the amount of time between successive beats is a specified fraction of a minute [68]. Relevant research based on perceptual experiments and analysis of big amount of data shows that the most preferable tempo among music pieces is around 120. A significant initial work from P. Fraisse is presented in [69]. Dirk Moelants in [70] [71] and together with Martin McKinney in [72] performs a series of experiments among musicians and non-musicians audience and analyses an enormous amount of data (74042 pieces) in order to extract a generic distribution of tempo in music. It appears that this tempo can be associated with the most natural speed to perform simple repetitive movements. Researchers believe that the human body acts like a resonator that starts to move under the influence of an external force, the beat of the music. Tempo 120 seems to be the resonance frequency affecting more than any other tempo the rhythmic reaction of the human body. Figure 4.4 shows the distribution of tempi as it is perceived in four sets of musical pieces. These sets contain: a) existing perceptual data using 'bpm-lists' available on the internet, b) experimental tempo collection using random music heard on the radio, changing program after 40 tabs, c) experimental tempo collection A selection of hits from the period 1960-90 as collected in a CD-series giving a historical overview of popular music and d) experimental tempo collection from music taken from specific, but divergent styles, ranging from renaissance polyphony to modern jazz [70].

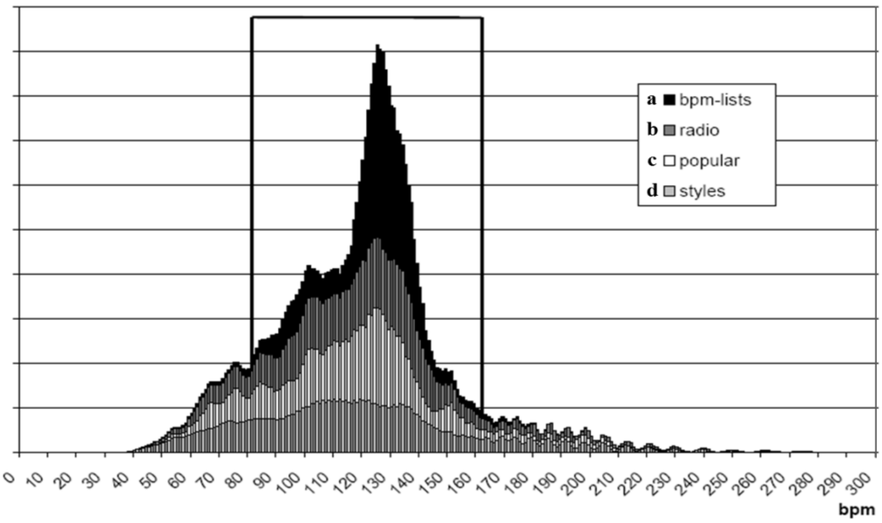

Fig. 4.4: Distribution of the tempi as perceived in four wide sets of musical pieces [70]

The traffic model proposed here generates burst of data based on a 120 bpm repeated pattern. 120 bpm means that 120 notes of ¼ (♩) has to be played in one minute. This means that a note has to be played every 0.5 seconds. However, this does not mean that each note should last 0.5 seconds. Depending on the type of sound source, music notes have various durations. The total duration of a sound produced from an instrument or vocal, consists of the sum of a series of separate time durations, these are the *attack*, *decay*, *sustain* and *release* time, as it is shown in figure 4.5. In some instruments like percussions for example, sustain and release can be particularly small, while in other cases, like wind instruments they can be

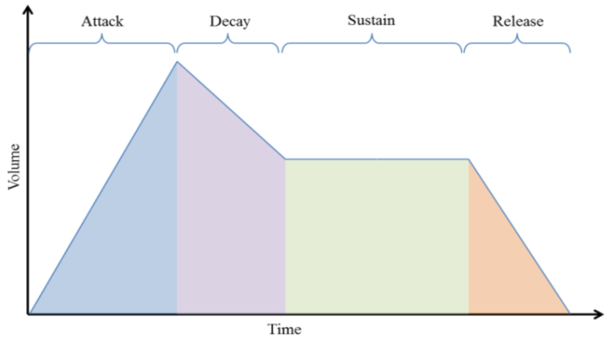

Fig. 4.5: The sound envelope

very long. Generally, the duration of time segments which form the sound envelope, depends on the nature of the source generating the sound, but also on the music performance. For this reason we use in our model an intermediate value for the sound envelope. Since we have a time space of 0.5 seconds from one note to the next one, we create a data payload during the half of this period. With this method we create a pattern of data bursts with duration of 0.25 seconds, followed by 0.25 seconds of silence. This data payload is based on a 16 bit/44.1 KHz sampling rate (PCM, no compression) known as CD quality audio. That gives a bit rate of 0.67 Mbps, which gives consequently for a packet size of 2200 bytes, an interarrival time of 24.3 msec. This traffic is to be applied to a WLAN with the parallel use of CTS-to-self protection mechanism. Therefore, a big packet size is used in order to take full advantage of this protection mechanism, as it is analysed in section 3.7.1. We also define the start time of data generation based on a normal distribution with mean outcome 1 and variation 10 msec, to emulate the stochastic nature of musical performance. The interarrival time is an OPNET attribute that define the time interval between two consecutive messages and it is set to be constant. The resulting load transmitted by each WSTA is not constant however, because of the normal distribution set in the start time attribute. It is approximately 383Kbps/STA, which is 48 Kbps higher than the generated load due to MAC overhead. All data traffic generation parameters are listed in Table 4.1.

Attributes	Values
Start Time	Normal Distribution (1, 0.01)
On-State	0.25 sec
Off-State	0.25 sec
Interarrival Time	Constant Distribution (24.3 msec)
Packet Size	2200 bytes

Table 4.1: Traffic Generation Parameters

4.4 Implementation of the EBNA algorithm in OPNET

In order to implement the EBNA algorithm in OPNET, the standard MAC process in the *wireless station node model* was modified. All the changes listed below [73]:

i. First, a custom *Station ID* attribute is created in order to allow users to give to each station a unique ID number. This attribute is set manually here. In a future implementation this information will be obtained automatically from each STA upon joining the network. Custom attributes is another important future in OPNET. Details regarding the custom attributes created for this project are given in Appendix A.
ii. All modifications described in chapter 3 regarding the extended use of CTS-to-self message as a protection mechanism in broadcasting, are also implemented here.
iii. Finally, the MAC process and most specific the random backoff algorithm is replaced with the EBNA algorithm as described in section 4.2.

4.4.1 Modifying the MAC process

For this study OPNET's *wlan_station_adv* node model is used. The model is thoroughly described in section 3.4.2. The MAC process is implemented inside the *wlan_mac* process which is a child process of the *wlan_dispatch* process as it is shown in figure 4.6-c and d. *Wlan_mac* is a relatively complex process handling the classic PCF and DCF medium arbitration methods of 802.11 standard. The hybrid coordination function (HCF) is implemented in independent process. *Wlan_mac* consist of a group of forced and unforced states that mainly implement the logic of 802.11 standards by calling the appropriate functions. The main body of the code is found in the function block (FB) where these functions are defined. The random backoff algorithm is implemented in the forced state called *BKOFF_NEEDED* which is located in the middle of the state transition diagram of the *wlan_mac* process (Fig 4.6-*d*). In this study the code in the *Enter Executives* has been modified in order to allocate backoff slots according to the proposed EBNA algorithm. A detailed description of the modified code is given in Appendix B.

4.4.2 Simulation characteristics

The simulated WLAN is built on the IEEE 802.11g PHY, with a bit rate of 54 Mbps. The topology is based on an ad-hoc network in a single BSS, with the WSTAs located randomly in a 30×40 m surface. The number of WSTAs is gradually increased up to 70 during the study. The simulation duration is 2 minutes and the traffic is generated according to the "music audio data traffic model" proposed earlier in this chapter. The transport method is broadcasting. Three separate simulations have been conducted for each scenario where all stations were relocated and also a different seed number has been set. The presented results are the average values, in the cases where significant differences occurred.

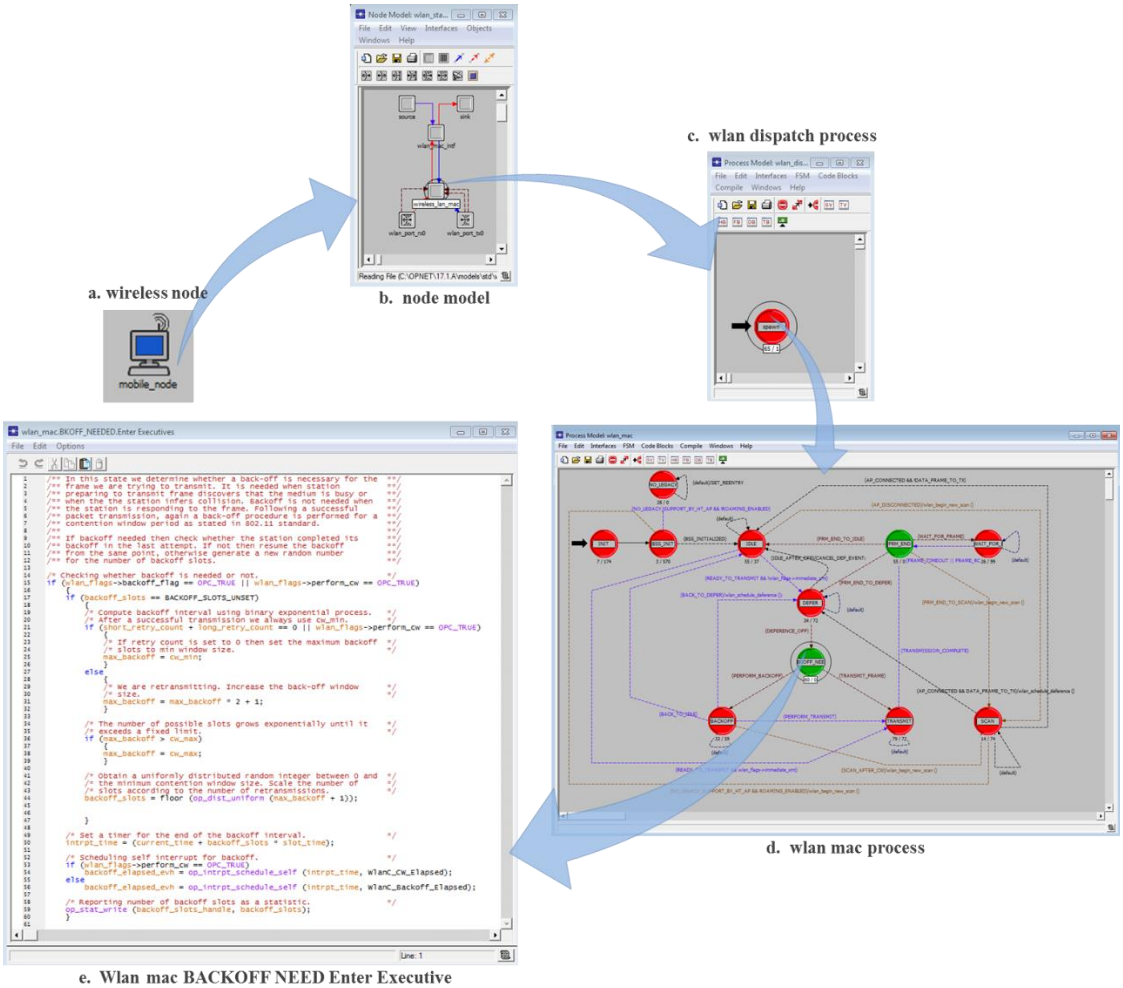

Fig. 4.6: The *wlan_mac* process in OPNET wireless model

4.4.3 Validation of the modified MAC process

Before proceeding with the analysis of the results we must first validate the proper operation of the modified model. We already verified the proper operation and effectiveness of the modified CTS-to-self mechanism in chapter 3. Therefore, our main concern here is the EBNA algorithm. For that reason a custom statistic has been created and collected during the simulation. This statistic monitors the values of backoff slots assigned by the MAC process during the simulation. According to the EBNA logic, a unique pair of integers, directly related to its ID, must be assigned to each STA. In addition, for each transmission attempt the backoff value must be randomly selected from this unique couple with equal probability. In figure 4.7 we can see the sequence of values that the variable *backoff_slots* gets for a short period of time (approximately 1 sec), within the simulation time.

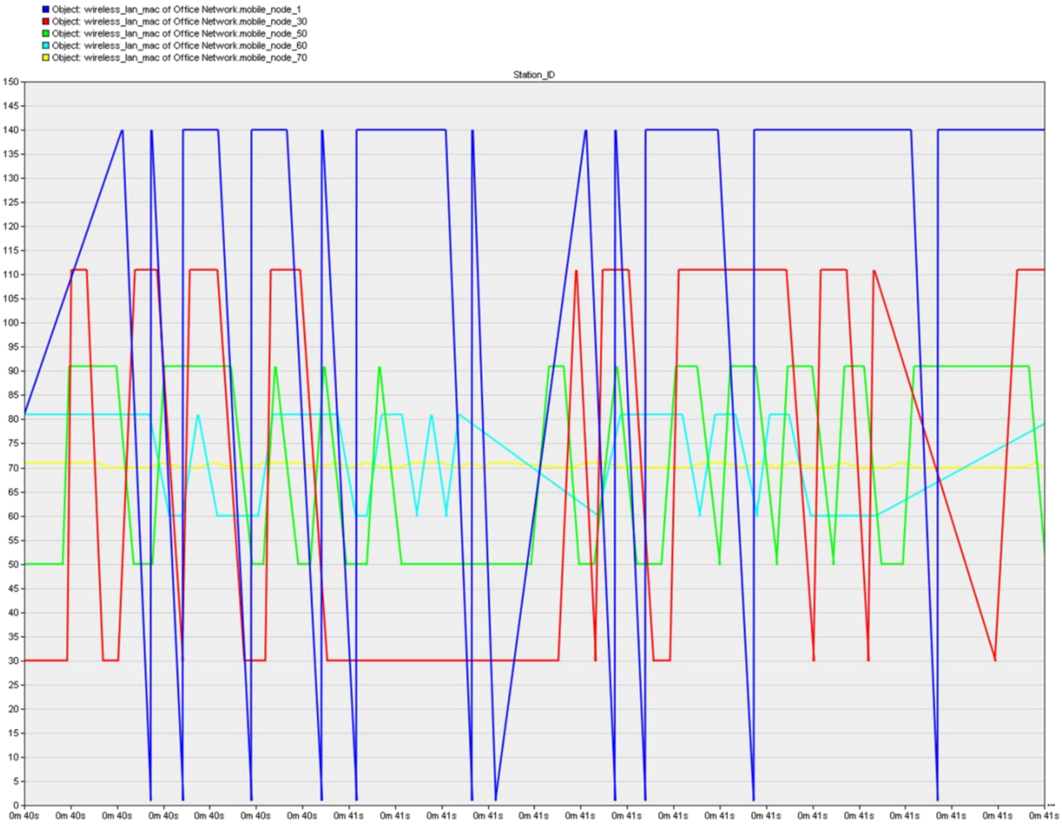

Fig. 4.7: Validation of the EBNA algorithm

This graph is taken from the “70 STAs” scenario and represents the allocation process of the backoff time (in time slots) for STA-1, STA-30, STA-50, STA-60 and STA-70. It is important to monitor the STAs with the lowest and the highest STID and also some intermediate, in order to have a full picture of the algorithm’s operation.

According to the EBNA logic, if the network consists of 70 STA the CW will be 140, (equation 4.1). The STA with STID=1 will randomly take the backoff value 1 or 140, (equation 4.4). Correspondingly, STA with STID=30 will take the backoff value 30 or 111, STA with STID=50 will take the backoff value 50 or 91, STA with STID=60 will take the backoff value 60 or 81 and STA with STID=70 will take the backoff value 70 or 71. Figure 4.7 shows that the simulation gives us exactly these values and thus verify the proper implementation and operation of EBNA algorithm.

Another important issue regarding the proper operation of the modified MAC process is the fairness among all STAs in the network. Fairness is a fundamental parameter in the design of communication protocols. When EBNA algorithm is implemented it is important in the long run, for all STAs to be assigned equal waiting times during the backoff process.

Averaging the values shown in figure 4.8 we get the mean waiting time (in time slots) for STAs 1, 30, 50, 60 and 70. The graph in figure 4.8 shows that, after the network enters its steady state, all STA are assigned on average, an equal number of time slots but never the same number between two or more STAs, avoiding in this way collisions.

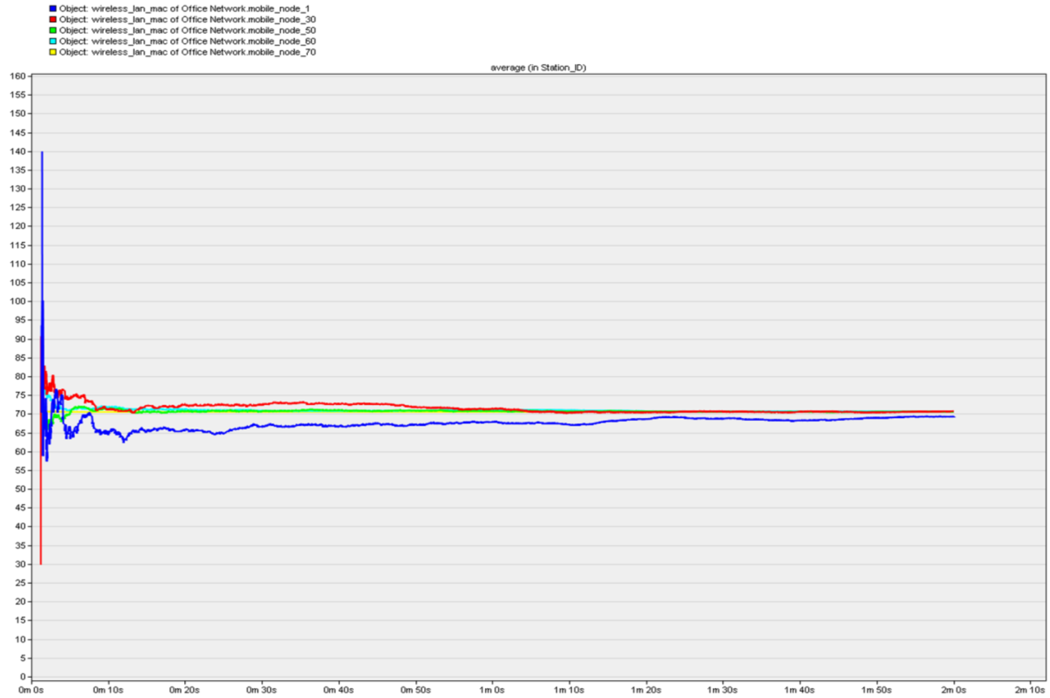

Fig. 4.8: Mean average of backoff values

4.5 Results, (presentation and analysis)

Throughput is the parameter that affects audio quality more than any other, in broadcasting. Undeliverable packets cause irreplaceable gaps in audio at the receiver. As there is no way to detect and retransmit these packets, due to the nature of broadcasting, throughput improvement is the only reliable solution to this problem.

The next very important parameter regarding real-time audio delivery is delay. In a wireless ad-hoc network, delay is caused by three main reasons. First is queuing delay, when packets are waiting in the transmitter's buffer for access to the medium and/or other packets to be delivered. Second is the transmission delay. And finally the overall delay caused by potential retransmissions. In case of broadcasting, time spend in the buffer queue is the main cause of delay and is directly related to the random backoff procedure. Throughput and overall delay measurements are presented and analysed below.

4.5.1 Throughput using EBNA

As it was discussed in 3.7.1, throughput measurements in broadcasting are giving values higher than the maximum nominal bit rate of the PHY in use. This is happening because each broadcasted packet reaches multiple destinations simultaneously and thus it is calculated as a successful delivery for several times. Figure 4.8 shows the throughput measurement in bits/sec for the first six scenarios for both the classic 802.11 MAC and the EBNA modified MAC process. The population of the WLAN is gradually increased in each scenario starting from 10 STAs and going up to 35 STAs. It is shown that throughput performance for both, classic and modified MAC increases equally as no collisions occur in the network. A 3D representation is used in figure 4.9, because graphs from classic and modified MAC overlap completely.

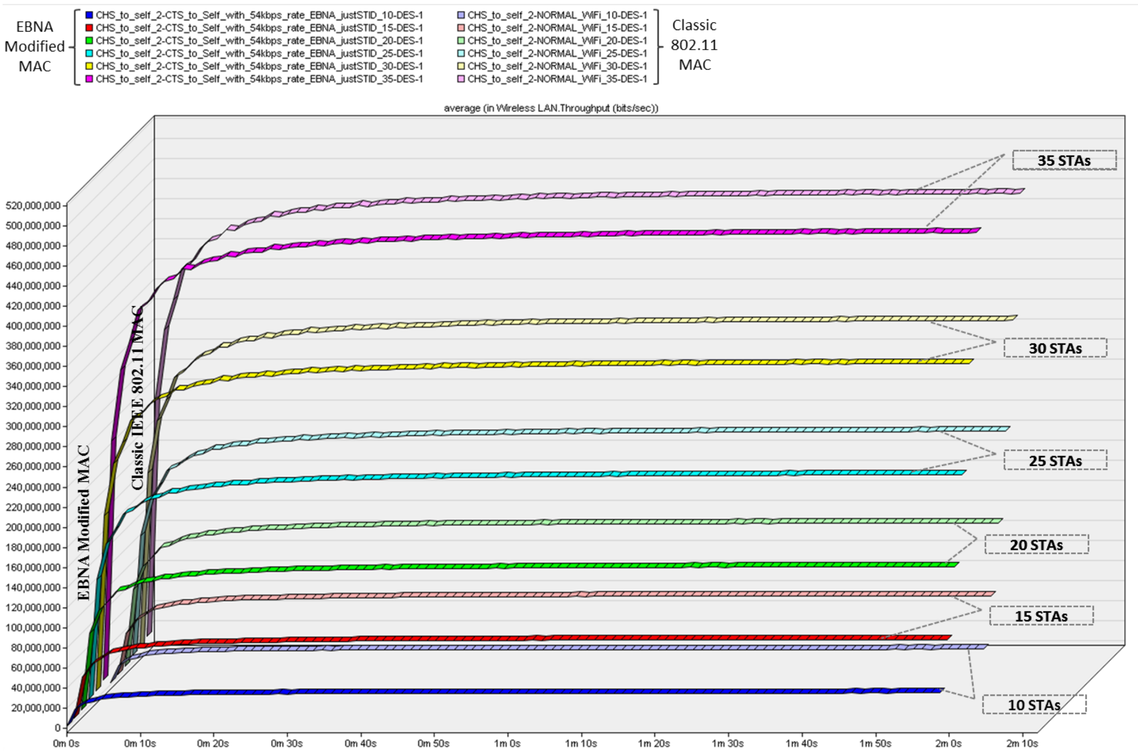

Fig. 4.9: Throughput performance for 10 to 35 STA scenarios - (classic and modified MAC)

Figure 4.10 illustrates throughput performance for scenarios with higher population that starts from 40 and increase by five up to 70 STAs. Here we can see that, as the number of STAs increases the difference in throughput between classic and modified MAC increases significantly. There is a subtle difference in the 40 STAs scenario but it increases to 120 Mbps in 60 STAs, 300 Mbps in 65 STAs and 440 Mbps in 70 STAs.

In order to understand throughput measurements, we must first calculate the maximum theoretical throughput expected in each case. Let us assume a wireless network with n STAs and each of them produces a data load of Ai bit/sec. If all transmissions are 100% successful, during a time period Δt, each STA will receive a total load given by the equation (4.5).

$$STA - Throughput_{(\Delta t)} = A_1 + A_2 + \cdots + A_{i-1} + A_{i+1} + \cdots + A_n = (n-1) \times A_i \qquad (4.5)$$

This is because each STA receives all data from all other STAs in the network except his own transmissions. Having n STAs in the network, overall throughput will be given by (4.6).

$$Overall - Throughput_{(\Delta t)} = [(n-1) \times A_1] + [(n-1) \times A_2] + \cdots + [(n-1) \times A_n]$$

$$\therefore \; Overall - Throughput_{(\Delta t)} = \sum_{i=1}^{n} (n-1) \times A_i = (n-1) \times \sum_{i=1}^{n} A_i \qquad (4.6)$$

In the case that all STAs producing an equal pay load A bit/sec, equation (4.6) can be reduced to (4.7).

$$Overall - Throughput_{(\Delta t)} = n \times [(n-1) \times A] \qquad (4.7)$$

Equation (4.7) gives the theoretical maximum throughput we can get during broadcasting in a wireless ad-hoc network in ideal conditions (all STAs in range and no collisions). Table 4.2 shows the calculated theoretical maximum throughput and also the experimental results for different network populations as derived from the simulation (figures 4.8 & 4.9). It also shows the percent of maximum theoretical throughput that finally succeed from classic and modified MAC, for each scenario.

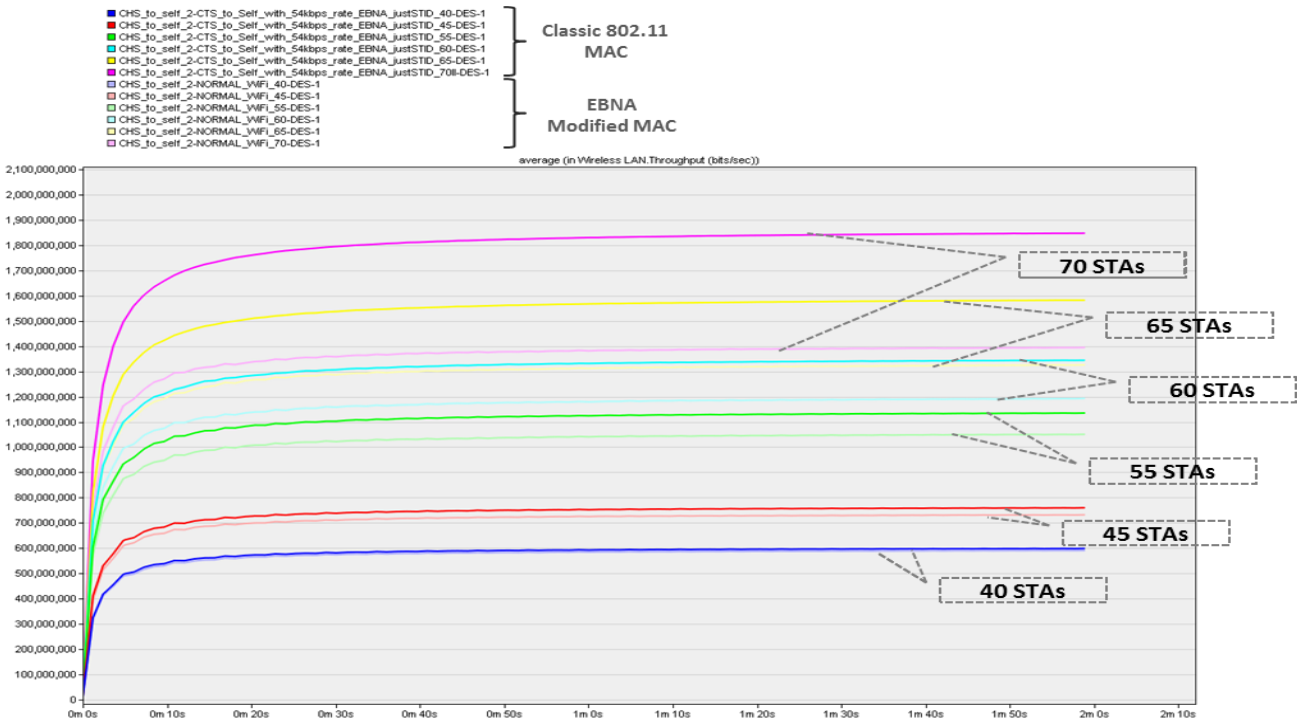

Fig. 4.10: Throughput performance for 10 to 35 STA scenarios - (classic and modified MAC)

Simulation Scenario	Number of STAs	Maximum Theoretical	Simulation-Classic 802.11 MAC (b/sec)	% of the Max Theoretical. Throughput	Simulation-EBNA Modified MAC (b/sec)	% of the Max Theoretical. Throughput
1	10	34,560,000	34,380,000	99.479	34,400,000	99.537
2	15	80,640,000	80,220,000	99.479	80,500,000	99.826
3	20	145,920,000	145,160,000	99.479	145,830,000	99.938
4	25	230,400,000	229,200,000	99.479	230,000,000	99.826
5	30	334,080,000	332,500,000	98.914	334,000,000	99.979
6	35	456,960,000	452,000,000	98.914	456,000,000	99.789
7	40	599,040,000	590,000,000	98.490	598,000,000	99.826
8	45	760,320,000	731,000,000	96.143	760,000,000	99.957
9	50	940,800,000	884,000,000	93.962	940,000,000	99.914
10	55	1,140,480,000	1,050,000,000	92.066	1,135,000,000	99.519
11	60	1,359,360,000	1,192,000,000	87.688	1,344,000,000	98.870
12	65	1,597,440,000	1,325,000,000	82.945	1,582,000,000	99.033
13	70	1,854,720,000	1,394,000,000	75.159	1,847,000,000	99.583

Table 4.2: Throughput results, (Max. theoretical vs classic 801.11 and EBNA modified MAC)

Plotting from table 4.2, the percent of the maximum theoretical throughput achieved by classic 802.11 and EBNA modified MAC (figure 4.11), we can see that as the number of STAs increases, classic 802.11 is not able to handle the increasing broadcasting traffic. The small size of CW causes an increase on the number of collisions and therefore decreases throughput. Instead, the EBNA algorithm by allocating exclusive backoff slots to each STA eliminate collisions and thus manage to successfully broadcast near 100% of the produced packets. The small drops in throughput after the 50 STAs scenario is caused by dropped packets due to buffer overflow at the sender.

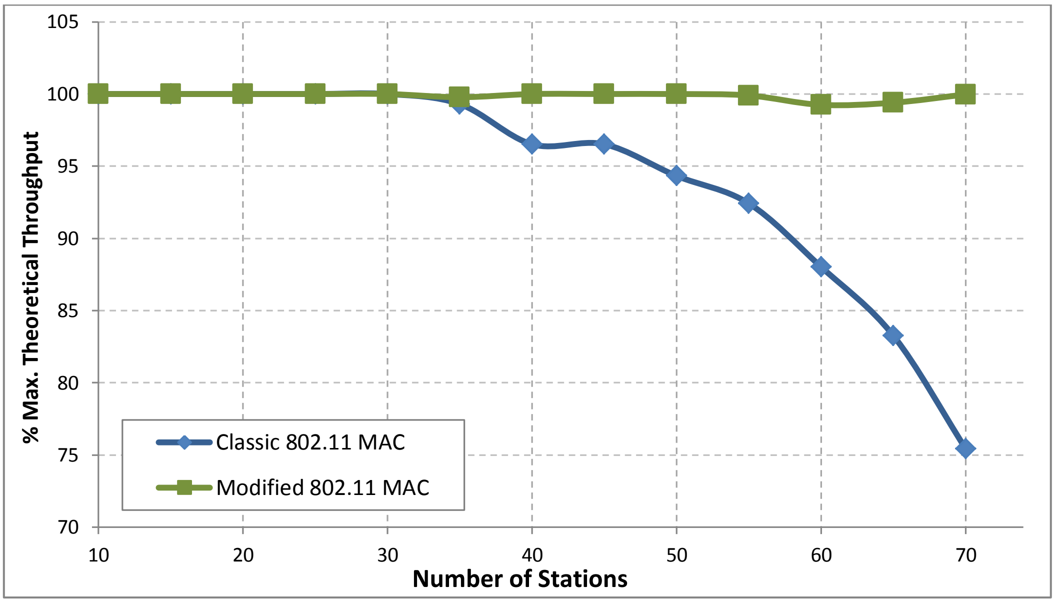

Fig. 4.11: Throughput performance, (Classic 802.11 vs EBNA modified MAC)

4.5.2 Delay

In IEEE 802.11 standard delay describes the overall time to send a packet from a source to a destination STA. In a single-hop ad-hoc network delay D is the sum of three main parts [74].

i. The queuing delay D_Q which represent the interval between the time the packet enters the queue in the transmitting STA and the time it reaches the head of the queue.
ii. The contention delay D_C which is the time spent from the STA to gain access to the wireless medium. This is caused mainly by the random backoff process and the virtual currier sense if it is applied.
iii. The transmission delay D_T which is the time needed in order for the whole packet to be transmitted including ACK and possible retransmission.

The total end-to-end delay D_{total} is given by:

$$D_{total} = D_Q + D_C + D_T \tag{4.8}$$

In the case of broadcasting in an ad-hoc network, there is no chance for retransmission. Assuming also that buffering and packet generation characteristics are identically set in all STAs, D_Q is expected to be equal for all transmission attempts for both the classic and the EBNA modified 802.11protocols. Therefore, all differences in delay, observed in this study between classic and modified 802.11 are due to variations in the contention mechanism and the transmission process.

Delay measurements in OPNET represent the end-to-end delay of all packets received by the WLAN MACs of all WSTAs in the network and forwarded to the higher layers. This includes the medium access delay at the transmitter and transmission delay. However, an objective comparison regarding broadcasting, between classic 802.11 and EBNA modified 802.11 protocols, based on delay measurements is not fully possible. OPNET measures the end-to-end delay only for packets that managed to be delivered. When big differences in throughput occur, which mean that collision rate increases; delay measurements cannot give us a representative picture of network’s operation.

However, delay is independently a major factor especially when it comes to audio networking and affects the applicability of the EBNA amendment. Figure 4.9 shows the measured delay for all scenarios, for both the classic 802.11 and EBNA modified 802.11 MAC processes.

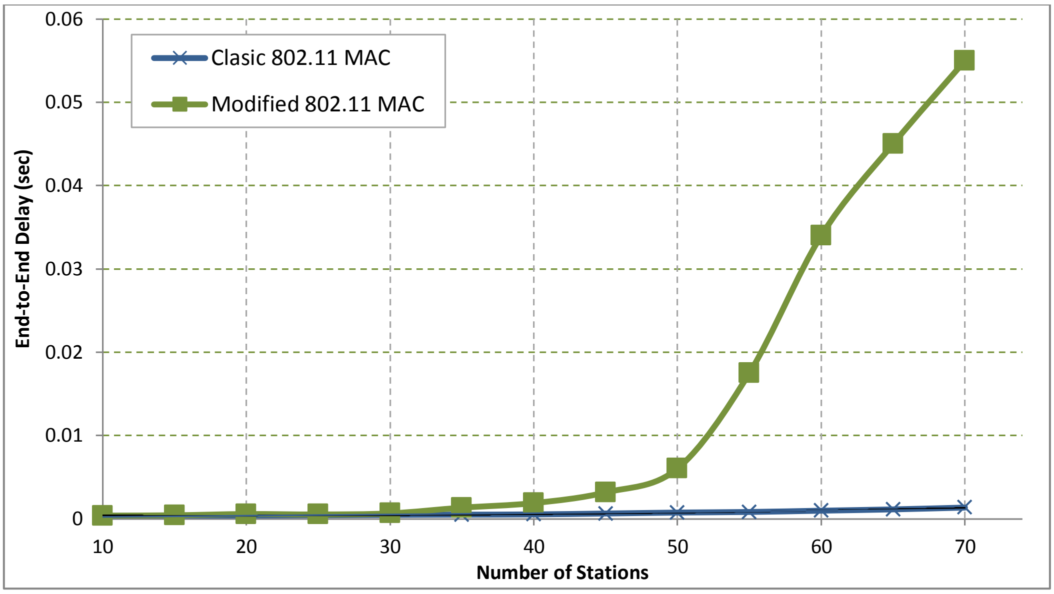

Fig. 4.12: Overall end-to-end delay for classic and EBNA modified 802.11 MAC

It shown from figure 4.12 that classic 802.11 maintains a significantly low delay, ranging from 0.3 to 1.3 milliseconds. The small increase in delay it is normal and it is caused by the busy network. When the number of STAs increases, a station, which is in the middle of the backoff process, has an increased probability to find the medium busy and thus to freeze the backoff process. This procedure, which is also known as *physical carrier sense*, increases D_C and therefore the total end-to-end delay D_{total}, (equation 4.8).

The proposed EBNA algorithm however adds two more parameters that affecting D_C. The first one is the implementation of a protection mechanism, based on the extended use of the CTS-to-self control message, which is discussed in detail in Chapter 3. This protection mechanism distributes a NAV and forces STAs to implement *virtual carries sense* together with the physical carrier sense. The implementation of virtual carrier sense reduces the probability of collision but also increases D_C. The second parameter is the linear increase of CW window. When the EBNA algorithm is used, an increase of the number of STAs causes a linear increase of the CW, (equation 4.1). A larger CW means that a greater number of time slots (on average), will be used during the backoff process. An increase of D_C is caused in this case and it is proportional to the increase of the size of CW. In addition, the use of CTS-to-self is also affecting D_T. According to the EBNA algorithm, the implementation of the proposed protection mechanism mandates the transmission of a CTS-to-self message prior to each packet transmission. This increases D_T by time equal to CTS transmission time plus a SIFS, (figure 3.8). All the above explain the significant raise of the delay, observed in figure 4.9, when the EBNA modification is used.

4.6 Summary

This chapter is dealing with the main issue, regarding the use of IEEE 802.11 technology in wireless audio networking applications, which is the improvement of throughput performance. The problem lies on the inability of the 802.11 MAC to handle multiple broadcasting networks due to its random backoff algorithm. For this reason, a novel medium access method is proposed. It is designed in a way that takes advantage of the special characteristics of audio networks such as the finite number of stations and their limited spatial expansion. The method allocates exclusive backoff numbers to each STA and thus excludes the occurrence of collisions.

In order to simulate and test the above method, an audio data traffic model is also proposed in this chapter. This model takes into account the specific form of audio, produced from instruments and vocals and also the digital audio qualitative requirements for professional level applications. Finally, it defines appropriate repeating patterns based on the related published research, in order to emulate musical performance.

The simulation was implemented in OPNET and a series of simulation performed in parallel for both the classic and the modified 802.11 MAC. The method was tested and validated for its proper operation and fairness. The results showed an exceptional behaviour of the proposed algorithm when it comes to throughput, not only in comparison with the conventional 802.11 protocol but also in absolute value, since it manage to reduce collisions at a close to zero level. However, a notable rise of overall end-to-end delay was observed. This resulting delay is suitable for many media applications but is marginally acceptable for audio networking applications. As it was discussed in 2.9, a potential wireless audio network must be able to support approximately 60 STAs with a delay close to 10 msec. Simulation results showed that using the proposed amendment we can support up to 52 STAs within acceptable level of delay.

In order to satisfy the demands of audio networks, further improvement in the proposed medium access method is needed. In the next chapter our effort focuses on improving delay performance, using our exclusive backoff number allocation (EBNA) idea, while maintaining throughput at high levels.

Chapter 5

A Hybrid-Exclusive Backoff Number Allocation algorithm (H-EBNA), with Traffic Adaptive capabilities

5.1 Introduction

A careful analysis of the simulation results in chapter 4 leads us into some interesting conclusions.

Initially we verify that the main cause of collisions in broadcasting is the random backoff process. Using the EBNA technique we can eliminate collisions and therefore maximize throughput. However, using EBNA we actually implement a linear increase of CW, proportional to the number of STAs in the network. Consequently, this leads to an increase of the average waiting time for accessing the medium, for each STA and thus raises overall end-to-end delay.

Another interesting observation is that classic 802.11 performs relatively well in both throughput and delay, when the number of the STAs in the network is small. When only few STAs broadcast simultaneously the probability of collision is low and as long as the CW is also small, classic 802.11 manages to achieve comparable throughput with the EBNA, but with a lower delay. It is important to note here that real-time audio can be tolerant to packet loss until certain limits. There are techniques that can cover up to 20% losses of the broadcasting data, depending on the compression technique and the quality demands [5]. Therefore, for real-time audio networks, an implementation with small, controllable packet loss and lower delay is preferable to a method with zero losses and greater delay.

The use of EBNA implies that all STAs are intending to broadcast at any given time and thus reserves for each one of them a unique, equally weighted pair of numbers within the CW. This keeps the size of CW constant and relatively big. However, musical performance has a strong stochastic nature. That means that there are time intervals where not all STAs have data to transmit. In those cases there is an unnecessary time spent in the backoff process. A proper exploitation of this phenomenon could help delay to remain in lower level. Monitoring the networks activity, we can use a statistical approach to define the actual active STAs and apply the EBNA technique to those only.

Taking into account the above elements we design and implement in this chapter a Hybrid-EBNA algorithm with Traffic Adaptive capabilities. This algorithm captures station's activity, calculates the probability of collisions, and automatically chooses whether to use the classical or the EBNA 802.11 approach. In addition, when EBNA has to be applied, it detects the number of actual active STAs in the network and implements a dynamically adjustable CW in order to keep average backoff time in the lower possible levels.

The rest of this chapter is organized as follows. Initially, the idea and the details of the Hybrid-Exclusive Backoff Number Allocation algorithm (H-EBNA) are discussed. Then, the implementation of the algorithm in OPNET is described and a detailed validation is performed and presented in order to ensure its acquired operation. Finally a series of simulations under realistic condition are performed and the results are presented and analyzed.

5.2 The Hybrid-EBNA concept

The enhanced version of the EBNA algorithm proposed in this chapter takes a series of actions regarding delay improvement. The target is clearly defined and it is to achieve performances close to the performances achieved by current commercial wired audio networking systems and also specified in the recently released AES-67 standard for audio-over-IP interoperability (chapter 2.5.5). These performances are mainly the facilitation of 60 networked devices in an acceptable for real time audio throughput (less than 20% losses) and a near to 10 milliseconds average end-to-end delivery. The actions taken by the proposed H-EBNA algorithm can be classified as follows:

- Constantly captures the broadcasting activity in the network
- When a packet is to be transmitted, calculates the number of active STAs in the network
- If the probability of collision is low, uses classic 802.11 to broadcast the data
- If the probability of collision is high uses EBNA approach based on the active STAs number
- Applies the enhanced CTS-to-self protection when EBNA is used.

The proposed algorithm follows the philosophy of *Distributed Coordination*, without centralized management from an AP. That means that each STA in the network implements the algorithm independently and all collected information are its own property and all calculations and decisions are taken according to its own consideration. This gives a high flexibility in the formation and maintenance of the WLAN, which is an important issue in audio networks generally.

5.2.1 Creation and maintenance of the List of STAs

The first step in implementation of the H-EBNA is the creation of list of STAs operating in the network. Upon joining the WLAN all STAs obtain a STID, as it was stated in section 4.2. Every time a STA successfully broadcast a CTS-to-self this portends that a data packet is pending and will be transmitted immediately after. The CTS-to-self includes in its body the senders STID. This way, all STAs will register their ID to all other STAs in the WLAN after their first transmission attempt. There are several ways to nominate STIDs in STAs within an ad-hoc network but a further discussion of this subject is beyond the scope of this thesis.

Every station after starting operation, creates a static table called *General List of STAs*. Each line of this table contains two parameters, a STID and the time of the most recent activity triggered by the station with this ID. During network's operation, every time a STA receives a CTS-to-self message checks its ID and updates the time stamp in the *General List of STAs* table. By using this technique a clock distribution is also avoided. Each STA uses its own timer because, as we will analyze below, time differences between packets rather than absolute time values are used from H-EBNA algorithm. At any given time, the *General List of STAs* it actually illustrates the current network activity.

The decision to distribute STIDs by using CTS-to-self instead of the data packet is very important, regarding interoperability and coexistence of an H-EBNA audio network with other wireless networks within sharing infrastructures. However, this will be discussed extensively in chapter 6 where a related study is performed.

5.2.2 Defining the active STAs in the network

Audio networks are mainly addressed to live music applications. Live musical performance has a strong stochastic nature. Therefore, in a network of STAs that handle data broadcasting of music instruments or vocals, there is a strong possibility to have short time intervals where not all STAs are intending to send data. Figure 5.1 illustrates an example of traffic created by 4 STAs that broadcast music type data. It is shown that for random time instances ($t_A, t_B, t_C, t_D, t_E, t_F$) the number of active STAs varies.

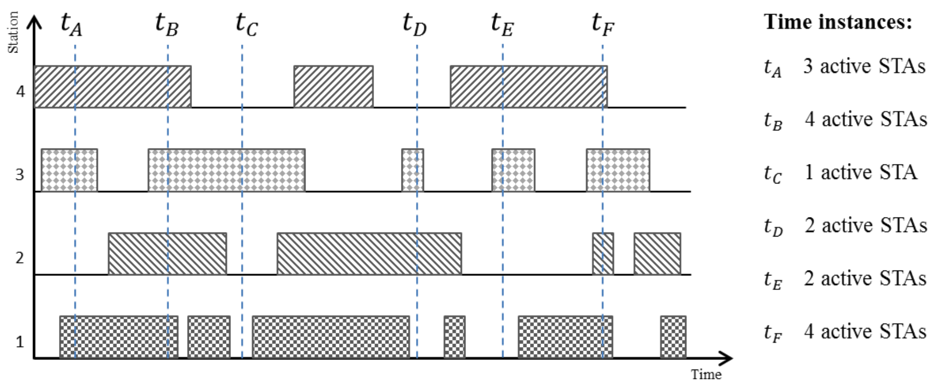

Fig 5.1: Variation of active stations in stochastic traffic generation

Basic EBNA guaranteed maximum throughput but it implies that all STAs are active at all times. This increases the mean backoff time and consequently raises overall delay. The proposed H-EBNA resolves this problem by monitoring the traffic in the network and by adapting the CW size according to the active STAs and not to the entire number of STAs in the wireless network. In order to do so, prior to each transmission a STA reads its *General List of STAs* and define through it a new list called *List of Active STAs.* This is created as follow. For each registered ID, the algorithm subtracts from the current time the time of the last CTS-to-self arrival and compares the outcome TD with a predefined time threshold $T_{threshold}$. The decision if a STA is active or inactive is taken by the following rule:

$$TD = [Current\ Time] - [Last\ "CTS - to - self"\ Arrival\ Time] \tag{5.1}$$

$$\left.\begin{array}{ll} TD \geq T_{threshold} & STA = Active \\ TD < T_{threshold} & STA = Inactive \end{array}\right\} \tag{5.2}$$

For 802.11g with a bit-rate of 54Mbps, we have a *bit transfer time* of $1.8\cdot10^{-8}$ and a *byte transfer time* of $1.481\cdot10^{-7}$ sec. For H-EBNA we use constant packet size. Each packet has a size of 2234 bytes, (2200 bytes of data plus 34 bytes overhead) [3-1], which finally gives us a *packet transfer time* $PTT=3.31\cdot10^{-4}$ sec. A CTS-to-self message consists of 14 bytes and as we analyzed in chapter 3, it is transmitted with the same bit-rate as data packets. Therefore, a *CTS-to-self transfer time* will be $CTSTT=2.0735\cdot10^{-6}$ sec. Due to the nature of wireless medium, only one STA transmits at a time. We denote *Round* time the minimum time needed for all STAs in the network to transmit a packet. For a network with 60 STAs, *Round* will be:

$$Round = 60 \times (PTT + CTSTT + DIFS + SIFS) \tag{5.3}$$

For DIFS=$50\cdot10^{-6}$ sec and SIFS=$10\cdot10^{-6}$ a round will be 0.01998 sec. The time threshold is adjusted in order to give to each STA a three rounds opportunity to transmit a packet, using the maximum expected population of 60 STAs. This gives a $Tthreshold_{-60}$=0.05995 sec. This three *rounds* scheme gives to all stations a minimum of 300% probability (100% in each *round*) to gain access to the medium and broadcast their packets. Stations that fail to broadcast a packet within this time interval are considered inactive and they are excluded from the active STAs list.

5.2.3 Benefits of using the active STAs list (an example)

The use of active STAs information to implement the EBNA algorithm is an important innovation. The reduction of time spent for unnecessary backoff slots contributes significantly in the reduction of the overall end-to-end delay during broadcasting. In order to make this clear, an example that compares H-EBNA with EBNA is presented below. Let as assume a WLAN with 53 STAs. The general list of STAs is shown in figure 5.2.a. We also assume that at a given time there are 4 active STAs in the network. The list of active STAs is presented in figure 5.2.b. Implementing EBNA in STA-2, based on the general list, we have to choose a backoff number between 2 and 105 (figure 5.3.a). On the contrary, implementing in STA-2 the H-EBNA, based on the list of active STAs we have to choose the backoff number between 2 and 7 (figure 5.3.b). For 802.11g the slot time can take values from 9 to 20 [75] microseconds. Therefore, in this particular case the maximum reduction we can get using H-EBNA can be 98 slots. This can reduce delivery delay up to 1.96 milliseconds per packet.

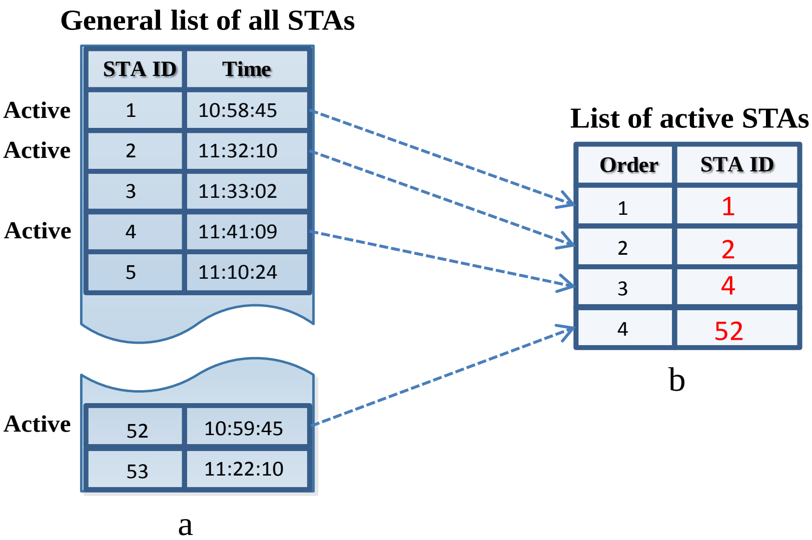

Fig 5.2: The general list of all STAs (a) and the list of active STAs according to H-EBNA (b)

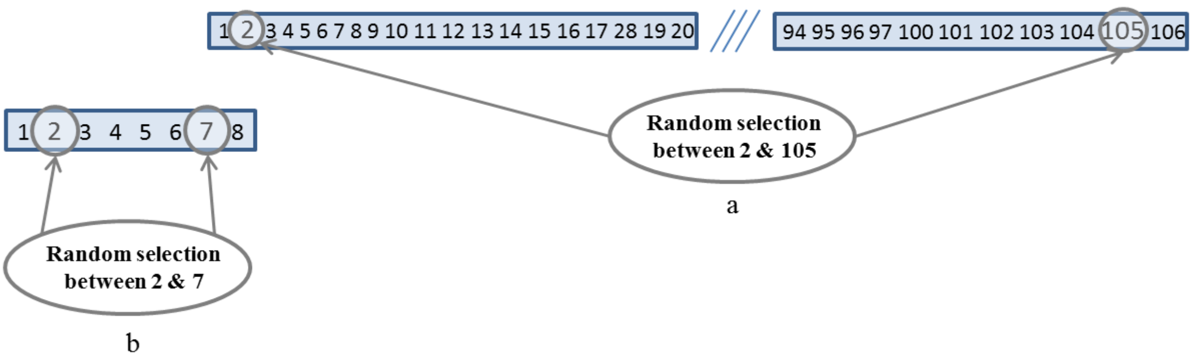

Fig 5.3: Backoff number range, H-EBNA (a), EBNA (b)

5.2.4 Switching between classic and EBNA 802.11 MAC

The algorithm prior to each transmission checks the traffic condition in the WLAN and decides whether to backoff using classic 802.11 or EBNA, based on the number of active STAs in the network. WSTAs are independent without any kind of central control. However, considering the fact that all of them monitor the wireless network using the same technique (H-EBNA), it is expected that they will all switch MAC algorithms coordinated. The algorithm is designed to give the chance to users to define the maximum loss rate depending on the demands in quality and also the type of digital audio in use (compressed or uncompressed). If P the percent of maximum acceptable loss, then the maximum acceptable probability of collision will be:

$$p = \frac{P}{100} \tag{5.4}$$

If N denotes the number of active STAs in the network, then the threshold of active stations N_T, needed in order for the algorithm to switch from classic 802.11 to EBNA, will be:

$$(From\ equation\ 3.6) \qquad p = 1 - \left(1 - \frac{1}{CW}\right)^{N_T - 1}$$

$$\therefore \quad 1 - p = \left(1 - \frac{1}{CW}\right)^{N_T - 1} \Rightarrow (N_T - 1) \times \log\left(1 - \frac{1}{CW}\right) = \log(1 - p)$$

$$\therefore \qquad N_T = \frac{\log(1-p)}{\log\left(1 - \frac{1}{CW}\right)} - 1 \tag{5.5}$$

Equation (5.5) allows us to set parameters p and CW when we are setting up the network but during network's operation remains constant. The number of active STAs however is a dynamic variable which changes in a stochastic manner. Switching between MACs will occur under the following rule:

$$\left.\begin{array}{ll} N > N_T & \text{EBNA MAC} \\ N \leq N_T & \text{Classic 802.11 MAC} \end{array}\right\} \tag{5.6}$$

When the number of active STAs increases the algorithm switches to the EBNA technique (using the list of active STAs), which eliminates collisions but adds delay. When the number of active STA drops below N_T the algorithm returns to the classic 802.11 backoff technique which keeps delay in lower levels. Thus, an average of high throughput is kept while the delay remains low.

5.2.5 The flowcharts of H-EBNA algorithm

In order to give a comprehensive description of the H-EBNA's operation, two flowcharts are presented below. The first flowchart (figure 5.4.a) describes the operations of the algorithm at the receiver's part where the *General List of STAs* is created and maintained. The second flowchart (figure 5.4.b) describes the operations of the algorithm at the transmitter's part where the decision for the optimum MAC process is taken and executed. Operation in (a) is a standard routine called every time a packet is received. Operation in (b) is part of the main MAC process executed every time a STA gains access to the wireless medium, completes successfully the CTS-to-self transmission and it is ready to broadcast a data packet.

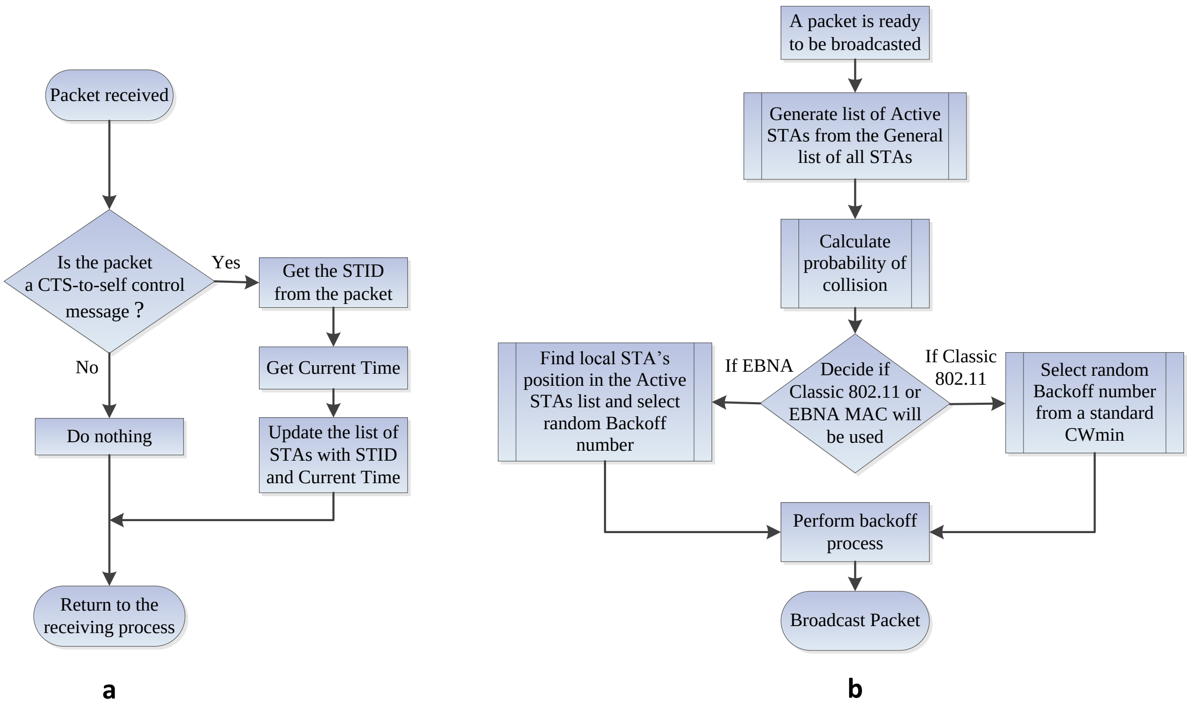

Fig 5.4: Operation of H-EBNA algorithm at the receiver (a) and the transmitter (b)

5.3 Implementation of H-EBNA in C++

The implementation of the H-EBNA algorithm is characterized by a complexity mainly because includes a considerable number of sub-operations that has to be executed simultaneously at the receiver and the transmitter. For this reason, during this research the algorithm was implemented in two stages. Initially the algorithm was implemented in a C++ environment in order to develop the code and test its fundamental operations. The idea at this first stage is to develop and test the algorithm's operation not as repeated process in a simulation environment but as a simple instant operation. This program describe the operations lying within the submission, from higher levels, of a packet and the final broadcasting of the packet and it contains all calculations, processes and decisions that has to be taken according to the H-EBNA logic. It is basically testing the operation of the main body of the algorithm and the code that implements it. Therefore, the process that creates and updates the *General List of STAs* in the network is not implemented in this stage but is provided manually during the start of program execution. This way, after code execution, we have the chance to compare the expected and received results and evaluate the proper operation of the algorithm. The complete C++ code describing the fundamental operation of H-EBNA algorithm is provided in Appendix B.

5.4 Implementation of the H-EBNA in OPNET

All OPNET models are open source and use C++ programing code to define their operation. Thus, the C++ code created at the first stage of the implementation of the H-EBNA algorithm was transferred, with some necessary modifications, in OPNET. However additional custom variables, attributes and statistics were set and numerous pieces of C++ code were injected in OPNET's 802.11 wireless model in order to modify its operation according to H-EBNA logic. This is a complex procedure since the implementation of the IEEE 802.11 MAC process in OPNET consists of many thousands of lines of code which are scattered in different states and functions within the Finite State Machine model that handle this process. Thereafter, a test scenario was created and a controllable test traffic model was applied in order to test and validate the behavior of the algorithm in the simulation environment. Finally, a complete study using the H-EBNA model, under realistic audio data traffic, was conducted.

A detailed presentation of all modifications made in 802.11 OPNET model, is provided in Appendix B. In this chapter a brief description of the variables and statistics used by the algorithm is given. This is necessary in order to understand its operation, test its behavior and analyses the simulation results.

5.4.1 Variables and statistics at the receiver

The main action of the algorithm at the receiving part of the MAC process is to identify the arrival of a CTS-to-self message and update the *General List of all STAs*. This list is created by using a one dimension array. This array has a fixed size defining this way the maximum number of STAs supported by an H-EBNA wireless audio network. To simplify the process, we give integer, ascending values as MAC addresses in all STAs in the network.

We also use the MAC address as a STID. This way we don't need a two dimension array to describe the *General List of STAs.* Each location in the array represents a STID and the content of this location keeps the arrival time of the latest CTS-to-self message arrived with this ID. To give an example; when a STA receives a CTS-to-self message with the source MAC address "3", updates the location number "3" of the array with the current simulation time.

To monitor the evolution of the *General list of STAs* array we also create a custom statistic called *sum_of_addr.* Every time the array is updated, we write in this statistic the content of a preselected location within the *General List of STAs* array. Hence, for STA with STID=3 for example, at any random simulation time this statistic will have a value very close or equal to the simulation time unless the preselected location is the 3^{rd} location. In such a case the statistic will contain the number "0" since a STA cannot receive its own transmission.

5.4.2 Variables and statistics at the transmitter

When a packet is to be transmitted the algorithm goes through the *General List of all STAs* in order to extract a second list which contains the active STAs in the network. This list is also created by using a one dimension array, similarly to the one at the receiver. However, this is a dynamic array with size varying according to the number of active STAs. In order to handle the number of active STAs in the network an *Active_STAs_number* variable and its homonymous statistic is also created. The core operation of the H-EBNA algorithm however is the decision between classic and EBNA 802.11 backoff method. To monitor this decision process, an *EBNA_Monitor* variable is created together with a homonymous statistic. This variable is designed to operate as a flag indicating whether the algorithm decides to use the EBNA or the classic 802.11 approach. It is *true* when the EBNA is used and *false* when classic 802.11 is used. When the EBNA approach is selected, the algorithm needs to identify its own order within the list of active STAs. Hence, an *Order* variable and a related statistic is also created in this stage. As we mention earlier, the algorithm runs independently inside each STA model. In order to identify the order of the STA, the algorithm compares the ID of the STA with all STID in the *List of Active STAs*, find its order, and update the *Order* statistic. We need to note here that the *List of Active STAs* is always created from a STA every time he has a packet to transmit. Therefore, it always considers itself active and by default includes itself in the *List of Active STAs.* Finally, a *General_purpose_statistic* was also created. This statistic is mainly used to record the backoff number before a packet transmission, regardless the method in use.

5.5 Test and validation of H-EBNA's implementation in OPNET

Before studying the performance of the H-EBNA under music audio data traffic and compares it with the basic EBNA and the classic 802.11 algorithms, we must first test and validate its proper operation. In order to do so, we create a test scenario with a four STAs ad-hoc network and a low traffic model. According to this model, STA-1 switches from active to inactive with a duty cycle of 0.25 sec, STA-2 switches from active to inactive with a duty cycle of 0.5 sec while STA-3 and STA-4 are always active (figure 5.5).

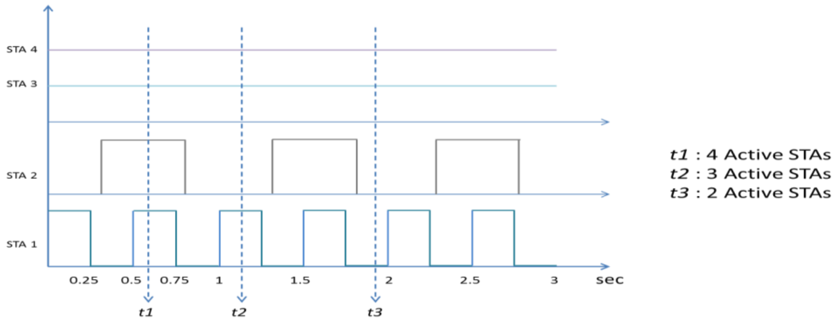

Fig 5.5: Test traffic model

The number of active STAs needed in order for the algorithm to switch to EBNA backoff method is greater than "2". The time threshold between two transmissions from the same STA, in order for him to be considered as active, is set to 0.0625 sec. The packet size is 2200 bytes and the interarrival time between packets 0.0243 sec. Thus the relation between active and inactive states varies during the simulation. The simulation runs for one minute which is enough time to enter a steady state. All the statistics described in section 5.4.1 and 5.4.2 where recorded and they are presented and analyzed below. Figure 5.6 gives us a collective picture of the main variables within the H-EBNA algorithm and how they vary during the simulation. We present here the statistics recorded in STA-4.

In 5.6.a we see the values taken by the *Active_STAs_number* variable. As long as there are 4 STAs in the WLAN it is expected for this variable to take values between 1 and 4. As we discussed above, this variable is updated every time STA-4 has a packet to broadcast, so by default consider it self active. This is way the minimum value for this variable is expected to be "1".

Figure 5.6.b shows the values recorded in the second location of the *Address_Sum* array which is hosting the *General List of all STAs* in STA-4. Therefore, this location always holds the last arrival time of the CTS-to-self messages send by STA-2. We can actually

observe from this graph the activity of STA-2 as it is perceived from STA-4. We can see that the last message was sent at the simulation time "1 min" (horizontal axis) and the time value recorded is 60 sec (vertical axis). This denotes the proper operation of the H-EBNA algorithm at the receiving part and also checks the proper operation of the proposed modification we discussed in chapter 3 and 4, regarding the CTS-to-self protection mechanism.

In figure 5.6.c we can see the variations of the *EBNA_Monitoring* variable which operates as flag and therefore takes values between "1" and "0". A careful observation shows clearly that EBNA flag is triggered whenever the *Active_STAs_number* variable take values greater than "2".

Figure 5.6.d shows the variations of the variable *Order*. This variable depicts the different positions that STA-4 takes within the *List of Active STAs* during the simulation. As long as we examine statistics collected by STA-4, it is expected that the maximum value of this variable will be "4". This will happen when the number of active STAs in the network will also be "4".

In order to examine the way the variables correlate between each other, and to validate the proper operation of the algorithm, a collective graph is presented in figure 5.7. This figure represents the simultaneous evolution of the variables: *Active_STAs_number*, *EBNA_Monitoring*, *Order* and *General_purpose_statistic*, for a time interval of 5 seconds within the simulation time for STA-3.

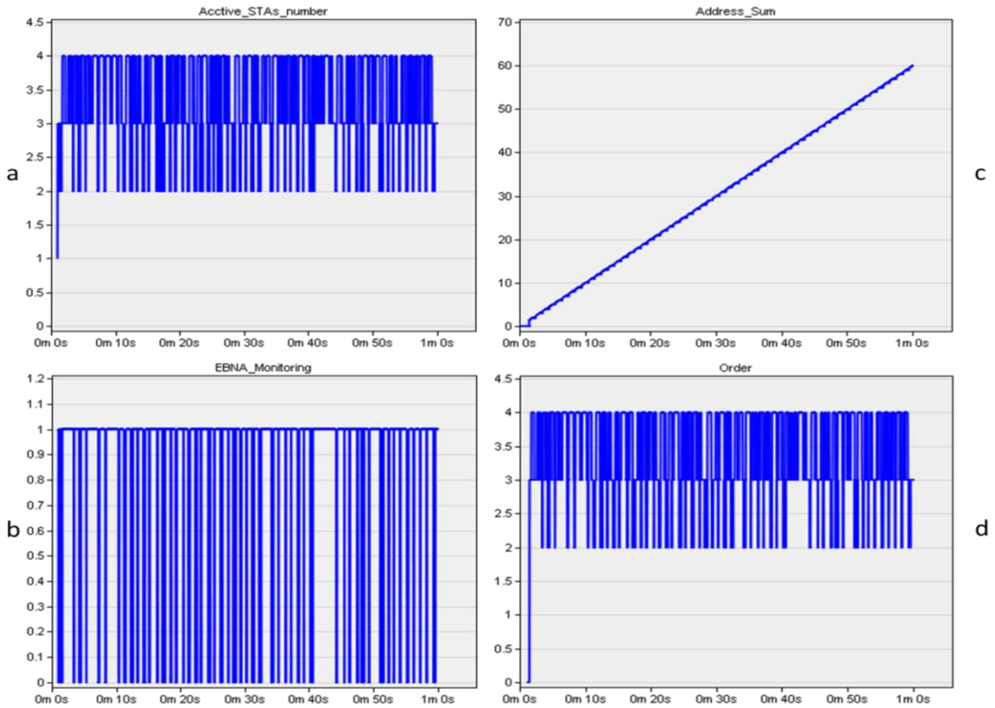

Fig 5.6: Operation test for H-EBNA's variables and statistics

We note that the *General_purpose_statistic* is set to collect the number of backoff slots assigned to STA-3 regardless of which method was eventually used. We particularly analyze the status of the algorithm for four different time instances t_1, t_2, t_3 and t_4.

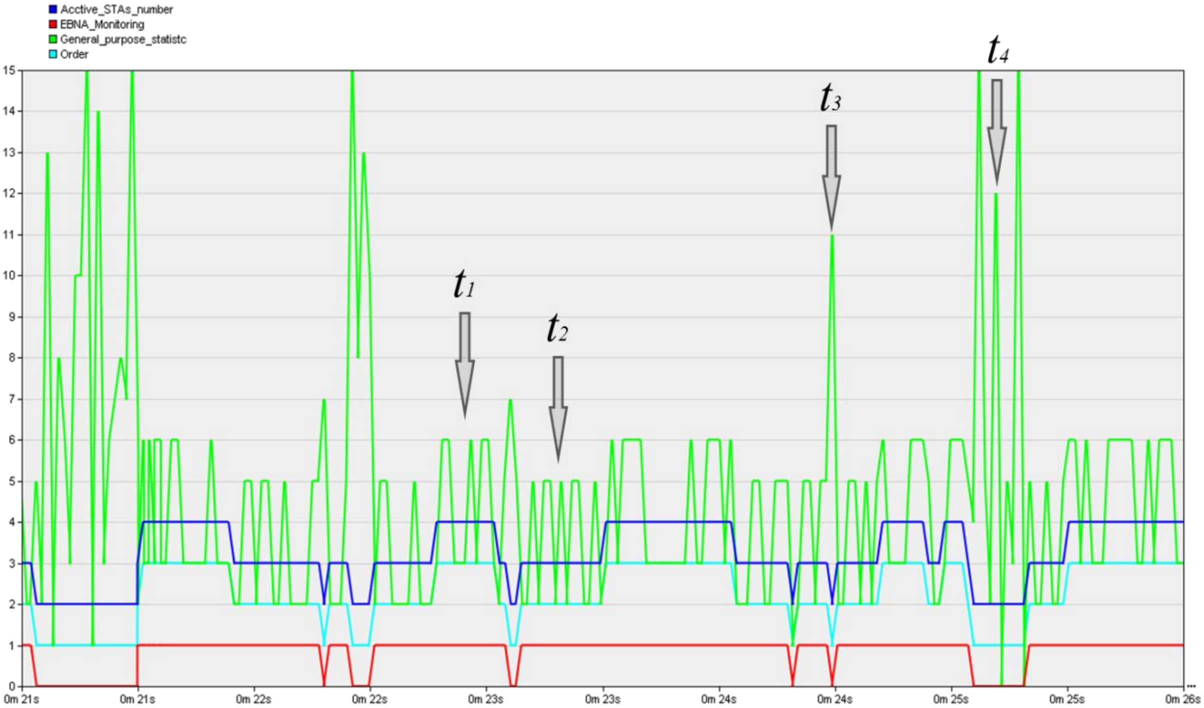

Fig 5.7: Collective representation of variables in STA-3

i. For time instant t_1, the number of active STAs is "4", (blue line), the order of STA-3 within the *List of Active STAs* is "3", (light blue line) and the EBNA backoff method is triggered, (red line). That means that the size of CW will be "8" (see equation 4.1 & figure 5.3.b). Therefore, according to the EBNA logic, the algorithm will choose randomly with equal probability, backoff numbers between number "3" and number "6". This is exactly what we observe here in the simulation (green line).
ii. For time instant t_2, the number of active STAs is "3", the EBNA method is active, the *Order* is "2" and therefore the CW size will be "6". In this case the backoff number values must be "2" or "5" and as we can see the algorithm operates correctly.
iii. At simulation time t_3, the EBNA is not triggered and hence the algorithm uses the classic 802.11 backoff scheme taking a random value from a $CW=\{0, 1, 2, 3\ldots 15\}$.
iv. Around time t_4, EBNA is also not triggered for a period of approximately 0.2 second. During this period several transmission attempts take place and the backoff values are randomly selected according to classic 802.11 method.

Another observation here is that, as long as the above statistics are collected from STA-3, the maximum value allowed for the variable *Order* is "3", just as it is shown in figure 5.7.

From the above it is clear that all the modifications in OPNET wireless LAN MAC process are properly set and the model is able to operate according to the H-EBNA algorithm rules.

5.6 Simulation characteristics

After completing the verification of the H-EBNA algorithm operation, a study under realistic conditions is performed. The simulation characteristics are similar to those described in 4.4.2. The IEEE 802.11g PHY with a bit rate of 54 Mbps is also used and the topology is again based on an ad-hoc WLAN with STAs located randomly in surface 30×40 meters. The number of STAs increases gradually up to 60 and for each change of population a separate scenario is created. We run the simulation for each scenario tree time with a different seed number and the average results are taken whenever a visible difference is observed. The simulation duration is 2 minute and the traffic is generated according to the “music audio data traffic model” proposed earlier in chapter 4. That gives a broadcasting data load of approximately 380 Kbps per STA. For each classification of the population (e.g. 40 STAs, 50 STAs etc.), both the classic 802.11 MAC process and the H-EBNA 802.11 MAC process are simulated in a separate scenario. In addition, for scenarios with 60 STAs which are the populations of our interest, the basic EBNA MAC process is also simulated.

5.7 Results (presentation and analysis)

The aim of the work presented in this chapter is to amend the idea of EBNA in order to maintain its throughput performance while achieving lower broadcasting delay. Thus, throughput and delay performance are the main parameters measured here.

5.7.1 Throughput performance using H-EBNA

Overall broadcasting throughput performance is measured in this simulation. That means that each successfully broadcasted packet is received by all STAs in the network which leads to throughput values that are greater than the nominal bit rate of the PHY in use. This is analyzed in detail in chapter (4.5.1). All STAs produces equal data load and thus the overall throughput is given by equation (4.7).

Figure 5.8 displays the average throughput measurement resulting from the simulation for network with 60 STAs. The figure contains results from all three MAC methods under examination. This is actually a stress test of the algorithm for both throughput and delay, because the support of such a number of STAs by a wireless audio network is a desirable target. It is shown that H-EBNA and simple EBNA are achieving practically similar throughput. In addition they both perform better than classic 802.11 which fail to broadcast successfully all created packets due to its disability to eliminate collisions.

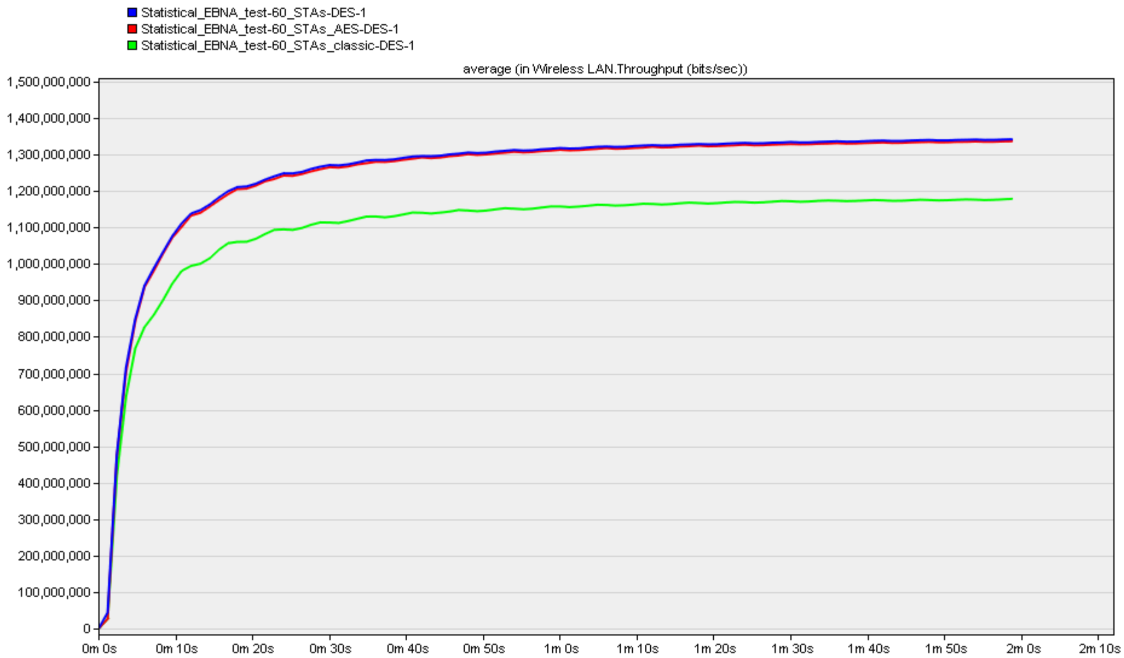

Fig 5.8: Throughput performance, WLAN of 60 STAs, H-EBNA (blue), EBNA (red), Classic 802.11 (green)

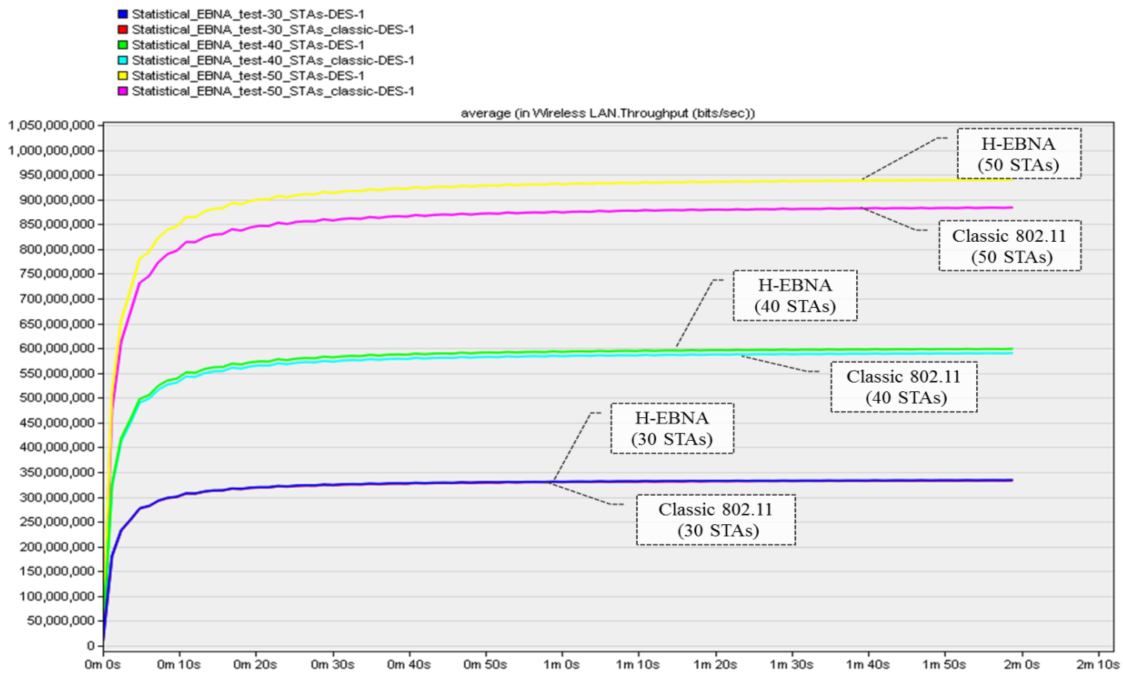

Fig 5.9: Throughput performance for H-EBNA and Classic 802.11 for 30, 40 & 50 STAs WLAN

Figure 5.9 displays the evolution of average throughput performance. It is shown that as the number of broadcasting STAs in the network increases the difference in throughput between H-EBNA and classic 802.11 also increases. In networks with 30 STAs and below we can see that overall throughput is identical for both methods. This is because classic 802.11 MAC can handle successfully multiple broadcasting when the overall data load is kept low.

Table 5.1 shows the calculated theoretical maximum throughput and also the experimental results for different network populations as derived from the simulation for both h-EBNA and classic 802.11 MAC.

Simulation Scenario	Number of STAs	Maximum Theoretical	Simulation-Classic 802.11 MAC (b/sec)	% of the Max Theoretical. Throughput	Simulation-H-EBNA Modified MAC (b/sec)	% of the Max Theoretical. Throughput
1	10	34,560,000	34,500,000	99.82638889	34,510,000	99.85532407
2	20	145,920,000	145,500,000	99.71217105	145,600,000	99.78070175
3	30	334,080,000	333,000,000	99.67672414	333,600,000	99.85632184
4	40	599,040,000	589,800,000	98.45753205	598,000,000	99.82638889
5	50	940,800,000	883,000,000	93.85629252	940,000,000	99.91496599
6	60	1,359,360,000	1,177,000,000	86.58486347	1,342,000,000	98.72292844

Table 5.1: Throughput results, (Max. theoretical vs classic 801.11 and H-EBNA modified MAC)

Plotting from table 5.1, the percent of the maximum theoretical throughput achieved by classic 802.11 and H-EBNA modified MAC (figure 5.10), we can see that H-EBNA manage to handle broadcasting successfully compering to classic 802.11 MAC. The overall broadcasting throughput is more than 99% and it is suitable for real time audio applications. It is also shown that throughput performance is similar for both H-EBNA and basic EBNA modified MAC methods (dashed line).

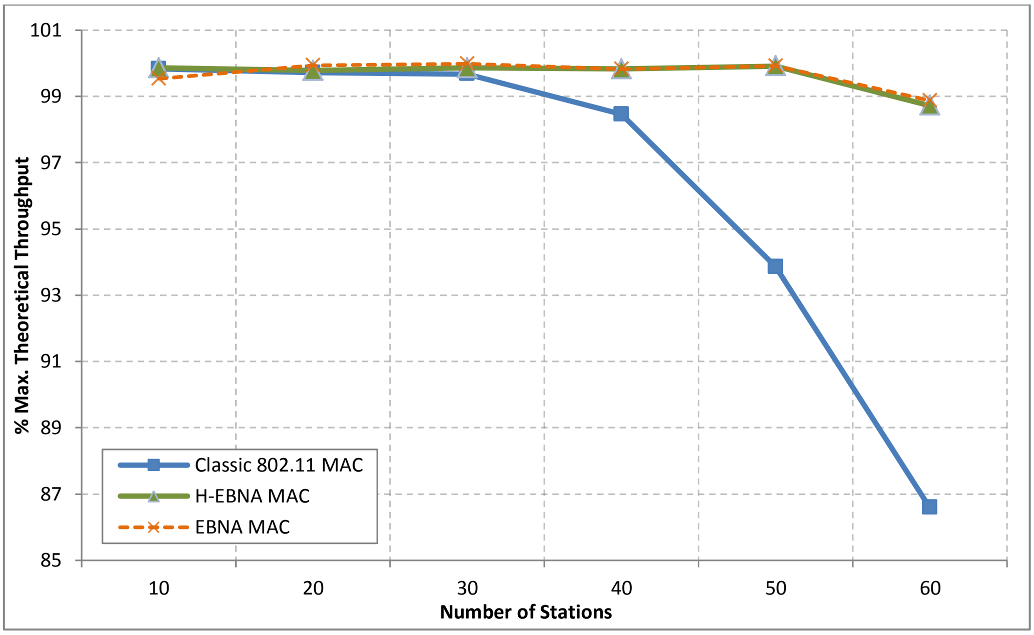

Fig. 5.10: Throughput performance, (Classic 802.11 vs H-EBNA and EBNA modified MAC)

5.7.2 Delay performance using H-EBNA

The delay measured in this simulation represents the overall end-to-end delay of all packets received by the WLAN MACs of all WSTAs in the network and forwarded to the higher layers. A detailed discussion regarding delay and its characteristics regarding broadcasting is given in section 4.5.2.

The simulation results show that a significant improvement has been achieved when it comes to delay by using the proposed H-EBNA algorithm.

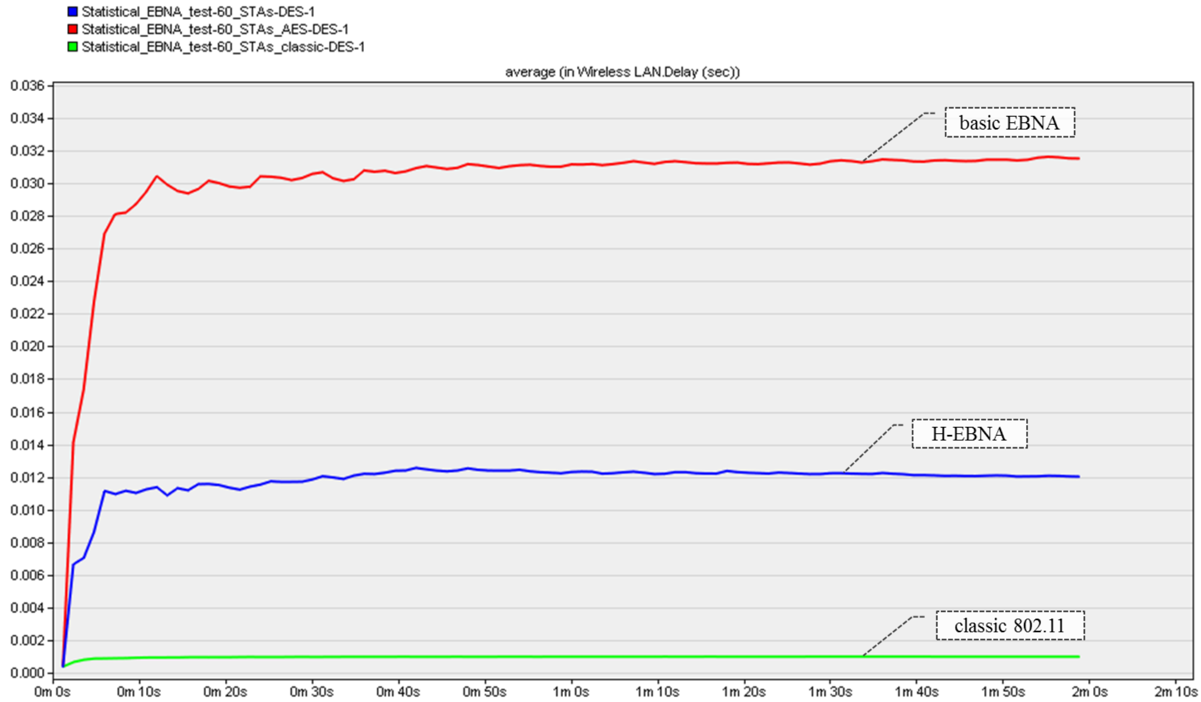

Fig. 5.11: Average end-to-end delay, WLAN of 60 STAs, Classic 802.11 (green), H-EBNA (blue), EBNA (red)

This algorithm takes advantage of the low delay behavior of classic 802.11 MAC algorithm in low traffic broadcasting. It monitors network's broadcasting traffic; calculates the probability of collision and switches between MAC methods. Thus, we achieve to improve delay comparing to the basic EBNA while maintaining high throughput. Figure 5.11 shows the average delay values for classic 802.11, basic EBNA and H-EBNA methods for a network with 60 STAs. It is shown in this stress test scenario that the delay does not exceed 12 milliseconds. This is a highly acceptable value regarding audio networks.

Simulation Scenario	Number of STAs	Classic 802.11 MAC (sec)	H-EBNA Modified MAC (sec)	EBNA Modified MAC (sec)
1	10	0.0003702	0.0003965	0.0003933
2	20	0.0004080	0.000463	0.000585
3	30	0.0004581	0.000748	0.000665
4	40	0.0005400	0.00123	0.00192
5	50	0.0007629	0.00440	0.00605
6	60	0.0009796	0.01203	0.0344

Table 5.2: Delay values for H-EBNA, EBNA & classic 802.11 MAC

In table 5.2 the aggregated results from the all backoff methods tested in this study, are presented. The plotting of the results in figure 5.12 shows that the H-EBNA algorithm outperforms the others and it also fulfils the requirement of delay regarding real-time audio networking applications as it is stated in chapter (2.9).

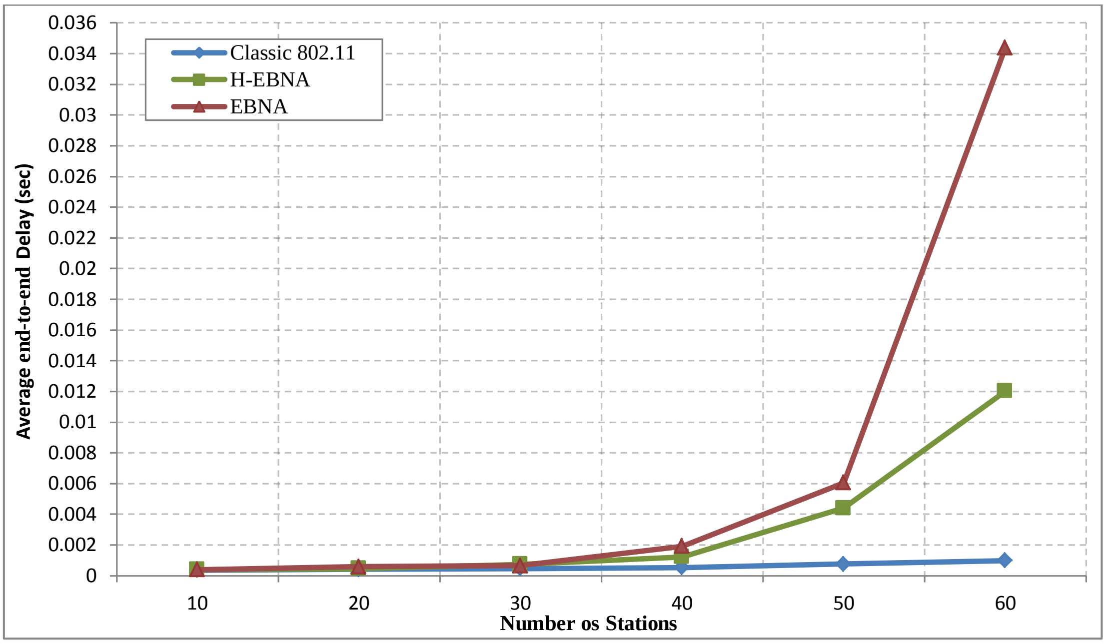

Fig. 5.12: Average Delay for H-EBNA, EBNA & classic 802.11 MAC

5.8 Summary

The proposed EBNA algorithm studied in chapter 4 shows that a significant improvement in throughput can be achieved in broadcasting of real-time audio data within a wireless ad-hoc network. However, when the number of STAs in the network increases, this method suffers from an increase of the overall broadcasting delay. The main aim of the research described in this chapter focuses in improving overall delay performance when the EBNA backoff method is used.

The causes that lead to the increase of the delay are two. First is the additional traffic caused by the use of the CTS-to-self protection mechanism proposed in chapter 3, and second is the linear increase of the CW in order to apply exclusive backoff numbers to each STA and thus, to eliminate collisions. Nevertheless, CTS-to-self protection mechanism is a key parameter in deploying the EBNA concept by distributing channel reservation information. Using this technique we protect STAs from collisions and also from unnecessary transmission attempt which leads to dropped packet. Therefore, the effort in this chapter focusses in deploying an efficient way to manage the size of CW.

We proposed and developed in this chapter an amendment of the EBNA algorithm, based on two important observations. The first observation is that both EBNA and classic 802.11 perform well regarding throughput, when the number of broadcasting STAs in the network remains small but with classic 802.11 to achieve lower transmission delay. The second observation is that regardless the number of the STAs in a WLAN, when the broadcasting data are produced from a music audio source, not all STAs appears to be active simultaneously. Due to the nature of the live musical performance there are short time intervals where STAs are not broadcasting.

The proposed amendment called Hybrid-EBNA algorithm and it is implemented independently in each STA adjusting with the distributed coordination philosophy of the IEEE 802.11 standard. The algorithm monitors the activity in the network and decides, based on several user defined parameters, whether to use classic 802.11 or EBNA approach as backoff method, for each individual packet that has to be transmitted. In addition, when the EBNA is chosen, it adjusts the CW size according to the number of truly active STAs rather to all STAs in the network. This dynamic CW helps STAs to maintain backoff waiting times to the minimum necessary level and thus to reduce overall broadcasting delay.

A special effort was made in order to properly define the decision parameters within the EBNA process. Initially, in order to define the switching parameter between classic 802.11 and EBNA, we took into account the widely accepted limits for packet loss that real-time audio can afford. The algorithm calculates the probability of collision and when it reaches these limits switches to the lossless EBNA method. We also defined the activeness parameter based on a proposed *three rounds broadcasting opportunity* scheme. This is actually the time threshold between successive transmissions in order for a STA to be considered as active or inactive.

The H-EBNA algorithm was initially developed in a C++ environment. Then it was implemented in OPNET and its operation was tested thoroughly. Finally, a scalable study was performed with the number of STAs gradually increased. Simulation results showed that H-EBNA maintains high throughput while reduces delay into acceptable levels for real-time audio delivery. Therefore, the IEEE 802.11 technology, appropriately modified with the H-EBNA algorithm, can be used as a networking platform for wireless audio networking applications.

Chapter 6

Interoperability:

(A study for the coexistence of the EBNA and H-EBNA algorithms with the conventional IEEE 802.11 MAC algorithm)

6.1 Introduction

One of the key initial objectives of this research was to propose solutions for the development of wireless audio networks which are widely acceptable and applicable. The design of a proprietary system in order to solve a specific problem is many times the easier way but the applicability of such a system is usually low and unprofitable. Our target was to design solutions that will be able to be implemented using the existing wireless networking technology and they will be able to share existing infrastructures. The proposed contributions, presented in chapters 3, 4 and 5, comply with the IEEE 802.11 concept and they can be implemented as optional modes of operation in the case of wireless audio networking applications.

Audio networks are implemented in several types of venues, which most of them are expected to feature wireless network infrastructures. Some characteristic examples are sport centres, malls, aerodromes and also music clubs, recording studios, and radio and television broadcasting studios. Building audio networks over existing wireless network infrastructure gives us significant operational and cost advantages. However, this raises a series of issues regarding the coexistence of the proposed systems with the regular 802.11 devices and the operation of this mixed network as a whole.

In this chapter we are investigating the ability of the proposed EBNA and H-EBNA system to operate in conjunction with conventional 802.11 devices within the same BSS and the effect of this coexistence. In order to do so, we create in simulation environment a network with two types of population; a constant number of STAs that exchange unicast information using the classic 802.11 protocol and a variable population where the modified MAC algorithms are applied. This variable population represents the wireless audio network, as all STAs are broadcasting audio data using our proposed “music audio data traffic” model.

The rest of this chapter is organized as follow. Initially the motivation of this study is thoroughly analysed and an overview of the proposed modifications is given in order to determine the objectives under examination. In addition, the proposed network setup and test procedures are described. Then the implementation in the simulation environment is presented and some issues related to the peculiarities of this application regarding OPNET, are also discussed. Finally the simulation results are presented and analysed.

6.2 Motivation of the study

Interoperability was always an important issue in networks. Audio networking especially, held back for many years due to this problem [76]. The AES X192 Task Group, during its 3 years of work had to consider many products and standards in order to define the optimum recommendations regarding interoperability in audio networking which are set out in the AES67 standard, released last year (2013) [77]. In this research we are investigating the chances of migrating this technology into the wireless domain. However, the aim of our work was to propose modifications that will improve the existing wireless networking technology towards audio networking, without major alterations and always maintaining its philosophy and its core operation. The novelties of this research lie mainly on the design of a protection mechanism for broadcasting in ad-hoc networks and also the design of two congestion control algorithms, the EBNA and the H-EBNA. EBNA and H-EBNA are both alternative methods for performing the random backoff process, a sub-operation of the MAC algorithm.

IEEE 802.11 standard is familiar in hosting alternative congestion control methods and different technologies. Data with different priorities can be assigned with different channel access parameters, TXOP can be used instead of the regular access control mechanism and legacy technologies can share the same channel, and even exchange data with modern technologies. All those techniques can interoperate due to the distributed coordination philosophy of the standard. The proposed modifications are aligned with this idea and they are designed to be able to work together with conventional 802.11 devices and to share infrastructures and resources without disturbing their operation.

More specific, distribution of NAV information in broadcasting is achieved by the use of CTS-to-self message. In its modified version this message is sent from a STA prior to each broadcasting packet using the higher operational bit-rate instead of a low bit-rate, as when it is regularly used as a protection from legacy technologies. However, this message can be received and utilized by all IEEE 802.11 STAs as it has identical structure with a typical CTS message used in the RTS/CTS protection technique.

Basic EBNA is also compatible with classic 802.11 systems. It protects broadcasting using the proposed CTS-to-self mechanism while offers an alternative backoff scheme for STAs that form a wireless audio LAN. This is achieved by linearly increasing the CW according to the population of the STAs in the audio network and assigning exclusive, equally weighted, pairs of backoff numbers to each STA. Increase of CW is a technique that is already in use within the 802.11 standard, in a different concept. Therefore, STAs using the proposed modifications are able to coexist with other STAs in the network, favouring however broadcasting STAs in the case of higher number of retransmission attempts.

Finally, H-EBNA operates in a similar to EBNA way but using a traffic monitoring technique that allows adjusting of the CW size according to the number of active STAs in the audio network. As it was mentioned in chapter 5, this traffic monitoring technique uses the CTS-to-self messages instead of data packets to record STAs activity. This way the H-EBNA algorithm is able to identify and separate traffic incoming only from the audio network STAs regardless the number of STAs in the BSS.

However, implementing the above described techniques in a mixed wireless network has consequences that have to be investigated. In this chapter we create a test setup where a wireless audio network, that uses the proposed EBNA and H-EBNA algorithms, coexists with a conventional WLAN. Then using simulation, we perform an empirical study in order to identify the ability of the two systems to operate undistracted between each other and to evaluate the effect of this coexistence in the overall performance of the network.

6.3 Setting up the test network

In order to identify the ability of cooperation between conventional and modified systems, a mixed network is be used as a test bed in this study. This network consists of a set of STAs forming an IBSS. The set of STAs is divided into two sub-sets with the first one operating as a regular ad-hoc network and the second to form a wireless audio network. The parameter which is expected to affect the network's operation is the application of the EBNA or the H-EBNA algorithm in the wireless audio sub-network STAs. For that reason, the number of STAs consist the regular sub-net is kept constant while the number of STAs in the audio sub-net increases gradually. However, these two sub-nets are not set to just shear the same coverage area and the same radio channels; they are actually arranged to be members of the same network. To make the test environment more realistic we also set the unicast traffic, created by the STAs in the regular sub-net, to be randomly directed to all STAs in the network. This way, STAs in the audio sub-net except transmitting and receiving broadcasting traffic from their peer members, they also occasionally have to handle unicast transmission from STAs in the regular sub-net, to manage the NAV information distributed by them, to response to their RTS control messages and generally to operate according to the rules the classic IEEE 802.11 mandates.

The number of STAs in the regular sub-net is M=56, using the IEEE 802.11g PHY and 54 Mbps bit rate. The intention behind this choice is to reserve not more than half of the protocol's capacity for the regular sub-net. Federico Cali et al in [78], investigate the IEEE 802.11 capacity in relation to the number of STAs in the network and the packet size. Packet size is expressed in this paper as an integer multiple of the slot length, t_{slot}. It is also assumed that all STAs are saturated, having always a packet for transmission. Figure 6.1 plots the IEEE 802.11 MAC capacity for three network configurations *(M=l0, 50 & 100)* and several packet lengths ranging from 2 slots to 100 slots. It is shown from this graph that with a number of 56 STAs in the network and using large packet size we reserve approximately 50% of the protocols capacity.

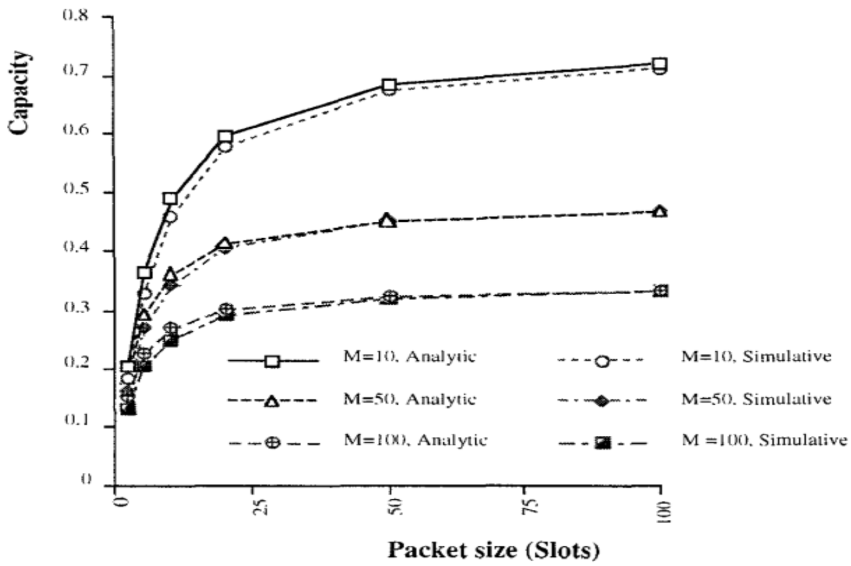

Fig. 6.1: Average IEEE 802.1 1 MAC protocol capacity (analytic and simulative estimates) [78]

The audio sub-net consists of STAs that have the same PHY with those in the regular sub-net (IEEE 802.11 & 54 Mbps). The number of STAs increases gradually ranging from 10 to 50, with a step of 5. For each variation of the network's population a separate simulation is performed. For each simulation however one type MAC method is studied.

As it was mentioned earlier in this chapter the aim of this study is to investigate the coexistence of the proposed backoff method with classic 802.11 and also to evaluate the mixed network's performance. For this reason a simulation using the classic 802.11 MAC for the audio sub-net is also performed. Another element that needs investigation is the effect of the use of CTS-to-self protection mechanisms. It is expected to play a significant role in the protection against collision in the entire network but it also adds significant traffic load. Thus, a study using an alternative EBNA algorithm without the use of CTS-to-self is also performed. It should be noted that an equivalent study using H-EBNA without CTS-to-self protection mechanism is not possible because the traffic monitoring algorithm, embedded in the H-EBNA, is based on this control message, as we already discussed in chapter 5. The interoperability test, performed in this chapter, is based on the following case studies:

- Broadcasting in audio sub-net using the classic IEEE 802.11 MAC
- Broadcasting in audio sub-net using the EBNA algorithm without the use of CTS-to-Self
- Broadcasting in audio sub-net using the EBNA algorithm with the use of CTS-to-Self
- Broadcasting in audio sub-net using the H-EBNA algorithm

6.4 Implementation in OPNET

The above described study is performed using OPNET modeler. The topology is based in an ad-hoc network spanning in a 60×60 meters surface. The network consists of a constant number of 56 unicast STAs located in the middle and a variable number of broadcasting STAs randomly surrounding the unicast group. This arrangement has been chosen for convenience reasons as each increase of STAs population requires a separate scenario in which, STAs are added manually in the network. However, several tests that were conducted with fully mixed STAs showed that there are no differences in the obtained results. Figure 6.2 shows the network's configuration in OPNET for 45 audio broadcasting and 56 unicasting STAs.

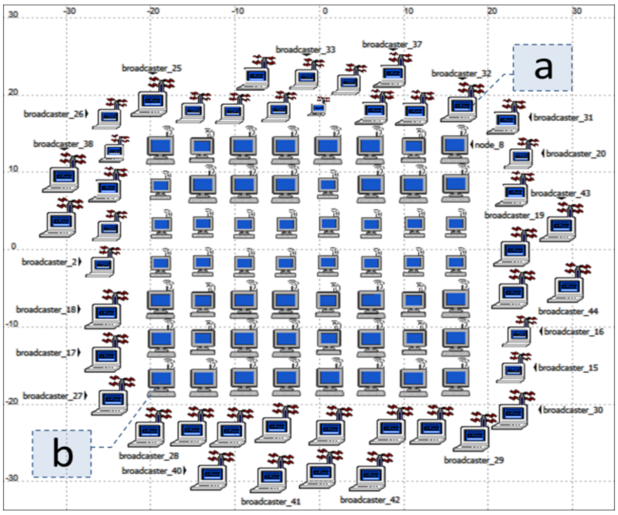

Fig. 6.2: Mixed network configuration in OPNET. Broadcasting STAs (a) and Unicasting STAs (b)

The two different types of STAs are set with different traffic generation characteristics. STAs in the audio sub-network are broadcasting data according to the "music audio data traffic model" described in 4.3, while STAs in the regular 802.11 sub-network are sending unicast data according to the parameters described in table 6.1.

Attributes	Values
Start Time	Normal Distribution (0.5, 0.1)
On-State	180 sec
Off-State	0 sec
Interarrival Time	Normal Distribution (0.1, 0.005)
Packet Size	2200 bytes

Table 6.1: Traffic generation characteristics for unicast STAs

The resulting load transmitted by each broadcasting STA is approximately 370Kbps while unicast STAs are transmitting with a bit rate of 77Kbps [79] [80]. The 56 unicast STAs are sending data to randomly chosen destinations including the existing broadcasting STAs.

6.4.1 The dual MAC operation in OPNET

OPNET modeler is a dynamic discrete event simulator based on hierarchical and object oriented modelling [81]. MAC algorithm is implemented in the *wlan_mac* process which is a child process of the *wlan_dispatch* process. The last one is the single process included in the *wireless_lan_mac* processor and it is invoked by the *wlan_station_adv* model every time a MAC algorithm has to be executed (fig 4.6). It wakes up only once, reads the attributes of the STAs and invokes the *wlan_mac* process which handles thereafter the whole channel access control procedure and also data transmission and reception. OPNET uses the same *wlan_dispatch* parent process for all its IEEE 802.11 models, including wireless station, wireless workstation and APs models. Although the *wlan_mac* processes can be modified and saved with a different name, only one *mac* process can be invoked during the simulation when any of the standard 802.11 OPNET models is used. However, in order to perform the interoperability study described in this chapter, we need two independent 802.11 *wlan_mac* processes running simultaneously. To achieve this, modifications were made in the *node domain* (chapter 2.16.2). More specific, a modified WLAN model created and also an alternative *wlan_dispatch* process. The modified WLAN model was assigned to the broadcasting STAs (fig 6.2.a) and through this the alternative *wlan_mac* process was invoked. This is an independent parent process and therefore is able to invoke any of the modified *wlan_mac* processes designed in the previous stage of this research and implement the EBNA and H-EBNA algorithms

6.4.2 Additional modifications

Some additional minor modifications had also to be made at this stage, in order for the proposed algorithms to be simulated in this mixed environment. Initially, an alternative EBNA process created to operate without the CTS-to-self protection mechanism. In the H-

EBNA process, the code that creates and maintains the *List of Active STAs* was modified in order to be able to separate the CTS-to-self protection messages from the simple CTS messages sent with the same STAID as response to RTS messages from STAs within the regular bub-net. In addition, the collision counter statistic added to all processed used in the audio sub-net STAs. Finally, a different icon was assigned to the STAs in audio sub-net for practical reasons (fig 6.2).

6.5 Results (presentation and analysis)

In order to investigate the interoperability between the proposed channel arbitration methods with the regular IEEE 802.11 MAC, several statistics are examined in this chapter. These are, throughput performance for the entire network, overall average end-to-end delay, the number of retransmission attempts, average backoff slots for both the unicast sub-net and the audio broadcasting sub-net independently and the number of collisions that experienced by the entire network. For each scenario the four case studies, described in section 6.3, are implemented. Therefore, each graph consist of four plots describing the behaviour of the network regarding the use of classic 802.11, EBNA without the CTS-to-self protection mechanism, regular EBNA and the H-EBNA.

6.5.1 Throughput performance

Throughput measurement describes the overall bit-rate of the successfully received data from all STAs in the network, which are forwarded to the higher layers. However, in this case we have two different types of data transmitted over the network; the broadcasted data that are received and forwarded to the higher layers from all STAs in the audio sub-net and the unicasted data that are received by only one STA. The throughput statistic's collection mode in OPNET is set to "bucket" (sum/time). That means that every t seconds, each bucket collects all values generated during t seconds of simulation time. Then, OPNET generates and reports a new value adding all values for this interval of simulation time. In our case, due to the dual mode of transmission described above and considering that all stations generate equal data load, throughput values in bps for each bucket will be given by equation (6.1). Where n is the number of broadcasting STAs, B the data load produced by the broadcasting STAs and U the data load produces by the unicasting STAs in the network.

$$Throughput_t = \left[(n-1) \times \sum_{i=0}^{t} B_i\right] + \sum_{i=0}^{t} U_i \tag{6.1}$$

Figure 6.3 shows the throughput performance for the entire network for all case studies.

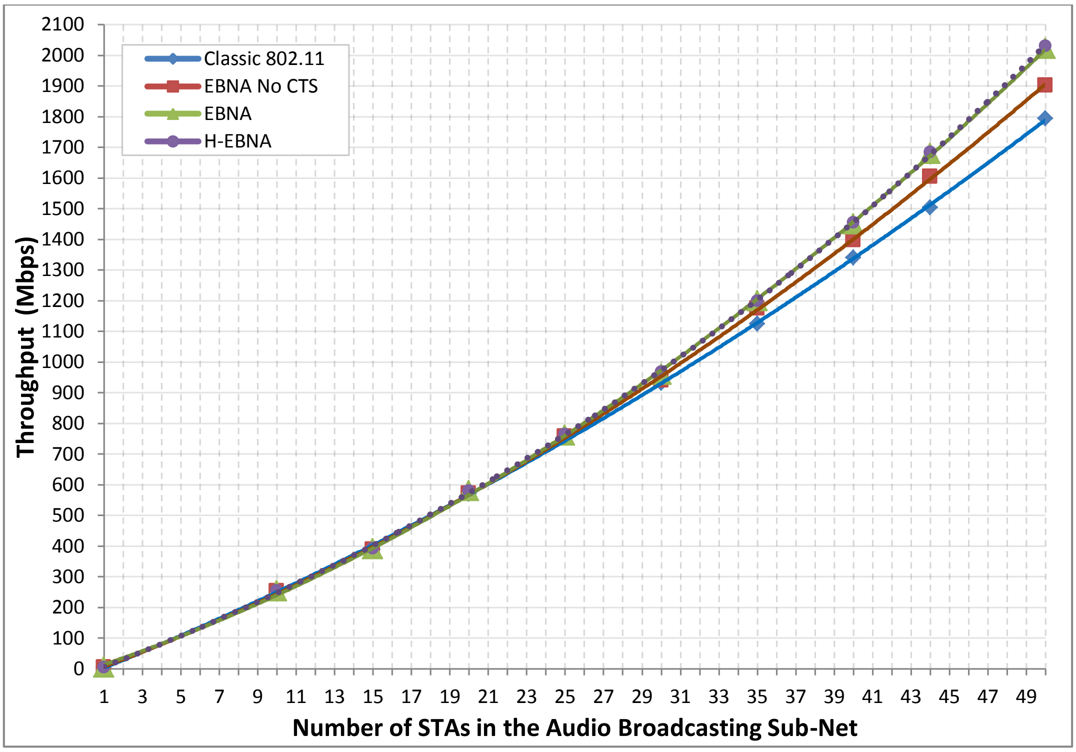

Fig. 6.3: Overall throughput performance

It is shown that when the number of broadcasting STAs is small all methods are performing equally. When the number of Broadcasting STAs increases all modified medium access processes are performed slightly better than the classic 802.11 MAC. We also see that EBNA and H-EBNA algorithms give the best results. This is because it guarantees that there are no collisions between broadcasting STAs which are handling the biggest part of the transmitted information in the entire network. Assuming that there are no hidden nodes in the network due to its compact size, and as long as the RTS/CTS protection mechanism is active, we expect that there is significantly low number of collisions occurring between unicasting STAs. Therefore, the majority of collisions at this stage are happening between broadcasting and unicasting traffic.

The fact that EBNA and H-EBNA are giving an equal throughput it is expected and it complies with the designing principals of the algorithms. Both are designed to eliminate collisions however H-EBNA is expected to perform better when it comes to delay.

It is also shown that the EBNA idea it is not complete without the use of NAV distribution. The improved results comparing to classic 802.11 are due to the linear increase of CW, correspondingly to the increase of the audio broadcasting STAs. It proves that this technique can be an alternative access control method, especially when the flooding of the wireless network with additional CTS-to-self control traffic has to be avoided.

6.5.2 End-to-end delay

This statistic shows the overall average end-to-end delay for all packets transmitted (broadcast and unicast), in the network. Therefore it gives us a general idea regarding the network's operation when various access control mechanisms are implemented in broadcasting sub-net. Also shows the effect size of the implementation of a wireless audio network in the entire WLAN. Figure 6.4 shows the measured average delay for all four case studies.

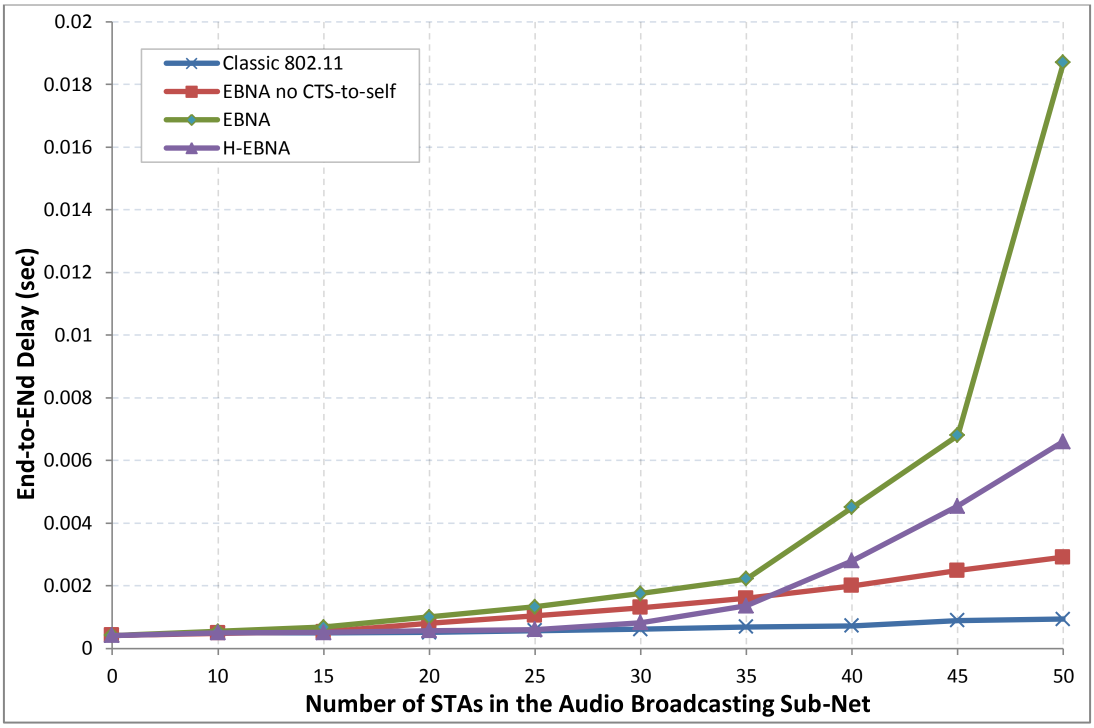

Fig. 6.4: Overall end-to-end delay

As it is expected, the overall delay increases in all modified MAC processes because in all of them a wider CW it is used. It remains though in acceptable levels. Classic 802.11 causes the lowest delay which however followed by a smaller number of successfully delivered packets, due to the large number of collisions.

EBNA without the use of CTS-to-self protection mechanism it also gives low delay. The transmission of a CTS-to-self control message prior to each broadcasting packet increases overall delay. The lack CTS-to-self messages reduces delay but it also reduces throughput, as we can see in figure 6.3.

By implementing basic EBNA, the delay increases significantly when the number of broadcasting STAs increases. In the case of 50 broadcasting STAs, which is a realistic scenario when it comes to audio networking; it reaches values that are marginal for real time media delivery. We note that this delay applies to all transmissions in the entire network. This significant increase in delay is caused by the increase of CW which in EBNA depends on the number of STAs in the audio sub-net regardless they are active or inactive.

By contrast, H-EBNA with its data traffic adaptability, it adjusts the CW size according to the number of active STAs in the network and thus manages to maintain delay in very satisfactory levels, regarding audio/media broadcasting. The operation of H-EBNA is shown clearly within the range of 1 to 25 broadcasting STAs. In this range the H-EBNA switches to classic 802.11 medium access method, as the probability of collision is low. Therefore, in low broadcasting populations, classic 802.11 and H-EBNA are expected to give similar delay values as it is shown in figure 6.4.

6.5.3 Backoff slots

The average number of backoff slots is measured in this statistic. As it is mentioned earlier, there are two different types of MAC algorithms running simultaneously in all simulations. Therefore, two separate statistics for the number of backoff slots are collected in this study. Figure 6.5 shows the average backoff slots measured in each unicasting STA and figure 6.6 shows the average backoff slots measured in each broadcasting STA.

It is shown from figure 6.5 that unicasting STAs have an average number of backoff slots close to 7.5. This means that they execute the random backoff process using a CW=15. This shows however, that most of the packets are managed to be transmitted during the first attempt. The use of EBNA introduces an increasing number of collisions in the WLAN which are happening mostly between CTS-to-self messages and thus are not affecting throughput. However, those collisions are affecting unicast transmission where STAs are forced to exponentially increase their CW. Figure 6.5 shows that this is happening to a limited degree and mostly when the number of broadcasting STAs excites 45. It also shows that the exponential increase of CW is taking place only once, from CW=15 to CW=31. The implementation of H-EBNA has a similar but more moderate effect comparing to EBNA, in the total number of backoff slots encountered in the STAs within the unicast sub-net.

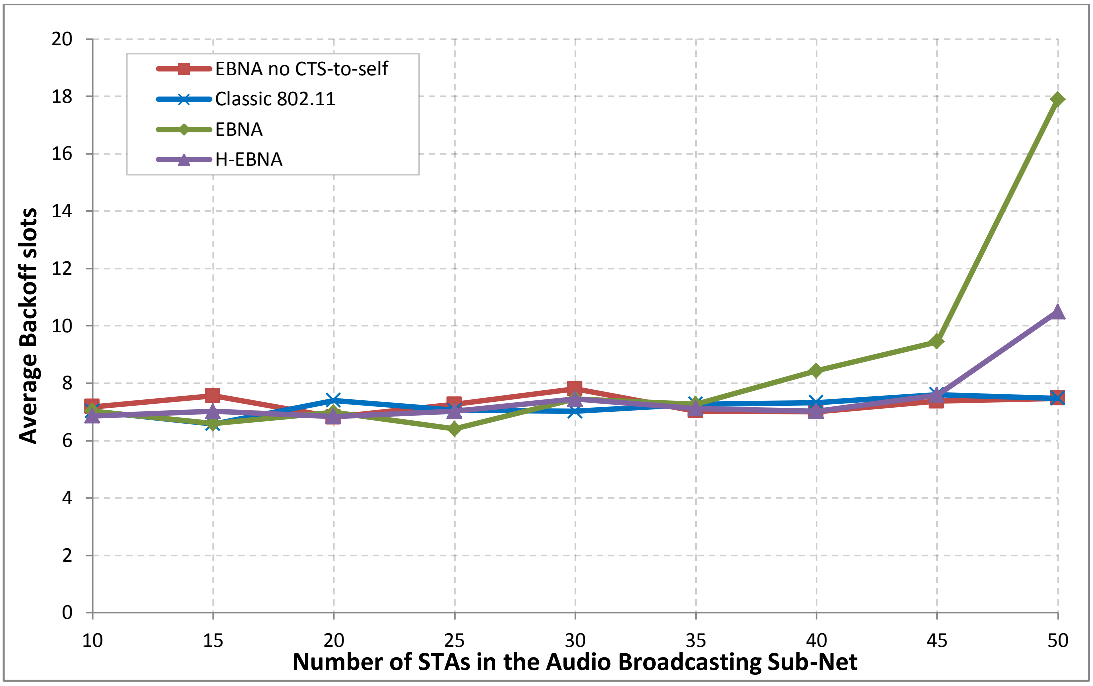

Fig. 6.5: Average backoff slots in unicasting STAs

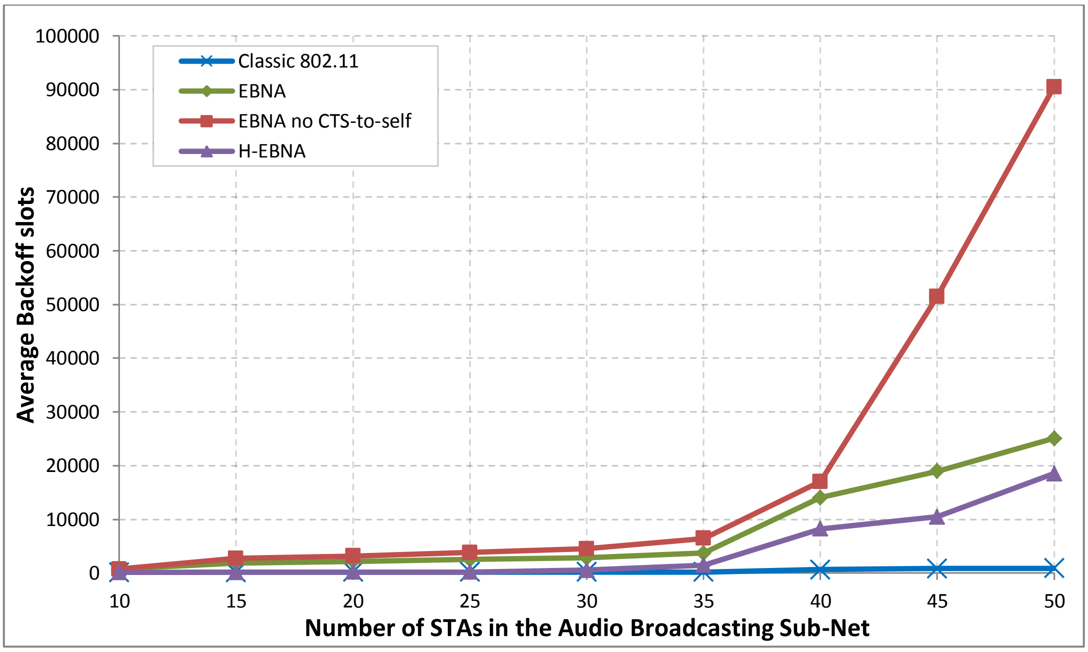

Fig. 6.6: Average backoff slots in broadcasting STAs

In STAs within the audio sub-net, the number of backoff slots affected only by the increase of CW size, in the cases where EBNA with or without CTS-to-self protection, and H-EBNA are applied. As it is expected, the implementation of EBNA without CTS-to-self and thus without NAV distribution, causes unnecessary backoff count downs as STAs are not able to implement virtual currier sense. The implementation of both the basic EBNA and H-EBNA causes reasonable waiting times. However, H-EBNA causes a smaller number of backoff slots comparing to basic EBNA. In addition when the number of broadcasting STAs is small, H-EBNA coincides with classic 802.11 as it is expected to do according to the designing principals of the algorithm.

6.5.4 Retransmission attempts

This statistic measures the average retransmission attempts attributable to each transmitted packet for the entire network. It is directly affected by the number of collisions, and as we can understand concerns only the unicast transmission. According to the IEEE 802.11 standard, broadcasting STAs have no chance for retransmission. Retransmission attempts measurement is important in order for us to understand the influence of the implementation of an audio network, in the operation of a regular wireless network in the case they share the same infrastructure. As we can see from figure 6.7, when the size of the audio network is small the operation of the regular data network is slightly affected in all case studies. When the number of STAs in the audio sub-net increases, EBNA causes the higher number of collisions, mostly between CTS-to-self messages, and thus it causes increases in the retransmission attempts of unicasting STAs. By contrast, EBNA without CTS-to-self protection allows more packets to be transmitted from the regular sub-net with the first attempt but many of them collide causing an increase of throughput (fig 6.3). Once again we can see here the superiority of the H-EBNA method which manages to maintain balance between throughput, delay and retransmission attempts in a mixed (broadcast/unicast) environment.

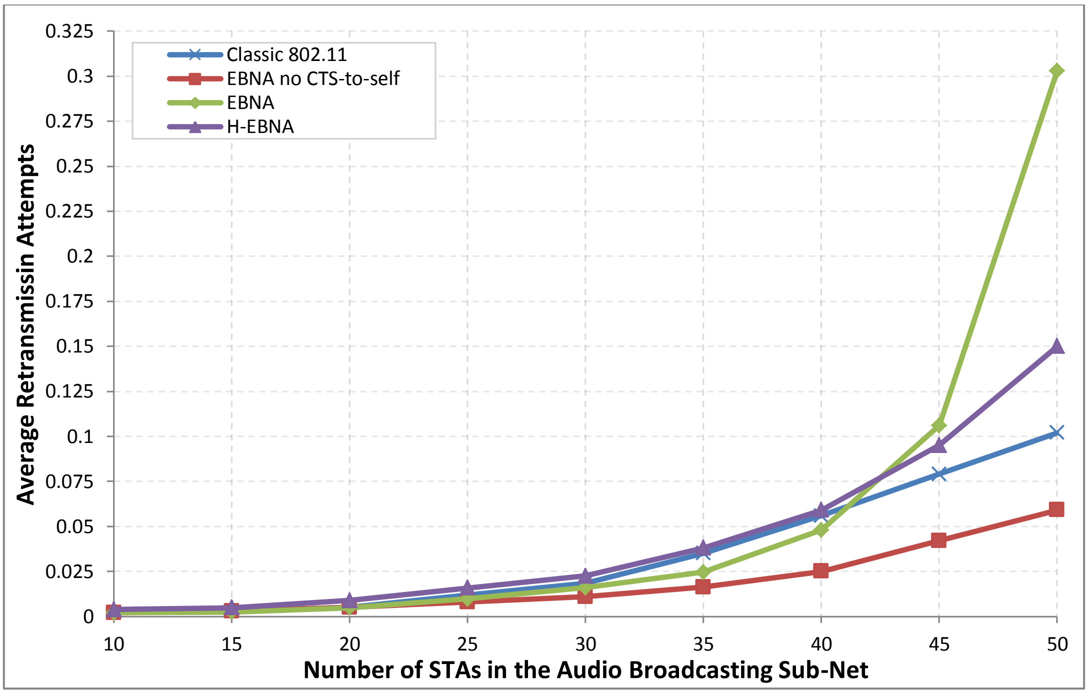

Fig. 6.7: Retransmission attempts in the regular sub-net

6.5.5 Collisions

This statistic describes the total number of collisions encountered in the entire network during each simulation. This is not a standard OPNET statistic. In order to obtain this measurement the OPNET *wlan_mac* process is equipped with a counter which increases every time the *collision flag* in OPNET is set. The accuracy of this custom statistic was validated using the OPNET *collision status* statistic which indicates the present of collisions along the simulation time. Figure 6.8 shows the total number of collisions for each case study for a simulation time of 1 minute.

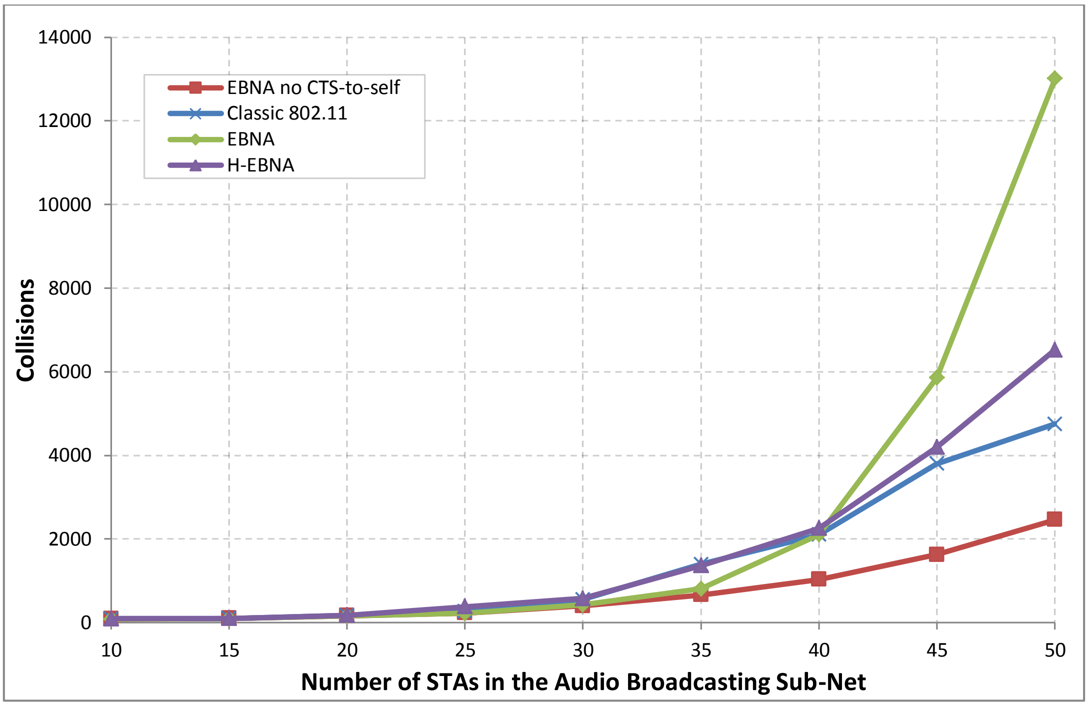

Fig. 6.8: The total number of collisions for 1 min simulation time

EBNA without the use of CTS-to-self protection mechanism cause the lower number of collision. This is because it avoids collision between broadcasting STAs from the audio sub-net by allocating unique equally waited couples of backoff numbers to each of them. However the lack of NAV distribution within the entire network cause collisions mostly between broadcast and unicast packets and thus reduces throughput.

The use of classic 802.11 MAC does not provide any collision avoidance mechanism. This cause a higher number of collisions between both types of transmissions, as we thoroughly discussed in chapter 2, and thus further reduces throughput.

Basic EBNA causes a significant number of collisions due to the additional traffic added by the CTS-to-self messages. However, these collisions are not affecting throughput and they are not contributing significantly to the increase of delay as they have small duration. The majority of the delay we experience when EBNA is used, created in the random backoff process of the broadcasting STAs within the audio sub-net. That explains the phenomenon we observe in this study where the number of collisions increases but also the throughput increases.

Finally, when the H-EBNA is used, similar throughput is achieved (fig 6.3) with less than half of the collisions comparing to the basic EBNA. This happens because H-EBNA regulates the collisions occurrence by adjusting its operation with the traffic in the network switching when is needed between classic 802.11 and EBNA MAC methods.

6.6 Summary

The purpose of the study described in this chapter is to investigate the interoperability between the medium access methods proposed in this research with the regular IEEE 802.11 MAC mechanism. The aim of this research was to propose solutions for the implementation of wireless audio networks that are compatible with the existing technology and able to coexist with regular data wireless networks under the same infrastructure. In chapters 3, 4 and 5 we analyse the proposed methods and test their ability to contribute in the implementation of wireless audio networks. In this chapter we are investigating the ability of these methods to operate in conjunction with conventional IEEE 802.11 devices within the same network and the effect of this coexistence.

In order to perform this study we create in a simulation environment a mixed wireless network consisting of two sub-networks. The first of the two sub-networks contains a variable number of STAs which are broadcasting data. This sub network represents the audio network and it where we implement the proposed medium access methods. The second sub-network consists of a constant number of STAs sending unicast data to all STAs in the network, including those in the audio sub-network. This way we not only force the two sub-networks to coexist but to cooperate as well. We also set four case studies in each of which a different medium access mechanism is tested under variable populations regarding the audio sub-network. In order to have a complete picture of the network's operation, various statistics are collected during the study. Those are, overall throughput, end-to-end delay, retransmission attempts in the unicasting STAs, average backoff slots measured independently in the two subnets and total number of collisions encountered in the entire network.

A detailed analysis of the results shows that all proposed medium access methods are able to interoperate with the regular IEEE 802.11 devises within a WLAN. It is also resulted that the integration of a wireless audio network into a WLAN, using the proposed modifications, it is possible and it does not dramatically affect the operation of the entire network. Therefore, existing wireless network infrastructure with sufficient bandwidth can be used for wireless audio networking applications.

More specific we can see that both proposed methods (EBNA and H-EBNA) equally increase overall throughput performance when they are used. However, the use of EBNA causes an increase of the overall end-to-end delay which reaches values that are marginal for real time audio delivery. By contrast, the use of the H-EBNA ensures a balanced operation of the network providing high throughput but also keeps the number of backoff slots, the retransmission attempts and the number of collisions within acceptable levels and thus achieve a low average delay, suitable for real time audio networking applications.

Chapter 7

Conclusions and further work

7.1 Introduction

The study presented in this thesis was set out to explore the possibilities of developing audio networks for professional real-time applications using the existing, widespread technology of wireless data networks.

The advent of digital audio over the past decades created the need for reliable ways of transporting and distributing it within sound system infrastructures. This met the perpetual need for simplification of sound installations, which admittedly are becoming complex and dysfunctional when their scale increases. The idea of using local area network to distribute digital sound in live music and studio installations emerged almost ten years ago but networking technology were not ready to curry out this task. This led to the creation of a multitude of proprietary systems that uses alternative networking approaches and therefore they are unable to interoperate between each other. Both the Audio networking industry and the audio engineering standardization bodies, realized the impasse and a significant effort has been made the last three years to develop of a commonly accepted standard that gives the directions for the development of audio networks in the future. In the discussion however accompanying this effort there were no reports regarding the use of wireless networking technology for the development of audio networks. Wireless networks were always seen as unable to meet the requirements of audio networks and thus were excluded from the discussion. This research conducted in parallel to this effort and was motivated by this specific challenge.

In this thesis the drawbacks and limitations of the current wireless networks to support the development of audio networks, are investigated. Then a series of modifications and amendments are proposed, in order to overcome these limitations and thus to allow existing wireless data networking technologies to be used for the development of wireless audio networks in the future.

7.2 Novelties and contribution to knowledge

At the first stage of this research, several wireless networking technologies were studied and tested in order to investigate their suitability for audio networking development. The criteria by which this evaluation was carried out were not only technical. The cost of these technologies, the range of their spread and their ability to assimilate modifications and new trends was also considered. The wireless networking technology that is superior in all these areas is admittedly the IEEE 802.11 standard. This is a technology that is constantly evolving and improving keeping its philosophy and characteristics while maintaining backward compatibility. It has the ability to operate independently and within network infrastructures and it is integrated in the majority of mobile computing devices. IEEE 802.11 network exist in all premises of interest and its cost is considered significantly low.

Studying and testing all the possible ways of distributing audio data within an IEEE 802.11 network, broadcasting appears to be the only way that covers the real-time delivery demands, required by such applications. However, broadcasting in IEEE 802.11 has a series of inherited malfunctions. Apart that it cannot provide any kind of delivery guaranteed it has not a way to distribute channel reservation information. In addition, since the CSMA/CA algorithm is embedded in the MAC process, it cannot properly operate, as there is not a mechanism to adjust its operation to the current traffic in the network as it is happening in unicast transmission. Thus, the implementation of a wireless audio network, where many STAs broadcast audio data simultaneously, results to a large number of collisions and therefore reduces performance and makes such a network infeasible. This is therefore the main reason that IEEE 802.11 standard appears to be unsuitable for real-time audio networks, despite the large available bandwidth. The rest of this research is focusing into the designing and testing of solutions that alleviate these problems but at the same time to be able to operate within the IEEE 802.11 framework and interoperate with regular IEEE 802.11 devices.

In order to emulate the conditions under which a wireless audio network operates, a widely acceptable audio data generation model had to be defined. Data production from an individual source based on musical performance, is a stochastic process which however has some significant properties. It is mainly a single data stream which has a stochastic beginning and end but it is not continuous. Instead, it follows repeated patterns which are directly related to the musical tempo. Published research shows however that the distribution of tempi within the global musical anthology is not uniform but is based on a normal distribution curve with the majority of tempi to be identified around the tempo of 120 bpm. Based on this evidence a traffic generation model that emulates data production derived from the music performance was designed. This model has a significant value beyond this particular research. By maintaining the timing framework and modifying only the density of the produced data, the model can support modelling of a variety of compressed and uncompressed audio data sources. Therefore, it can be used as a standard data traffic model for audio networking, allowing the comparative study between different research efforts.

One of the main issues regarding broadcasting in IEEE 802.11 is undoubtedly the lack of any kind of protection mechanism against collisions. The designers of the protocol have provided the possibility of broadcasting in the IEEE 802.11 standard, mostly for control and arbitration purposes. Thus, the reliability issue was not raised. However, in media broadcasting, where recovery of lost data is neither possible nor desirable, the protection against collision is a major factor. For that reason, a collision protection mechanism for broadcasting was designed in this research. This mechanism was based on the basic idea of channel reservation used by RTS/CTS protection mechanism in unicast transmission. With the exchange of those messages, vital information regarding the duration of the forthcoming transmission is distributed in the entire network. This forces all other STAs to defer transmission and thus minimize the probability of collision. We apply this technique in broadcasting by performing a "blind" CTS transmission using a modified CTS-to-self message, originally used for protection against interference from legacy technologies, in mixed networks. This is possible thanks to a loophole in the standard which, due to its distributed coordination nature, is not able to identify whether a CTS message is transmitted independently or as reply to a RTS. The advantage of this technique is that it does not require any kind of modification on the receiver and thus makes its implementation simple and interoperable. Tests of this technique using simulation showed that it significantly improve throughput when many saturated STAs are broadcasting in an IEEE 802.11 network. However this improvement can be achieved mainly when large size packets are used. For throughput-sensitive applications a combination of the proposed protection technique with the appropriate packet size can guaranteed reliable broadcasting, adding a small delay, acceptable however from time-sensitive applications.

The next step of the research focuses in the core problem that causes the collisions in broadcasting using the IEEE 802.11 standard, which is identified in the MAC algorithm. In IEEE 802.11 the access to the medium is achieved based on a random backoff scheme. STAs intended to transmit are forced to wait for a time frame by randomly selecting an integer from a minimum CW. Successful delivery is determined by positive acknowledgment the lack of which implies a collision and consequently an increased traffic in the network. In such a case an exponential increase of the CW is performed by the standard in order to decrease the probability of collision. In broadcasting however, the lack of an acknowledgment mechanism holds the CW in its smaller size and when the number of broadcasting STAs increases the number of collisions also increases dramatically. An algorithm that solved the collision problem is proposed in this research. The algorithm takes advantage of the finite number of stations that form an audio network and assigns an exclusive couple of backoff numbers to each individual STA. All couples of backoff numbers however are equally weighted. According to this exclusive backoff number allocation algorithm (EBNA), STAs randomly select one of the two numbers from their exclusive couple every time they have a packet to transmit. Thus, In the long run, all STAs are assigned equal waiting time during the backoff process. This algorithm has the significant advantage to totally eliminate collisions caused by the random backoff process in the 802.11 MAC algorithm. It can be applied in all cases where a throughput-sensitive data has to be broadcasted in a wireless network. The only requirement is that during the formation of the network an identification process must be implemented and an ID has to be assigned to each STAs. The characteristic of this algorithm is that increases linearly the CW size proportionally to the number of STAs in the WLAN. Therefore, for delay sensitive data, like real-time audio, the efficiency of the algorithm decrease in networks with many STAs. Simulation results showed that the algorithm provides excellent throughput with acceptable transmission delay for networks with up to 52 STAs. However, this population is considered marginal for audio networks according to the initial targets of this research and thus an improved version of the EBNA algorithm was required.

In order to resolve the delay issues in the EBNA algorithm proposed in this thesis, an improved version of the algorithm was created. This improved EBNA algorithm is called Hybrid-EBNA (H-EBNA) because it used both the classic IEEE 802.11 MAC and the EBNA modified MAC occasionally. Classic 802.11 has the advantage of very low transmission delay in broadcasting of data in a wireless network due to its constantly small CW. This is not a problem when the number of STAs that compete for access to the medium. When the number of STAs increases, the number of collisions also increases and the throughput decreases dramatically. However, the number of competing STAs is not necessary equal to the total number of STAs in the network, as it is assumed in the simple EBNA. STAs can participate in the network but they can be inactive for a time interval. This is the key characteristic that H-EBNA is taking into account. The algorithm monitors network’s activity and switches between classic 802.11 and EBNA depending of the probability of collision. When the EBNA approach is selected, the algorithm identifies within the STAs in the entire network, those who are active at any given time and implement the EBNA technique only to those. With this dual layer improvement we manage to achieve equally high throughput with suitable for audio networks overall transmission delay.

The aim of this research, according to our initial statement, was to propose solutions for the implementation of wireless audio networks that are compatible with the existing technology and able to coexist and interoperate with regular data wireless networks under the same infrastructure. In the last part of this work, the ability of the proposed methods to operate in conjunction with conventional IEEE 802.11 devices within the same network is examined and the effect of this coexistence is analysed. Simulation results shows that broadcasting music audio data in a mixed mode environment, where also a regular wireless data network operates, is possible and a higher throughput also can be achieved if both the EBNA and H-EBNA is used. However the use of H-EBNA additionally gives significantly low transmission delay in the entire network and it can be used for wireless real-time audio networks that operate sharing the same infrastructure with regular IEEE 802.11 networks.

7.3 Further work

The work presented in this thesis raises a number of research challenges that has to be addressed by the future research. These research challenges can be classified into three main areas.

The first area of research is defined by a number of open problems that must be resolved in order for the modifications, proposed in this thesis, to be able to operate as a complete alternative coordination function for wireless audio networks. This includes the design and development of association, authentication and administration techniques that will allow the formation and control of IEEE 802.11 based wireless audio networks, according to the principles proposed in this thesis.

A second significant research field that emerges from this research is related to the chance of developing a MAC level QoS for audio data traffic derived from musical execution. This can be based on a prioritization scheme, similar to the one used by the EDCA (IEEE 802.11e), to assign different priorities into different classes of data traffic. There are evidence that human perception has a variable tolerance when it comes to delay for different types of sound envelopes and different frequencies. Therefore, a further research in the psychoacoustic domain has to be conducted in order to define the boundaries of human hearing in delay, for different types of music sound. This will allows the design of a prioritization scheme that will improve not only network's performance but also the actual user satisfaction.

Finally, emerging wireless networking technologies must be researched regarding their suitability to accommodate wireless audio networking applications. There is a significant number of published standards, some of which have been also implemented in physical level, that provide exceptional bandwidth, and networking functionalities and could be remarkable candidates for carrying out of the wireless audio networks in the future.

Appendix A: Custom Statistic and Attributes in OPNET

A.1 The creation of Custom Statistic

In OPNET modeler a Custom Statistic is collected within a specific process model. Custom Statistic can be *Local* or *Global*. In this section the creation of Local Statistic is described. The process of adding a custom statistic in OPNET modeler consists of four general steps. These are:

- Declare the name of the Statistic
- Declare (create) a *Stathandle* variable to handle the Statistic
- Register the Statistic at the initial state of the process, where the Statistic is collected
- Write the code for collecting the Statistic at the appropriate state of the process

Declare the name of the Statistic

In order to create a new Custom Statistic the name of the Statistic must be declared first. To do so, we open the appropriate process model, select the *Interfaces* dropdown menu, select *Local Statistic* and declare the name of the statistic in the list as it is shown in figure A.1.1. In this figure we can see some of the Custom Statistics used in this project.

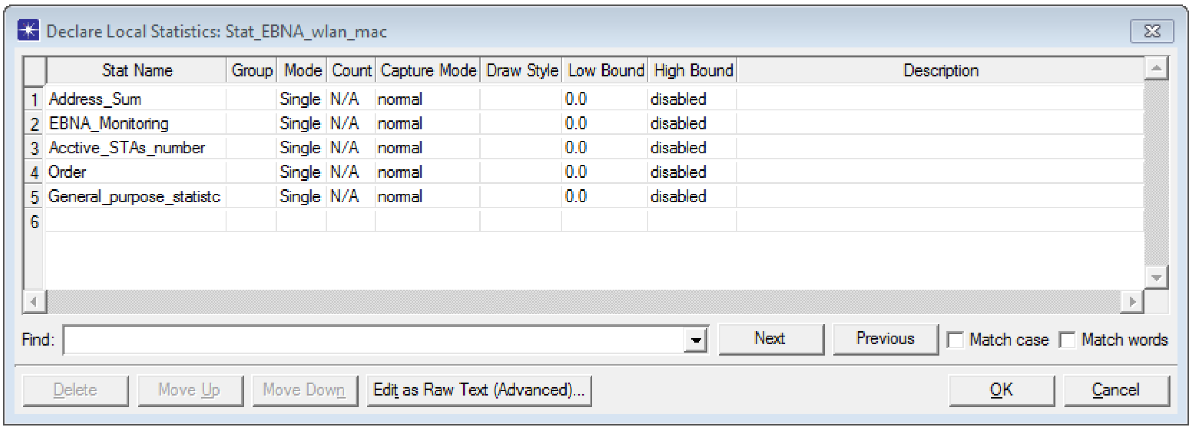

Fig A.1.1: Declaration of the name of Custom Statistics

Create a Stathandle to handle the Custom Statistic

OPNET modeler uses a specific category of variables to record Statistics. These are called *Stathandles*. In order to create a Custom Statistic a corresponding Stathandle must be declared. Stathandles are declared in the list of State Variables (SV). This list (fig A.1.2) can be accessed through the process model by selecting the SV function.

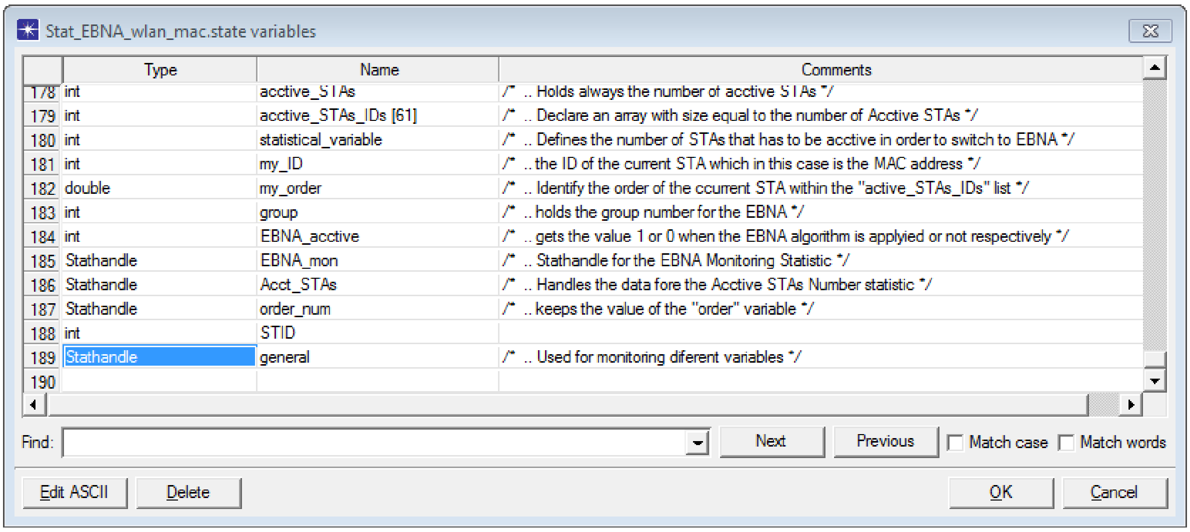

Fig A.1.2: Declaration of the Stathandle variable

Figure A.1.2 shows the Stathandles for the Statistics shown in figure A.1.1 and also some additional custom variables used in this research.

Register the Statistic at the initial state of the process

Custom Statistics has to be registered every time a process is invoked. This is achieved using the OPNET function *op_stat_reg*. The registration is taking place in the initialisation (INIT) state of the appropriate OPNET process model as it is shown in figure A.1.3.

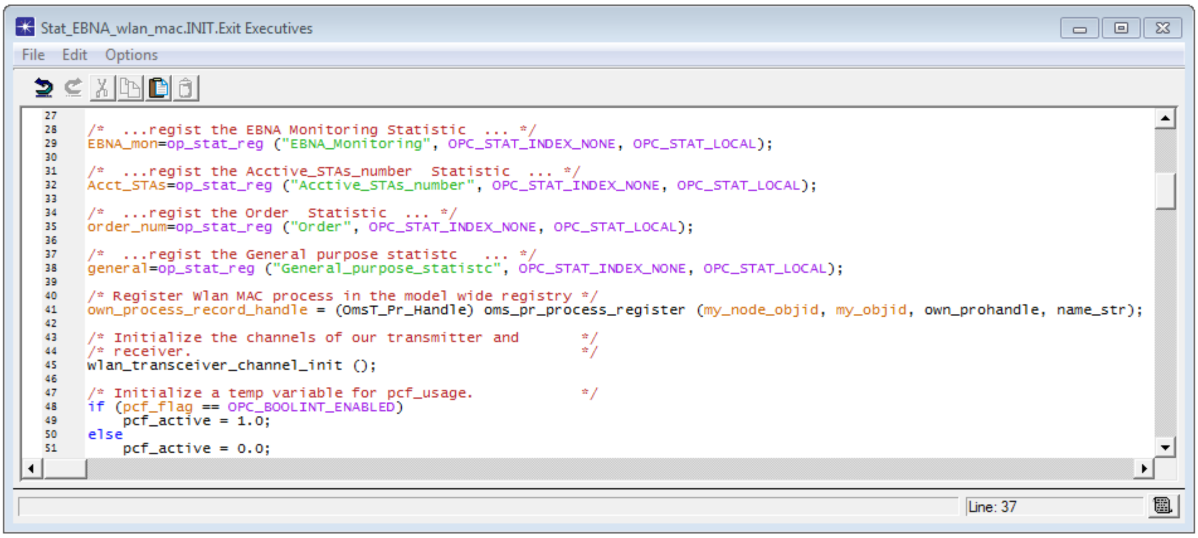

Fig A.1.3: Register the Statistic at the INIT state

Write the code for collecting the Statistic

Collecting the Custom Statistic is a very sensitive process. The code that executes this task must be placed in the appropriate state within the process model and also in the appropriate place within the general code executed in this state, in order for the Statistic to collect reliable data. OPNET function *op_stat_write* it is used for the collection of Statistic. Figure A.1.4 shows the syntax of this function as it appears in the Enter Executive of the *Backoff_Need* state of the *wireless_lan_mac* OPNET process model.

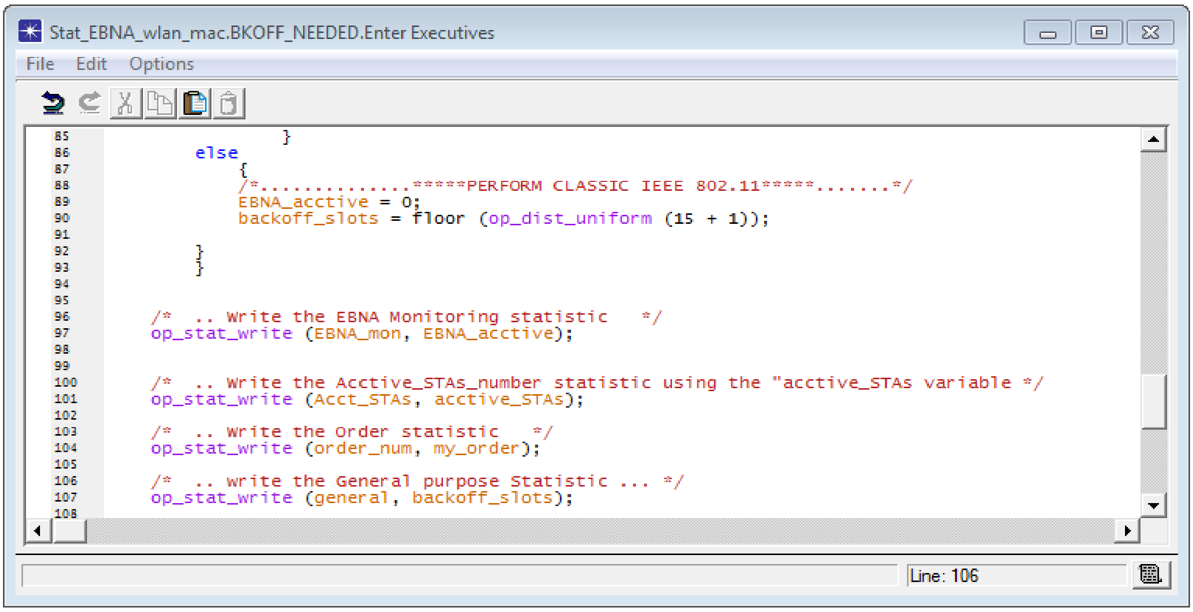

Fig A.1.4: Statistic Collection Code

A.2 The creation of Custom Attributes

Custom Attributes are used in OPNET modeler to insert additional parameters especially when the operation of a model is modified. Custom Attributes are actually static variables that are given from the user prior the simulation. Custom Attributes, once created and register, they can be entered as regular attributes through the "edit attribute" process in each node model of OPNET or they can be promoted and entered in the final configuration before running the simulation. The process of adding a Custom Attributes in OPNET modeler consists of four general steps. These are:

- Declare the name of the attribute
- Declare (create) a variable to handle the values of the attribute
- Declare (create) a temporary variable to be used as a pointer, to store the attribute's value
- Write the code to obtain the value of the attribute

Declare the name of the attribute

Custom Attributes are to be used from a specific OPNET process model. Thus, its details are set in this particular model. In order to create a new Custom Attribute the name of the Attribute must be declared first. To do so, we open the appropriate process model, select the Interfaces dropdown menu, select Model Attributes and declare the name of the Attribute in the list as it is shown in figure A.2.1. In this example the declaration of the Custom Attribute “Station ID” is described.

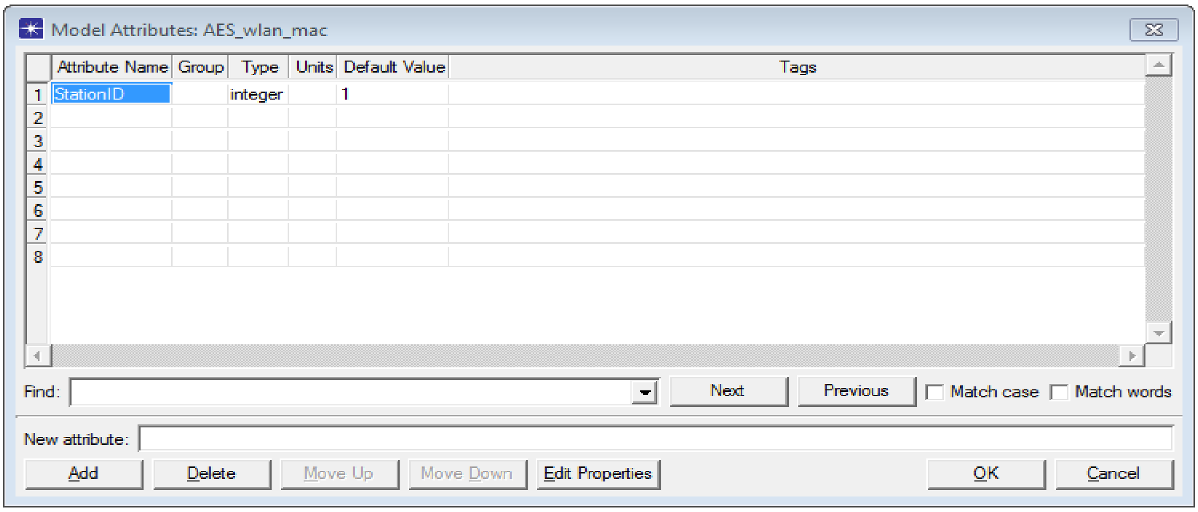

Fig A.2.1: Declaration of the name of Custom Attribute

Create a variable that handles the values of the attribute

This variable is declared within the list of State Variables and it is used by the OPNET code whenever the value of the Custom Attribute is needed. This list (fig A.2.2) can be accessed through the process model by selecting the SV function.

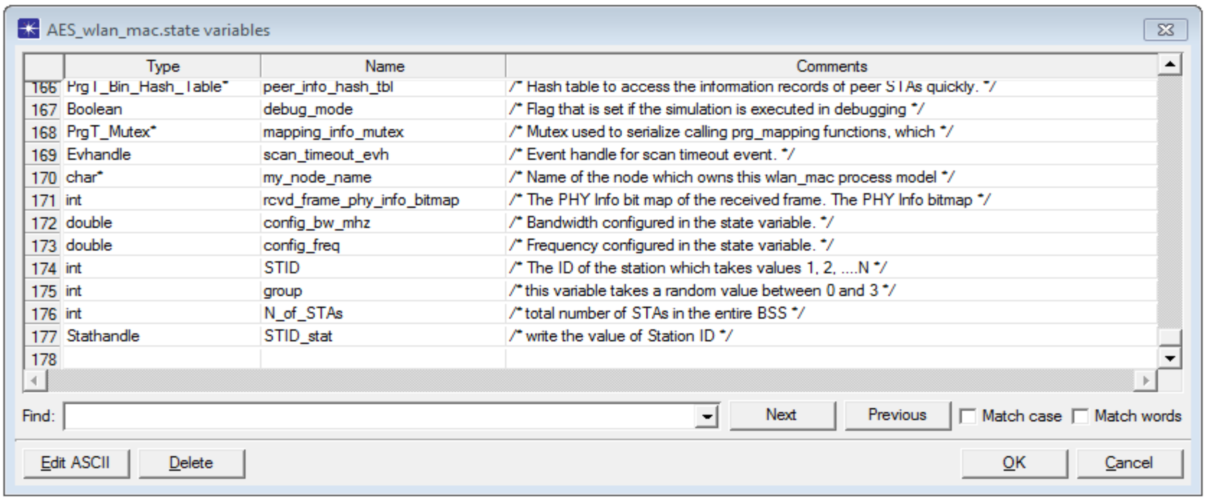

Fig A.2.2: Declaration of a SV to handle the Attribute’s value

Create a temporary variable

Custom Attribute values are stores in a pointer defined by a temporary variable. This variable must be declared within the list of temporary variables (TV) of the process model by selecting the TV function as it is shown in figure A.2.3.

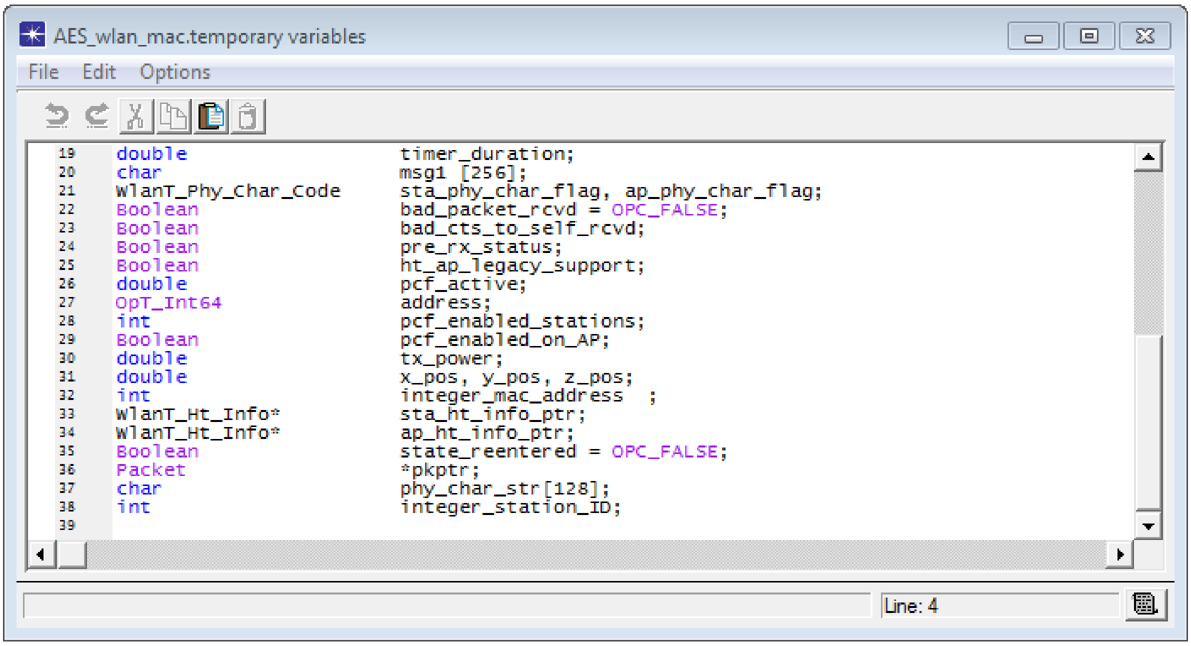

Fig A.2.3: Declaration of a SV to handle the Attribute's value

Write the code to obtain the value of the attribute

In order to use the value of a Custom Attribute provided by the user, this value must be first obtained. This is achieved by using the specific OPNET function *"op_ima_obj_attr_get()"*. The syntax of this function contains three arguments and it is shown below:

op_ima_obj_attr_get (Object ID, Attribute's name, Pointer to a variable to be filled with the attributes value)

For our example the code that has to be added in order to obtain and use the Custom Attribute "StationID" is shown below:

```
/* ... gets the Station ID value from the StationID attribute and store it in the pointer &integer_station_ID */
op_ima_obj_attr_get (my_objid, "StationID", &integer_station_ID);

/* Get the Station ID attribute       */
STID = integer_station_ID;
```

Appendix B: Code and Code Modifications

B.1 Code modifications for the implementation CTS-to-self protection

The modification of the code in OPNET's wireless model in order to implement the proposed CTS-to-self protection mechanism described in chapter 3 is presented below. All modifications are taking place in the Function Block (FB) of the *wlan_mac* process.

Adjust the bit rate of the CTS-to-self message to the operational level:

```
        /*.....   change in the control data rate... */
        control_data_rate = data_tx_rate;
```

Replace the data rate allocation algorithm of the regular CTS-to-self with the constant operational data rate:

```
/* SOS .... change the CTS bit rate.....................................*/
/*if (dcf_destination_addr < 0 ||
wlan_dest_is_11g_enabled (dcf_destination_addr) == OPC_FALSE) */
    /*{  */
    /* Pick the highest 11b data rate that is lower than our regular    */
    /* 11g data rate.                                                   */
    /*if (data_tx_rate > 11000000.0)                        */
    /*  operational_speed = 11000000.0;                     */
    /*else if (data_tx_rate > 5500000.0)                    */
    /*  operational_speed = 5500000.0;                      */

    /* .... Changed to:......*/
    operational_speed = data_tx_rate ;
    /*}  */
else
    operational_speed = data_tx_rate ;
/*} */
```

Modify the wlan_mac code to send a CTS-to-self message prior to each broadcasting data packet:

```
                /* If we are an ERP-STA, and we are not going to use an 802.11/11b data */
                /* rate for the transmission data, and there are non-ERP STAs in the    */
                /* BSS, then we need to "use protection" by sending an RTS or           */
                /* CTS-to-self message.                                                 */
                /*   ... change *2 */
                if (phy_type == WlanC_11g_PHY  && wlan_flags->polled == OPC_FALSE &&
                    (operational_speed > 5500000.0 && operational_speed != 11000000.0))

                    {
                    /* Use the "CTS-to-self" approach and send a CTS message with       */
                    /* destination address set to our own address, if CTS-to-self      */
                    /* option is enabled or the data packet is a broadcast packet.      */

                    /*  ... CHANGE *1  ............................................*/
                    if (dcf_destination_addr < 0)
                        {
                        operational_speed =  54000000.0;
                        control_data_rate = 54000000.0;
                        wlan_prepare_frame_to_send (WlanC_Cts);

                        }

                    /* Otherwise initiate a RTS/CTS exchange as the protection          */
                    /* mechanism.                                                       */
                    else
                        wlan_prepare_frame_to_send (WlanC_Rts);

                    /* Exit the function.                                               */
                    FOUT;
                    }
            }
        }
```

An adjustment for the bulk size of the data packet has to be done for the proper operation of the modified code:

```
                /*  ..... change in the bulk size by diviting  control_data_rate /2 */
                bulk_size = PLCP_OVERHEAD_CTRL (WLANC_CTS_LENGTH) * (control_data_rate/2) -
                WLANC_DEFAULT_PLCP_OVERHEAD;
                op_pk_bulk_size_set (wlan_transmit_frame_ptr, (OpT_Packet_Size) bulk_size);
                }
```

B.2 Implementation of the EBNA algorithm

The EBNA algorithm described in chapter 5 replaces the classic random backoff algorithm of IEEE 802.11 MAC. The implementation of the EBNA algorithm in OPNET is achieved by replacing the code in the *enter executive* of the “BKOFF_NEED” state of the *wlan_mac* process model with the following code:

```
/* Checking whether backoff is needed or not.                              */
if (wlan_flags->backoff_flag == OPC_TRUE || wlan_flags->perform_cw == OPC_TRUE)
    {
    if (backoff_slots == BACKOFF_SLOTS_UNSET)
        {

        /* slots according to the number of retransmissions.              */

        /*  .... the EBNA algorithm code is below... */

                /* Give the total number of STAs in the BSS  */
                N_of_STAs = 60;

                /* ... gets the Station ID value from the StationID attribute
                and store it in the pointer &integer_station_ID */
                op_ima_obj_attr_get (my_objid, "StationID", &integer_station_ID);

                /* Get the Station ID attribute        */
                STID = integer_station_ID;

                /* Create a random value for the group */
                 group = floor (op_dist_uniform (2)) +1 ;


                /*backoff_slots = STID; */

                /* generate a EBN */
                /*backoff_slots = ((group * N_of_STAs) + (STID - 1))+1; */

                if (group==1)

                    backoff_slots = STID;

                else

                    backoff_slots = (N_of_STAs * 2 - STID)+1 ;


                op_stat_write (STID_stat, backoff_slots);

        }

    /* Set a timer for the end of the backoff interval.                    */
    intrpt_time = (current_time + backoff_slots * slot_time);

    /* Scheduling self interrupt for backoff.                              */
    if (wlan_flags->perform_cw == OPC_TRUE)
        backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_CW_Elapsed);
    else
        backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_Backoff_Elapsed);

    /* Reporting number of backoff slots as a statistic.                   */
    op_stat_write (backoff_slots_handle, backoff_slots);
    }
```

B.3 Implementation of the H-EBNA algorithm

Implementation of H-EBNA in C++

Due to its complexity, the H-EBNA algorithm was first developed in C++ in order to verify its proper operation. The "general list" of active STAs as well as other variables that the algorithm in its normal operation obtains from the simulation, are provided manually in this implementation. The code of this test version of the H-EBNA is listed below:

```
          // *********** STATISTICAL EBNA IMPLEMENTATION ALGORITHM ****************//
          // This code is applied in BACKOFF_NEED state in OPNET wireless model.   //
          // It gets the "General List" array which is kept and updated in the     //
          // reception process, and from there calculate the number of "Active     //
          // STAs and also the "order" of the current STA in the list of currently //
          // active stations. Then, according to the given "statistical variable   //
          // it decide whether EBNA or Classic WiFi will be implemented. Finally,  //
          // Prints information about the decision parameters..................... //


#include <iostream>
#include <stdio.h>      /* printf, NULL */
#include <stdlib.h>     /* srand, rand */
#include <time.h>       /* time */
using namespace std;

int Threshold; //the value of time that defines if a STA is active//
int Statistic_variable; // The number of STAs has to be active in order to switch to EBNA //
int My_ID; // The STA ID of the current STA
int n;
int m=0;
int order ;
int Backoff_slots;
int acctive_STAs; // The number of Active STAs in the BSS//
double General_list [10] = {100, 10, 50, 75, 120, 20, 500, 27, 82, 5,}; // the General List//

int main ()
{
    // Enter all Variables//
    cout << "Enter the Threshold :";
    cin >> Threshold ;
    cout << endl;
    cout << "Enter the Statistic_variable:";
    cin >> Statistic_variable;
    cout <<endl;
    cout << "Enter My ID:";
    cin >> My_ID;
    cout <<endl;

    // Find the number of Active STAs//
    for (n=0; n<10; n++)
    {
        if (General_list [n]< Threshold)
        acctive_STAs = acctive_STAs+1;
    }

    // Create the List of Active STAs IDs //
    int acctive_STAs_IDs [acctive_STAs];  // declare an array with size equal to the number
of Active STAs//
        for (n=0; n<10; n++)
        {
            if (General_list [n]< Threshold)
            {
               acctive_STAs_IDs [m] = n; // feels the active STA ID array with the active
STAs IDs//
               m=m+1;
            }
        }
```

```
    // Choose Backoff Method //
    if (acctive_STAs > Statistic_variable)
       {

           // ******PERFORM  EBNA*******//
           // first find my order number//

           for (n=0; n<acctive_STAs; n++)
           {
               if (My_ID == acctive_STAs_IDs [n])

               order = n;
           }

           int Group; // declare the group variable//

           // generate random number between 1 and 2: //
            srand (time(NULL));    // This is a type of "seed" based on time and force the
function "rand" //
                                   // to give diferent number for each run//
            Group = rand()% 2 +1;  // Attention...!! the syntax [rand()%2] gives numbers
betwwen 0 and 1 //

            if (Group < 2)
               Backoff_slots = order;
            else
               Backoff_slots = 2 * acctive_STAs - (order + 1);

            cout << "***EBNA IS PERFORMED***" << endl;      // print information about EBNA//
            cout << "the Number of Backoff Slots is:";
            cout << Backoff_slots << endl;
            cout << "The selected Group is:" << Group;
            cout << endl;

        }
    else // *****PERFORM CLASSIC 802.11 RANDOM BACKOFF USING CWmin**********//
            {
            srand (time(NULL));
            Backoff_slots = rand() % 16;  // Generate random variables between 0 and 15 //

            cout << "***CLASSIC WiFi IS PERFORMED***" << endl;
            cout << "the Number of Backoff Slots is:";
            cout << Backoff_slots;
            }

cout << endl;                                                   // Print relevant information//
cout << "The number of acctive STAs is: " <<  acctive_STAs;
cout << endl;
cout << "The List of Acctive STAs IDs is:" << endl;
for (n=0; n<acctive_STAs; n++)
        {
        cout << acctive_STAs_IDs [n] << endl;
        }
cout << "The Order Value is: " << order <<endl;

    system ("pause");
    return 0;
}
```

Implementation of the H-EBNA in OPNET

The implementation of the H-EBNA in OPNET consists of two parts. The first one is an additional code running within the reception process *(wlan_mac process model>Function Block> function: [wlan_physical_layer_data_arrival])* and creates and updates the list of active STAs in the wireless network. The second part is the code executed in the *enter* *executive* of the “BKOFF_NEED” state of the *wlan_mac* OPNET process model and implement the H-EBNA concept as it is described in chapter 5.

The first part of the H-EBNA code, responsible for the generation and maintenance of the *general list of active STAs* is shown below:

```
    /*  ... the folowing code creats the General List of active STAs....*/


        if (rcvd_frame_type == WlanC_Cts && remote_sta_addr <60  )
            {
            general_list [remote_sta_addr]=current_time;    /* ...set current simulation time in the
                                                            location of the array indicated by the MAC ADDRESS*/

            op_stat_write (sum_of_addr, general_list [10]);/* ...Use this statistict to monitor one of
                                                            the locations of the array (general list) */
            }
```

The second part of the H-EBNA code, located in the enter executive of the “BKOFF_NEED” state of the wlan_mac OPNET process model, is shown below:

```
/** ....In this state we implement the H-EBNA consept.....   **/


/* Checking whether backoff is needed or not.                          */
if (wlan_flags->backoff_flag == OPC_TRUE || wlan_flags->perform_cw == OPC_TRUE)

    {
    if (backoff_slots == BACKOFF_SLOTS_UNSET)
        {

        /*.. SOS .. Give values to variables 1) Statistical_variable 2)Threshold_time */
            statistical_variable=0.9;
            Threshold_time=0.06;

            z=0;
            m=1;
            j=1;



        /*S O S ... gets the Station ID value from the StationID attribute and
        store it in the pointer &integer_station_ID */
                op_ima_obj_attr_get (my_objid, "STA_ID", &integer_station_ID);
            /* Get the Station ID attribute           */
                STID = integer_station_ID;


        /*.. SOS ... Implements the Statistical EBNA Algorithm ............... */

            /*.. SOS Find the total number of acctive STAs */
            for (n=1; n<26;  n++)
                {
                    if ((current_time - general_list [n])< Threshold_time )
                        acctive_STAs=acctive_STAs+1;
                }

            /*.. SOS .. Calculate the probability of collision according to the Acctive STAs */
            /* using the formula: p=1-[(1-(1/CW))]^N-1, ware N is the number of acctive STAs */

            z= acctive_STAs;
            z=z-1;
            j=0.9375;
            j=pow(j,z);
            p=1-j;          /*probability of collision */



            /*.. SOS Creat the list of Acctive STAs IDs .../
            for (n=1; n<26; n++)
                {
                    if (((current_time - general_list [n]) < Threshold_time &&
                        (current_time - general_list [n])!= current_time)|| general_list [n] == 0 )
                        {
                            acctive_STAs_IDs [m] = n;
                            m=m+1;
                        }
                }

            /* find the "order" of the current STA in the acctive_STAs_IDs list */
                for (i=1; i<(acctive_STAs+1); i++)
                    {

                        if (acctive_STAs_IDs [i] == STID)
                            my_order=i;

                        }

        /*........ SOS Choose backoff method ........................... */
        if (acctive_STAs >25 )
                {
```

```
                    -
                    /* .................****PERFORM EBNA******.......... */

                    EBNA_acctive = 1;

                    /* Create a random value for the group between 1 & 2 */
                     group = floor (op_dist_uniform (2)) +1 ;

                    if (group<2)
                        {
                        backoff_slots=my_order;

                        }
                    else
                        {
                        backoff_slots = 2 * acctive_STAs - (my_order-1);


                        }
                    }
            else
                {
                /*.............*****PERFORM CLASSIC IEEE 802.11*****......*/
                EBNA_acctive = 0;
                backoff_slots = floor (op_dist_uniform (15 + 1));

            }
            }


    /* SOS .. Write the EBNA Monitoring statistic   */
    op_stat_write (EBNA_mon, EBNA_acctive);


    /* SOS .. Write the Acctive_STAs_number statistic using the "acctive_STAs variable */
    op_stat_write (Acct_STAs, acctive_STAs);

    /* SOS .. Write the Order statistic   */
    op_stat_write (order_num, my_order);

    /* SOS .. write the General purpose Statistic ... */
    op_stat_write (general, backoff_slots);

    /* ................***********RESETS*******.............*/
    acctive_STAs=1; /* SOS .. Reset acctive_STAs variable... VERY IMPORTANT..*/
    /*my_order=0;*/

    /*SOS .. Reset the acctive_STAs_IDs array */
        for(n=1; n<26; n++)
        acctive_STAs_IDs[n] = 0;



    /* Set a timer for the end of the backoff interval.                 */
    intrpt_time = (current_time + backoff_slots * slot_time);

    /* Scheduling self interrupt for backoff.                           */
    if (wlan_flags->perform_cw == OPC_TRUE)
        backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_CW_Elapsed);
    else
        backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_Backoff_Elapsed);

    /* Reporting number of backoff slots as a statistic.                */
    op_stat_write (backoff_slots_handle, backoff_slots);
    }
```

REFERENCES:

[1] Gary Davis Ralph Jones, Sound Reinforcement Handbook, Yamaha-Hal Leonard Corporation, 1986

[2] Audio over Ethernet, Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Audio_over_Ethernet. [Accessed: June, 14, 2014]

[3] Steve Church Skip Pizzi, Audio over IP “Building Pro AoIP Systems with Livewire”, Elsevier, 2010

[4] Kevin Gross, white paper, “Digital Audio Distribution Systems”, Cirrus Logic Inc. 2004

[5] EtherSound, Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/EtherSound. [Accessed: May, 10, 2014]

[6] EtherSound Overview, an Introduction to the ES-100 Technology. [Online]. Available: http://www.ethersound.com/technology/overview.php. [Accessed: May, 11, 1014]

[7] CobraNet, the Standard in Networked Audio. [Online]. Available: http://www.cobranet.info. [Accessed: May, 11, 1014]

[8] YAMAHA System Solutions white paper, “Networked audio system design with CobraNet”, Yamaha Commercial Audio, 2006

[9] Aviom Audio Networks. [Online]. Available: http://www.aviom.com/AviomApplications/AudioNetworks/Basics.php . [Accessed: May, 15, 1014]

[10] AudioRail. [Online]. Available: http://www.audiorail.com/simple.htm . [Accessed Jan. 12, 2014]

[11] H. Juszkiewicz, N. Yeakel, S. Arora, A. Beliaev, R. Frantz and J. Flaks, “Media-accelerated Global Information Carrier (MaGIC)”, Gibson Guitar Corporation, Engineering Specification, Revision 3.0c, 2003

[12] Audinet, Dante Overview. [Online]. Available: https://www.audinate.com/solutions/dante-overview. [Accessed: Feb. 22, 2014]

[13] SuperMAC (AES50-Compliant) and HyperMAC digital audio interconnections. [Online]. Available: http://www.supermac-hypermac.com. [Accessed: Jan. 10, 2014]

[14] AES white paper, Technology Report TC-NAS 98/1: "Networking audio and music using internet2 and next-generation internet capabilities", Audio Engineering Society, 1998

[15] Audio Engineering Society, [Online]. Available: http://www.aes.org/technical/documentDownloads.cfm?docID=38. [Accessed: June, 10, 2014]

[16] AES, "Standard for digital audio-Digital input-output interfacing, Serial transmission format for two-channel linearly-represented digital audio data", AES3-2002, incorporating amendments 5 & 6, 2009

[17] AES, "Recommended Practice for Digital Audio Engineering - Serial Multichannel Audio Digital Interface (MADI)", AES10-2008 Revision of AES10-2003, Audio Engineering Society, 2008

[18] AES47, "Standard for digital audio Digital input-output interfacing Transmission of digital audio over asynchronous transfer mode (ATM) networks", Audio Engineering Society, 2006

[19] AES50, "Standard for digital audio engineering High-resolution multi-channel audio interconnection (HRMAI)", Audio Engineering Society, 2005

[20] AES67, "Standard for audio applications of networks - High-performance streaming audio-over-IP interoperability", Audio Engineering Society, 2013

[21] EBU Tech 3329, "A Tutorial on Audio Contribution over IP", European Broadcasting Union, 2008

[22] EBU Tech 3326, "Audio Contribution over IP - Requirements for Interoperability", European Broadcasting Union, 2008

[23] R. Foss, J. R. Cooperstock, Guest editor's note, Special issue on audio networking, Journal of Audio Engineering Society, Vol. 61, No. 4, 2013

[24] AES-Technical Committee on Network Audio Systems, white paper, "Best Practices in Network Audio", Audio Engineering Society, 2009

[25] IEEE 802.15.3b Standards, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs), Amendment 1: MAC Sublayer, IEEE, 2005

[26] D. Jakubisin, M. Davis, C. Roberts, and I. Howitt. "Real-time audio transceiver utilizing 802.11 b wireless technology", In SoutheastCon, 2007 Proceedings, IEEE, pp. 692-697, IEEE, 2007.

[27] N. Seppo, T, Lindeman, and V. Manea, "Wireless High Definition Multichannel Streaming Audio Network Technology Based on the IEEE 802.11 Standards." In Audio Engineering Society Convention 130, Audio Engineering Society, 2011

[28] C.K. Toh, W.K. Tsai, V. Li, A.D. Scott, and G. Guichai, "Transporting audio over wireless ad hoc networks: experiments & new insights", In Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on, vol. 1, pp. 772-777. IEEE, 2003

[29] N.A. Tatlas, A. Floros, and J. Mourjopoulos, "Wireless digital audio delivery analysis and evaluation", In International Conference in Acoustics, Speech and Signal Processing ICASSP 2006 Proceedings, vol. 5, pp. V-V, IEEE, 2006

[30] K. Curran, "Introducing song form intelligence into streaming audio", In Information Technology: Coding and Computing Conference (ITCC 2003) Proceedings, pp. 701-705. IEEE, 2003

[31] K. Lavangnananda, C. Angsuchotmetee, and P. Bouvry. "Effect of packetization interval on number of connections in AAC audio streaming over WLAN 802.11 g", In 17th Asia-Pacific Conference Communications (APCC), pp. 530-535, IEEE, 2011

[32] ITU-T, P.800, "Methods for subjective determination of transmission quality", International Telecommunication Union, 1996

[33] ITU-T Rec. G.107, "The E-model", [Online]. Available: http://www.itu.int/ITUT/studygroups/com12/emodelv1/tut.htm. [Accessed: Sep. 15, 2013]

[34] M. Ahmed, H. Afifl, and D. Zeghlache. "Packet-e-model: e-model for wireless VoIP quality evaluation", In Personal, Indoor and Mobile Radio Communications, (PIMRC 2003), vol. 3, pp. 2421-2425, IEEE, 2003

[35] ITU-TREC-P.862, “Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs”, [Online]. Available: http://www.itu.int/rec/T-REC-P.862, [Accessed: January. 12, 2014]

[36] G. Zeynep, "Security mechanisms and their performance impacts on wireless local area networks", In 2006 International Symposium on Computer Networks proceedings, 2006

[37] N. Arlen, et al. "Can I add a secure VoIP call?", Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE Computer Society, 2006

[38] R.W. Benjamin, B.E. Mullins, R.W. Thomas, and T.R. Andel, "Subjective Audio Quality over Secure IEEE 802.11 Wireless Local Area Networks", In Performance, Computing and Communications Conference (IPCCC 2008, pp. 469-474, 2008

[39] M. Purat and T. Ritter, “Comparison of Receiver-Based Concealment and Multiple-Description Coding in an 808.11-Based Wireless Multicast Audio Distribution Network,” Journal of Audio Engineering Society, vol. 59, 2011

[40] K. Gross, “Audio Networking: Application and Requirements”, Journal of Audio Engineering Society, vol. 54, pp. 62-66, 2006

[41] IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, 2007

[42] X. Ma and X. Chen “Performance Analysis of IEEE 802.11 Broadcast Scheme in Ad-Hoc Wireless LANs,” IEEE Transaction on Vehicular Technology, vol. 57, no. 6, pp. 3757-3768, 2008

[43] N. Karthikeyan, V. Palanisamy, K. Duraiswamy, “Performance comparison of broadcasting methods in mobile ad hoc network”, International Journal of Future Generation Communication and Networking, vol. 2, no. 2, pp. 47-58, 2009

[44] J. Lipman, H. Liu, I. Stojmenovic, “Broadcast in Ad Hoc Networks”, Chapter 6, “Guide to Wireless Ad Hoc Networks”, Springer-Verlag, London, 2009

[45] B. Williams, T. Camp,"Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks", Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 194-205, 2002

[46] M. Gerla, J.T.C. Tsai, "Multicluster, Mobil Multimedia Radio Network", Journal of Wireless Networks, vol. 1, no. 3, pp. 255–265, 1995

[47] S. Ni, Y. Tseng, Y. Chen and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad- hoc Network", International Workshop on Mobile Computing and Networks, pp. 151–162, 1999

[48] W. Peng, X. C. Lu, "On the reduction of broadcast redundancy in mobile ad-hoc networks", In Proceedings 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing, pp. 129–130, 2000

[49] W. Lou, J. Wu, "On Reducing Broadcast Redundancy in Ad hoc Wireless Networks", IEEE Transactions on Mobile Computing, vol. 1, no. 2, pp.111–122, 2002

[50] J. Sucec, I. Marsic, "An Efficient Distributed Network-wide Broadcast Algorithm for Mobile Ad-hoc Networks", CAIP Technical Report 248 - Rutgers University, 2000

[51] J. Lipman, P. Boustead, JF. Chicharo, "Reliable optimised flooding in ad hoc networks", Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies, vol. 2, 521-524, 2004

[52] I. Stojmenovic, M. Seddigh, J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks", IEEE Transactions on Parallel and Distributed Systems, vol. 3, no. 1, pp. 14 – 25, 2002

[53] C. Fragouli, J. Widmer, J.Y.L. Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice", Proceedings of IEEE INFOCOM, 2006

[54] B. Li, E. Li, R.R. Milind, and S. Miller, "Network coding-based broadcast in mobile ad-hoc networks", IEEE INFOCOM, 2007

[55] K. Wehrle, M. Günes, J. Gross, "Modelling and Tools for Network Simulation", Springer, 2010

[56] OPNET Modeler 17.1, Software documentation, OPNET Technologies, Inc © 1986-2011

[57] A.V. Aho, J.E. Hopcroft, J.D. Ullman, "The design and analysis of computer algorithms", Addison-Wesley, 1974

[58] 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, (Revision of IEEE Std. 802.11-2007), IEEE, 2012

[59] "Ensuring 802.11n and 802.11a/b/g Compatibility", Whitepaper, AirMagnet, 2008

[60] D.D. Coleman, D.A. Westcott, "Certified Wireless Network Administrator Official Study Guide", John Wiley & Sons, 2012

[61] D. Vassis, G. Kormentzas, A. Rouskas, I. Maglogiannis, "The IEEE 802.11g Standard for High Data Rate WLANs", Network IEEE, vol. 19, no. 3, 2005

[62] Meraki, White Paper, "802.11n Technology", Meraki Inc, 2011

[63] L. Hai, V.T. Sakurai, "Collision Probability in Saturated IEEE 802.11 Networks", Australian Telecommunication & Application Conference (ATNAC), 2006

[64] C. Chatfield, "Statistics for Technology: A Course in Applied Statistics", Chapman & Hall, London, 1978

[65] T. Zhang, L. Lei, J. Zhou, L. Qi, "Collision Probability Analysis in Multi-hop Ad hoc Networks", Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2012

[66] C. Chousidis, R. Nilavalan, L. Lipan, "Expanding the use of CTS-to-Self mechanism for reliable broadcasting on IEEE 802.11 networks", International Wireless Communications and Mobile Computing Conference (IWCMC), Cyprus, 2014

[67] Sibelius Academy, "Music Theory 1, Time", [Online]. Available: http://www2.siba.fi/muste1/index.php?id=102&la=en, [Accessed: June 11, 2014]

[68] Wikipedia, "Tempo", [Online]. Available: http://en.wikipedia.org/wiki/Tempo, [Accessed: June 11, 2014]

[69] P. Fraisse, "Rhythm and Tempo", The Psychology of Music, pp. 149-180, Academic Press, New York, 1982.

[70] D. Moelants, "Dance music, movement and tempo preferences", Proceedings of the 5th triennial ESCOM Conference, Hanover University of Music and Drama, 2003

[71] D. Moelants, "Preferred tempo reconsidered", Proceedings of the 7th international conference on music perception and cognition, 2002.

[72] D. Moelants, M. McKinney, "Tempo perception and musical content: What makes a piece fast, slow or temporally ambiguous", Proceedings of the 8th International Conference on Music Perception and Cognition, 2004.

[73] C. Chousidis, R. Nilavalan, A. Floros, "Enhancement of IEEE 802.11 in Handling Multiple Broadcasting Audio Data in Wireless ad-hoc Networks", Journal of the Audio Engineering Society, vol. 61, no. 4, pp. 165-173, 2013

[74] S, Cheikh, I Guérin-Lassous. "Estimating average end-to-end delays in IEEE 802.11 multihop wireless networks", Institute National de Recherche en Informatique et en Automatique, 2007

[75] M. Burton, white paper, "802.11 Arbitration,", Certified Wireless Network professional, (CWNP), 2009

[76] G.F. Shay, "Taking the 'Sting' Out of Evolving Digital Audio Networks", white paper, The Telos Alliance, 2013

[77] Axia Essentials: "Understanding the AES67 AoIP Interoperability Standard", Axia Audio, 2013

[78] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement" INFOCOM'98, 17th Annual Joint Conference of the IEEE Computer and Communications Societies, 1998

[79] C. Christos, R. Nilavalan, "Modifying the IEEE 802.11 MAC to improve performance of multiple broadcasting of multimedia data in wireless ad-hoc networks", International Journal of Advanced Computer Science and Applications (IJACSA), 2013

[80] C. Christos, R. Nilavalan, "Improving multiple broadcasting of multimedia data traffic in wireless ad-hoc networks", Science and Information Conference (SAI), IEEE, 2013

[81] Z. Lu, H. Yang, "Unlocking the Power of OPNET Modeler", Cambridge University Press, 2012