Traffic Aware Backoff Scheduling of Low Data Rate Applications in Wireless Body Area Networks

By

NESAE MOUZEHKESH PIR BORJ

Thesis Submitted to Charles Sturt University, in Fulfillment of the Requirements for the Degree of Doctor of Philosophy

December, 2014

DEDICATION

To my parents, the only true angels of my life, and to my soulemate and husband, Saman, without whom I would have never been able to accomplish this.

ABSTRACT

In recent years Wireless Body Area Networks, or WBAN for short, have emerged as an extension to conventional wireless sensor networks to comply with the need to provide timely and effective hospital care, especially in urgent and mass casualty situations where a lack of time or personnel is observed. An increase in the elderly population in the world in the last few years has also given rise to a range of different techniques for telemedicine applications to be deployed, both in-hospital and for remote patient health monitoring. As the market grows in diversity in wireless body area networks offered by different vendors in the industry, there is a need for a strong infrastructure in the way these small networks operate. The main focus of this thesis is on the low data rate, short-range wireless communications established between sensor nodes on the body of a patient to monitor certain vital signs and a gateway device commonly known as a coordinator in a star topology. There are many concerns to consider for maintaining a desired level of Quality of Service (QoS) in terms of reliability, energy efficiency and timely arrival of sensory data to where the caregiver is supposed to respond. IEEE 802.15.4 is the most popular standard used in academia and in many of the practical and pilot study works around the world as the standard used for the communication links between the sensors and their coordinator device. It defines the specifications of the PHY and MAC layers of the protocol stack upon which the sensors establish their communication link with the coordinator on the body. Other wide-range and high data rate communication protocols such as GPRS may be used to relay the sensed data from the coordinator (PDA/Smart phone) to a wireless local area network and ultimately other networks, all inter-connected by the internet. Considering the asymmetry in resources that exists between the physiological sensors on the body and the coordinator as a smart device, the star topology has proved to be the most efficient topology so far for such small networks. Sensor nodes on the body send the sensory data to the coordinator based on a beacon-enabled access mode defined in IEEE 802.15.4 MAC specifications. It operates upon a super frame structure where the access to the shared medium is obtained through a combination of contention-based and schedule-based access techniques. In a wireless body area network with diverse application requirements imposed by each sensor node, it is very

difficult to organize the access to the medium while always satisfying all the desired QoS parameters. This research addresses the heterogeneity of applications and traffic diversity in wireless body area networks as the main problem, and studies an effective backoff algorithm in the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) procedure of the IEEE 802.15.4 to optimize reliability in a low data rate WBAN with periodic traffic generated. A fuzzy logic system is used to interpret two selected parameters during an interval of time, defined in terms of super frames, upon which a maximum bound on the backoff window will be decided to achieve fair access among all nodes in the WBAN to the shared medium. The primary contributions of the research are achieving a high level of reliability whilst retaining the energy consumption level. Less delay is experienced for the average number of packets received at the coordinator device when deploying our proposed fuzzy-enabled MAC compared to the original IEEE 802.15.4 MAC performance. The proposed MAC technique has been tested both through simulations (using the Castalia simulator) and experimental set up (using SHIMMER sensor platforms).

AKNOWLEDGMENTS

I am grateful to my principal supervisor, Dr. Tanveer Zia, for being a great inspiration throughout my candidature, for always leading me to the right direction whenever I became meticulous in my way of research as I always do! His great advice, arguments, discussions, and admirable patience in my work, from the beginning to the end, really helped me to get through the difficult stages of my thesis. I would also like to thank him for letting me be on such a wonderful journey by agreeing to be my supervisor in 2011. Since then he has also been a tremendous mentor and gave me that bit of encouragement I needed to keep me going whenever things seemed to be falling apart.

My deep appreciation goes to my co-supervisor, Dr. Lihong Zheng, for her useful comments, remarks, and engagement throughout my research. Her constant support contributed a lot to completion of this thesis: even in her busiest days she never hesitated to allocate her time to provide me with her guidance.

I would like to express my deepest gratitude to Saman Shafigh, for his willingness and endless help on technical aspects of my work, where lack of sufficient resources and materials on the required coding- and hardware-related parts would hugely slow me down. This thesis could not have reached its current stage without his experience and expertise in the field. Those long hours of discussions and coding together, on- and off-campus, finally paid off to get us through our experiments. I cannot thank him enough for having played such an important role, especially towards the implementation phases of the work, and for never letting me feel alone in such a big task.

I would also like to thank Shimmer Research group, Jong Chern Lim and Mike Healy, for their great technical support of their products. Great credit goes to Jong Chern Lim, the application engineer of Shimmer team, for his support with programming problems of TinyOs-1, the operating system of Shimmer sensor platforms. Mike Healy helped me greatly with measuring and monitoring the energy of Shimmer sensor platforms. I also appreciate all the technical help I received from Jan-Hinrich Hauer from the Telecommunication Network Group at the faculty of Electrical Engineering and

Computer Science at Technische Universität Berlin. His massive help with TKN15.4, the only implementation of the IEEE 802.15.4 MAC in TinyOS, really helped me with the final stages of my work.

Thanks to my colleagues in the school of Computing and Mathematics in Charles Sturt University: my officemate, Lisa, and other PhD students in our building: Adrian, Carolina, Geaur, Javid, Jason, Robin, Sabih, Saeed, Sharon, and Sohail, for all the positive energy and enthusiasm they gave me. It would have been much harder, without you all, to overcome those frustrating days of research when nothing seemed to be working right! Thank you guys, I shall never forget our memories in Building 1.

A special thank to the lovely and encouraging staff in our building: Amna Sabih, Annie Andersen, Catherine Wade, Chelsea Williams, Dmitry Demskoy, Jodie Mitchell, Kerrie Cullis, Geoff Fellows, Prof. Kenneth Russel, Sharon Nielsen, Shona Cameron, and Sue Kendall. You made us feel at home and made it simply a peaceful and lovely place to be and work in. It is hard for me to believe it has almost been four years since I joined you in CSU. My memories with you will stay with me forever. I have also a huge appreciation and gratitude to Jacqueline Blomfield, the International Student Support Officer in Wagga Wagga campus, for always being there for me, especially in the early days of my candidature. I always felt blessed with her encouraging words and warm support.

Last but not least, I would like to express my warmest thanks and love to my beautiful parents and family back home. It is never easy to study while being apart from family, but their love and support have always inspired me to go beyond what I could imagine. You taught me not to break by the volume of the work ahead of me and to just face, live, and enjoy each day as it passes. You make me feel determined by being such a wonderful motivation in my life, and to always stand by my side no matter how far the distance. God bless you. And a big thank to the love of my life, Saman, for always being there for me, sweet or sour, you have proved your support throughout and I could not be any happier in life than having you by my side.

This thesis has been edited by professional editor, Dr Gaye Wilson, according to the Guidelines for Editing Research Theses agreed in 2004 by the Deans and Directors of

Graduate Studies. All remaining errors are my own. She has done such an amazing job in editing my thesis and proofreading my work. I sincerely appreciate her huge efforts and the amount of time she took to do this for me.

PUBLICATIONS

Publications resulted from this thesis:

Conferences:

1. Mouzehkesh, N., Zia, T., Shafigh, S., & Zheng, L. (2013). Light-Weight, History-Based Medium Access Control (MAC) Protocol for Body Area Networks. *Proc. of the 7th International Conference on Sensing Technology*, Wellington, NewZealand, 91-96. doi: 10.1109/ICSensT.2013.6727622.

2. Mouzehkesh, N., Zia, T., Shafigh, S., & Zheng, L. (2013). Towards A Reliable Traffic-Aware Medium Access Control (MAC) Protocol Design for Body Area Networks. *Short paper submitted to the 3rd Annual Higher Degree by Research Symposium*, Charles Sturt University, Wagga Wagga. (Published in symposium catalogue). Received best presentation award for this work.

3. Mouzehkesh, N., Zia, T., Shafigh, S., & Zheng, L. (2013). D2MAC: Dynamic Delayed Medium Access Control (MAC) Protocol with Fuzzy Technique for Wireless Body Area Networks. *IEEE BSN 2013, the 10th International Conference on Wearable and Implantable Body Sensor Networks*, MIT, Cambridge, (USA), 1-6. doi: 10.1109/BSN.2013.6575472.

4. Mouzehkesh, N., Zia, T., Shafigh, S., & Zheng, L. (2013). Traffic Aware Fuzzy-tuned Delay Range for Wireless Body Area Networks Medium Access Control Protocol (MAC). *8th International Conference on Intelligent Sensors and Sensor Networks*, Melbourne, Australia, 60-65. doi: 10.1109/ISSNIP.2013.6529765.

5. Mouzehkesh, N. & Zia, T. (2011). A Dynamic Backoff Approach in Wireless Sensor Networks for Environmental Monitoring. *Seventh International Conference on Intelligent Sensors, Sensor Networks, and Information Processing*, Adelaide, Australia, 247-252. doi: 10.1109/ISSNIP.2011.6146532.

Journals:

6. Mouzehkesh, N., Zia, T., Shafigh, S., & Zheng, L. (2014). Dynamic Backoff Scheduling of Low Data Rate Applications in Wireless Body Area Networks. (Submitted to *Journal of Wireless Networks: The Journal of Mobile Communication, Computation and Information, Springer*, June 2014).

7. Mouzehkesh, N., Zia, T., & Shafigh, S. (2015). Suitable Design of MAC Protocols for Wireless Body Area Network Applications: A Survey. (Submitted to *ACM Transactions on Sensor Networks (TOSN)*, January 2015).

Related publications:

8. Shafigh, S., Zia, T., & Mouzehkesh, N. (2013). Wireless Acceleration Sensor Data Filtering Using Recursive Least Squares Filter. *8th International Conference on Intelligent Sensors and Sensor Networks*, Melbourne, Australia, 66-70.
doi: 10.1109/ISSNIP.2013.6529766.

9. Shafigh, S., Zia, T., & Mouzehkesh, N. (2012). Position Aware MAC (PA-MAC) Protocol for Wireless Body Area Networks. *IB2COM: The 2012 International Conference on Broadband and Biomedical Communications*, University of Technology Sydney, Australia.

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

ACK	Acknowledgment
ACLS	Advanced Cardiovascular Life Support
ALOHA	A type of random access control protocol Originally developed for packet radio communications at the campuses of U. of Hawaii in 1970
ATM	Automatic Teller Machine
BAN	Body Area Network
BCU	Body Central Unit
BE	Backoff Exponent
BEB	Binary Exponential Backoff
BI	Beacon Interval
BLE	Battery Life Extension
BO	Beacon Order
BPSK	Binary Phase Shift Keying
BSN	Body Sensor Network
CAC	Connection Admission Control
CAP	Contention Access Period
CBR	Constant Bit Rate
CCA	Channel Clear Assessment
CDMA	Code Division Multiple Access
CE	Conformité Européenne; mandatory conformity marking for certain products sold within the European Economic Area (EEA) since 1985
CFP	Contention Free Period
CICADA	Cascading Information retrieval by Controlling Access with Distributed slot Assignment
CPU	Central Processing Unit
CRBN	Counter for Requested Beacon Number
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DQRAP	Distributde Queueing Random Access Protocol
DSSS	Direct Sequence Spread Spectrum

DSR	Dynamic Source Routing
DTQ	Data Transmission Queue
EAP	Exclsive Access Period
ECG	Electrocardiogram
ED	Energy Detection
ED	Emergency Department
EEG	Electroencephalogram
EMG	Electromyogram
FCS	Frame Check Sequence
FCFS	First Come First Serve
FDMA	Frequency Division Multiple Access
FFD	Full Function Device
FHSS	Frequency Hopping Spread Spectrum
FIFS	First In First Served
GPRS	General Packet Radio Service
GSR	Galvanic Skin Response
GTS	Guaranteed Time Slot
HBC	Human Body Communication
ID	Identity
IEEE	Institute of Electrical and Electronics Engineers
IFS	Inter Frame Space
IRB	Institutional Review Board
ISM	Industrial Scientific Medical
LAN	Local Area Network
LCD	Liquid Cristal Display
LEACH	Long Energy Adaptive Clustering Hierarchy
LED	Light-Emitting Diode
LLC	Logical Link Control
LQI	Link Quality Indicator
MA	Massachusetts
MAC	Medium Access Control
MACA	Multiple Access with Collision Avoidance

MACAW	Multiple Access with Collision Avoidance for Wireless
MAP	Managed Access Period
MCPS	MAC Common Part Sublayer
MICS	Medical Implant Communication Service
MLME	MAC (Sub)Layer Management Entity
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NAV	Network Allocator Vector
NB	Number of Backoff Periods
NB	Narrowband
NCD	Non-Communicable Diseases
NICTA	National Information and Communication Technology Australia
OQPSK	Offset Quadrature Phase-shift keying
OS	Operating System
OTW	Oppurtune Transmission Window
PAN	Personal Area Network
PC	Personal Computer
PCR	(Predicted) Peak Cell Rate
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PHY	Physical
PHR	Physical Header
PIB	PAN Information Base
PKT	Packet
PPDU	PHY Protocol Data Unit
PPG	Photoplethysmogram
QOS	Quality of Service
RAP	Random Access Period
RBI	Recommended Beacon Interval
RBN	Recommended Beacon Number
REQ	Request
RFD	Reduced Function Device

RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
RTS	Request To Send
SAP	Service Access Point
SAR	Specific Absorption Rate
SD	Secure Digital
SDMA	Space Division Multiple Access
SHIMMER	Sensing Health with Intelligence, Modularity, Mobility, and Experimental Reusability
SHR	Synchronization Header
SMART	Scalable Medical Alert Response Technology
S-MAC	Sensor MAC
SN	Sensor Node
SNR	Signal to Noise Ratio
SO	Super frame Order
SPO2	Peripheral capillary oxygen saturation (It is an estimation of the oxygen saturation level)
SSCS	Service-Specific Convergence Sublayer
STD	Standard
SYNC	Synchronization
TCL	Tool Command Language
TDMA	Time Division Multiple Access
T-MAC	Timeout MAC
UART	Universal Asynchronous Receiver/Transmitter
UPC	Usage Parameter Control
USB	Universal Serial Bus
UWB	Ultra Wide Band
WBAN	Wireless Body Area Network
WBSN	Wireless Body Sensor Network
WFT	Wakeup Fallback Time
WMTS	Wireless Medical Telemetry Service
WSN	Wireless Sensor Network

CERTIFICATE OF AUTHORSHIP

I hereby declare that this submission is my own work and to the best of my knowledge and belief, understand that it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at Charles Sturt University or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues with whom I have worked at Charles Sturt University or elsewhere during my candidature is fully acknowledged. I agree that this thesis be accessible for the purpose of study and research in accordance with normal conditions established by the Executive Director, Library Services, Charles Sturt University or nominee, for the care, loan and reproduction of thesis, subject to confidentiality provisions as approved by the University.

Name: Nesae Mouzehkesh Pir Borj

Signature:

Date:

1. CHAPTER 1: INTRODUCTION

This chapter gives an overall view and brief introduction into different concepts related to wireless body area networks (WBAN). It touches on the basic structure of these networks and possible wireless technologies that may be involved with the practical deployment of such technology. With the main focus of the thesis being on healthcare applications, this chapter investigates some of the well-known and up-to-date medical deployments of WBAN in today's hospital environments. The real challenges of implementing a WBAN for vital sign monitoring are elucidated through practical examples in Section 1.3, which encourages the real test bed experiments as a step forward in academic research to more clearly understand and encounter the underlaying challenges. In Section 1.4 we explore some of the main characteristics of WBAN that mostly distinguish them from traditional wireless sensor networks (WSN). The chapter introduces the layers of the IEEE 802.15.4 standard ("IEEE, Std. 802.15.4", 2003) protocol stack, which defines the specifications of the physical (PHY) and medium access control (MAC) protocols of the communication stack. This thesis will contribute to a reliability-guaranteed MAC protocol specifically designed for WBAN and therefore shifts its main focus to the tasks and procedures associated with the MAC layer as described extensively in Sections 1.5, 1.6, and 1.7. The proposed MAC protocol is implemented on top of the IEEE 802.15.4 standard. The chapter finishes by reviewing different classifications of MAC protocols such as contention-based and schedule-based MAC protocols in Sections 1.8 and 1.9.

1.1 Wireless Body Area Networks (WBAN) Structure

It has been more than a decade since the term WBAN was first mentioned in a research paper (Van Dam, Pitchers, & Bernard, 2001). These networks were also known as *patient Personal Area Networks* (*pPAN*) earlier in 2000 in some research papers (Bauer, Sichitiu, Istepanian, & Premaratne, 2000; Jovanov, Price, Raskovic, Kavi, Martin, & Adhami, 2000), and sometimes known as Wireless Biomedical Sensor Networks (WBSN) (Xijun, Meng, & Hongliang, 2005). They are composed of a small number of sensor nodes, implanted in or on the body, to monitor desired vital signs as a means to both expedite and ease ambulatory, long-term, and continuous vital sign monitoring. A basic structure of such a network is shown in Figure 1-1.

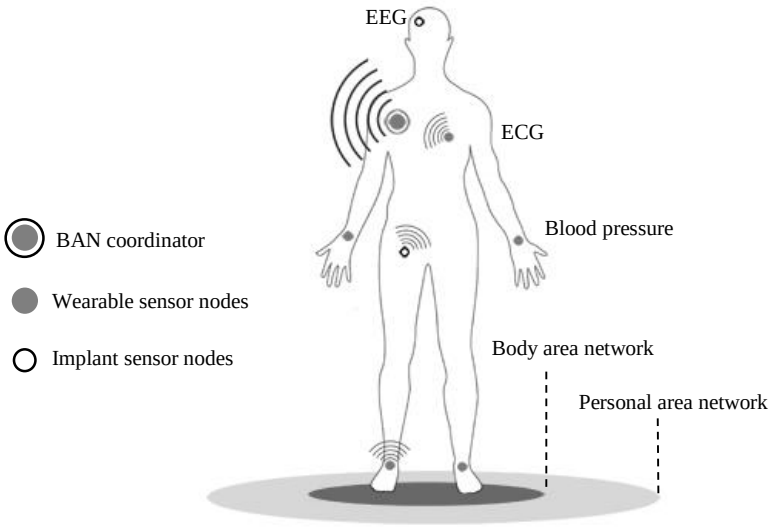


Figure 1-1: Wireless Body Area Network Basic Structure

A wireless personal area network (PAN) is the closest peer to WBAN conforming to the specifications of the same standard, IEEE 802.15.4 ("IEEE, Std. 802.15.4", 2003). One inherent advantage of WBANs over the personal area networks is the number of nodes composing such networks, which eases their implementation but also imposes some unique challenges for the environment: the human body. Usually numbering less than 20 (Chipara, Lu, Bailey, & Roman, 2009; Ko et al., 2008; Curtis et al, 2008; Culmaraes, Ferrelra, & Cabral, 2008) and a centralized star topology adapted yields to less end-to-end delay, which is inevitably higher in larger-scale wireless personal area networks especially in a decentralized manner (non-beacon enabled mode). The size of the network in the reviewed literature is between five and twenty nodes practically, whilst Isikman, Cazalon, Chen, & Li, (2011) compares the size of a typical body area network (BAN) to be modest and less than 256 nodes as compared to a huge scaled WSN. It is obvious that no real implemented WBAN can have as many as 256 nodes on a body, especially in medical applications where comfort and wearability also play an important role, but if the wearable sensors are small enough, a larger number can be used for motion monitoring in sport, entertainment, and, very recently, in movie making industries. The number of nodes can be as big as 256 only in multi-hop implementations such as the ones described in Ko et al., 2008 and Bonnici, Orphanidou, Vallance, & Darrell, 2012, where most of the nodes act as relay nodes between their coordinator and a base station.

Three types of biomedical sensors are reported in the literature, each devoted to a certain class of disability or disease. They are commonly known as physiological, biokinetic, and ambient sensors. *Physiological* sensors are for sensing phenomena such as blood pressure, glucose, electrocardiogram (ECG) for monitoring heart activity, and electroencephalography (EEG) for monitoring brain activity. *Biokinetic* sensors are used for motion-related diseases and monitor the movement of body limbs. A*mbient* sensors are designed to sense environmental signs such as temperature, light, sound, or humidity (Hanson et al., 2009). The wireless sensor platform is equipped with several units, each performing a certain task such as sensing, processing, transmitting, and storing data, working in parallel with an energy unit called the battery and a storage unit. Different biological sensors can be attached to this wireless sensor platform via customized daughter boards that are able to do some basic signal processing tasks and data sampling (Milenkovic, Otto, & Jovanov, 2006).

In a basic structure as shown in Figure 1-2 (Hussain & Kwak, 2009), the biological sensors on the body communicate with a personal server that is richer in resources and considered to be a Full Function Device (FFD), later discussed in this chapter. A mobile phone or a PDA plays the role of the personal server (coordinator) and is assumed to be carried by the patient at all times, or at least located in the same room as the patient. In some research papers it is also known as the *hub* or the *sink* (Tselishchev, Libman, & Boulis, 2011). In Isikman et al. (2011) the term Body Central Unit (BCU) is used for the coordinator and it is sometimes referred to as the *lead-sensor* or *fusion sensor*, especially in wireless sensor network implementations that acts as a low-end data gathering sensor node (Ren &Liang, 2006). In Lamprinos, Prentza, Sakka, & Koutsouris (2005), a coordinator is also known as the *supervising node* and acts as the bridging node to the remote monitoring facility. In some research works such as Lorincz et al. (2009), when more complex controlling tasks need to be performed by the personal server, a base station or laptop is considered as the coordinator and many of the decision-making tasks will be dealt with by its processor, which triggers some simple modules and components running on the resource-limited sensor nodes on request. This centralized structure is inherent to the body area networks as opposed to wireless sensor networks, which have a distributed and co-operative nature.

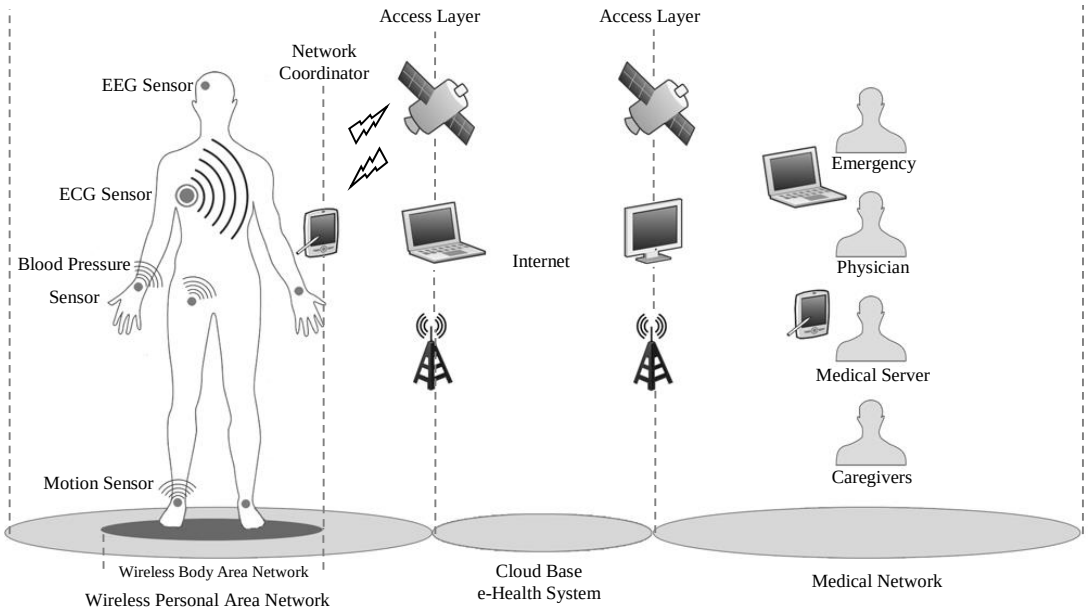


Figure 1-2: WBAN: A Comprehensive Structure

There are other formations of body area networks as reported in Yuce (2009, p. 119), where multi-patient monitoring is enabled by placing the coordinator in the same room where several patients are. The coordinator in this scenario is actually shared among several wireless body area networks. The information gathered by this single coordinator is relayed to a local base station such as a laptop, which is connected to a local area network (LAN) and ultimately the Internet. There is also the possibility of having a wireless network of several coordinators located in different rooms of a hospital where all the data is aggregated by a central coordinator in that hospital and later relayed to remote base stations. The latter examples of other possible structures of a wireless body area network encourage the flexibility and scalability of such networks to different hospital needs, based on different patient types (in terms of their disabilities). The advantage of having a network of coordinators is also a remedy to the problem of a single point of failure where there is only one coordinator as the data aggregation point. Some researchers have also considered different subnets of on-body and in-body sensor nodes for each of which there is a parent node that communicates with the main coordinator (Lee, Lee, & Choi, 2010). Some research papers such as Isikman et al. (2011) have also considered an individual and autonomous WBAN in which no command comes from the coordinator based on the data processing of the received data through other longer-range networks. In such a structure, the coordinator is supposed to be smart enough to make a local decision based on its own data processing logics and how it has been programmed. So no extra WBAN

communication is considered between the coordinator and other wide band networks such as GSM and eventually the internet. A more comprehensive structure of a network and its intermediate networks is shown in Figure 1-2, where the data gathered on the coordinator device is able to be sent over the cloud to any remote area where a physical practitioner possibly is. A network of different care givers could also co-operate to monitor the same patient in a medical network either in the same location or in different locations. In Ghaboosi, Pahlaban, and Pomalaza-Raez, (2011), a multi-hop structure of WBAN has been implemented where a few WBANs co-exist in the same vicinity where the network connectivity is provided for the sensors of different WBANs. The data from each sensor in any WBAN can be sent directly to the sink (coordinator) of the other WBAN.

1.2 **Potentials and Possibilities in Healthcare**

In healthcare, WBAN can be applied to a wide range of illnesses, both longitudinal and short term, with different requirements to ensure a timely and reliable data transfer from where the patient is located to where the caregivers are. However, the application range for WBAN is not confined to healthcare and can be applied to other types of emergency response and even athletic performance evaluation (Gao et al., 2008; Bachlin, Forster, & Troster, 2009). The broad deployment of such networks in healthcare is mostly motivated by decreasing hospital capacities and staff shortages, especially in mass casualty events during disasters (Ko et al., 2008), and a massive growth in the elderly population. Latré, Braem, Moerman, Blondia, & Demeester (2011) consider cardiovascular diseases and diabetes to be two of the major causes of death, and they are rising each year. Bloom, Boersch-Supan, McGee, & Seike (2011) consider non-communicable diseases (NCDs) such as cardiovascular disease, cancer, diabetes, and chronic respiratory disease to be the leading causes of death for which the care costs are reported to be relatively higher than many other diseases. Apart from the two main diseases that lead to death mentioned above, age-associated illnesses such as Alzheimer's disease are escalating as well. The elderly population is growing year by year and therefore the number of people living with chronic diseases is increasing (Omeni, Toumaz Technol. Ltd., Didcot, Eljamaly, & Burdett, 2007). Dishong & McGrowth (2010, p. 3) reported that 861 million people around the world are living with chronic diseases, which imposes a high level of expenditure on governments to provide healthcare facilities for this group of people. It is even expected that the old population will outnumber the young population in less than ten years (Cohen, 2003). The growing population of age 65+ is briefly analyzed in Dishong &

McGrowth (2010, p. 4), and is likely to reach its highest peak by 2025 in both the US and Europe. An increase of life expectancy of two decades between 1950 and 2005 has been reported (Bloom, et al., 2011). Declining fertility is also one of the implied reasons for the increase in the elderly share of the population and is predicted to decrease in 2050 by half of the percentage back in 1950. Whilst the elderly population of over 65 yearolds only represents 11% of the current population (reported in 2011), it is expected to soar to about 22% of the total population by 2050 (Bloom, et al., 2011). An implicit fact about the growing elderly population is their ability to work productively into much later age than currently (Bloom, et al., 2011); this would leave the population with less time in a day to allocate to their healthcare needs such as visiting a doctor or doing an age-related necessary test. Such a rapid growth in the elderly population imposes new challenges to the world of traditional healthcare systems. An increase in the cost and demand of the workforce in hospitals stands as a direct result of such a growth, which declines in-hospital care systems as a perpetual solution in near future. A shift towards a reliable, caring system where subjects can have easy and fast access to their general practitioners in case of urgencywill become a necessity in the coming decades.

Prevention and early detection of abnormality in any vital sign has been an absolute remedy in treating many threatening health conditions, especially in cases of chronic but dangerous diseases. It is not always easy to anticipate a critical condition since, as reported in Hillman et al. (2001), at least 50% of patients have abnormalities in their vital signs only a few hours before their death. This factor has been mentioned as a lack of “clinical deterioration” in the traditional hospital care system in Chipara, et al. (2009), who emphasizes early detection as a necessary tool in assessing patients with instantaneous critical conditions. This places an urgent need for a system to provide a means of remote monitoring with instantaneous feedback to the practitioner that has the capability of providing a seamless connectivity between the subject and its caregiver for as long as needed.

Although the healthcare sector seems to have become increasingly keen in recent years to act upon remote healthcare delivery (Omeni et al., 2007), there are many concerns arising from such an infrastructure.Certain paramount factors must be kept in any variable situation the patient’s body may be exposed to that may restrain WBANs from wide acceptance. A number of these constraints are discussed in Sections 1.4.1–1.4.8, among which are interference from other wide radio bands, different mobility and channel models, a need for unobtrusive monitoring, and application diversity as a few sources causing unsustainable conditions for

having a smooth and reliable communication. In Hanson et al. (2009), a host of factors to be fulfilled in body area networks for a global adoption of smart healthcare solutions is given amongst which are safety, security, privacy, compatibility, and ease of use. In Bonnici et al. (2012), a practical study has been conducted on real subjects to investigate the real causes for sensor failure when recording data from patients monitored in actual hospital environments. Seamless reliable communication links between the sensors and their coordinator become the most paramount contributor to wide acceptance in clinical use cases. Bonnici et al. (2012) declare a few widely used sensors in non-clinical environments as a result of requiring less accuracy and reliability in such environments as opposed to clinical deployments. The authors state that "providing a distinct advantage with respect to the existing monitoring devices and also meeting the hospital standards" remains the main hindrance in WBAN extensive approval in medical applications. At least one possible factor leading to such a result in their experiments would be using sensors of different types (which are anonymized in the paper) so the general conclusion might have been affected by having less reliable sensors of one specific type. The main reasons to blame have been identified as battery-related issues, transmission problems, poor ergonomics, and non-intuitive software installed on smart phones. In Chipara, et al. (2009), however, sensing outage related to disconnections of sensors was identified as the main reason behind unreliability but was not considered as a problem at low data rates. A cut down on the price by WBAN remote monitoring compared to the current traditional methods is also one of the reasons declared in Isikman et al. (2011) and will motivate a "mass adoption" of such technology more in the future. They also consider easy implementation of the technology to be a great factor in their wide acceptance. A more comfortable design of such networks as opposed to the traditional wired ones would also be a motivating factor in their broad adoption (Lamprinos et al., 2005). Besides great advances made so far towards remote monitoring applications in terms of both research at an academic level and prototyping it in real clinical environments, the technology is still considered premature for more crucial and critical tasks such as acute health care in hospital emergency rooms. Section 1.3 below discusses how some successful deployments of a WBAN such as Curtis et al. (2008) have only been able to obtain institutional review board (IRB) (IRB, Institutional Review Board Services, n. d.) full approval for testing their system in emergency department waiting rooms and not the emergency room itself. As mentioned in their pilot study, the IRB approval was conditional to an individual with Advanced Cardiovascular Life Support (ACLS) training to monitor the SMART's modules' centre at all times, which prevented their pilot study from reflecting the actual behavior of caregivers to the generated alarms of their system. IRB demanded human intervention in their

system as it was the sole alarm generator in an emergency room waiting area to the caregiver and thus expected to be as efficient as possible.

1.3 From Concept to Delivery Challenges

Besides all the practically proven facts and possible failures mentioned in Section 1.2, there has been a successful thread of practical deployments in this area that has opened a new horizon in remote patient monitoring and its great capabilities in reducing the costs associated with traditional hospital care. In Chipara, et al. (2009), for instance, an in-depth clinical experiment was conducted to assess the feasibility of wireless sensors for patient monitoring in a real hospital environment. Although it was a successful deployment, the results presented in their recent work, Chipara et al. (2010), clash with the reports later reviewed in Bonnici et al. (2012) in this section, which is a clinical experiment on a smaller scale. This implies that not every practical deployment of a WBAN is successful and many factors could be contributing to a decent system performance. As an example, while Chipara et al. (2010) promise the feasibility of using WBAN as a future resolution to continuous patient monitoring, Bonnici et al. (2012) conclude it the other way. One major drawback of the system in Chipara et al. (2009) was that there was no seamless connection between the sensors on the patient's body and its base station once out of the radio range, which was reflected in their experiments. The problems faced within a practical context such as a real test-bed development signify the importance of current pilot studies that have contributed to a better perception of the real challenges ahead for WBANs. The focus of this section is to review some of the well-known and related implementations of a WBAN in different hospital environments to become more familiar with the real implementations' issues encountered. The investigated works here entail the importance of test-bed experiments (if possible), or at least some real environmental measurements to be coupled with any simulation work. This section summarizes some of the currently implemented WBAN projects in the world, which are examined works through different trials and experimentations.The main purpose of this section is to draw some of the real challenges of practical set ups that cannot possibly be simulated but their awareness plays a crucial role in a successful implementation.

In a recent case study by the SHIMMER research group (SHIMMER Sensing Technology, 2014), a Tele-Epilepsy and Remote Seizure Monitoring using SHIMMER sensor platforms (Culmaraes et al., 2008) was experimented by a research team in the Netherlands in collaboration with UMC Utrecht, Kempenhaeghe-Heeze, and SEIN-Zwolle. A small WBAN

of only two sensor nodes was developed that used only ECG and 3D accelerometer data from the two sensors placed on the left and right arms of the subject to generate alarms in the event of a major nocturnal epileptic seizure. The seizure is detectable whenever a certain threshold is passed from the data gathered on a PC from the two sensor nodes on the body. The CE Certification of the SHIMMER device has been stated as an advantage, as it shows the eligibility of such devices to be used in real life experiments with the human body. They tested their remote monitoring system on 50 patients in four different centers and contrasted the results of their monitoring against the traditional gold-standard for EEG-video monitoring for epilepsy patients, and showing an efficiency of 90% compared to the gold-standard system. The Pediatrics Epilepsy Remote Monitoring System founded by Vodafone Portugal was a project initiated in 2007in Portugal in agreement with the West Lisbon Hospital Centre (CHLO), which proposed the establishment of a remote monitoring system for patients suffering from epilepsy. More than 50,000 cases of epilepsy patients are reported in Portugal, with 2,500–5,000 of them having high chances of recovery if they undergo surgery (Culmaraes et al., 2008). The availability of monitoring, though, is a great factor in increasing the number of potentially successful surgeries carried out on certain candidates.With current and traditional in-hospital monitoring methods; the patients are often kept on surgery waiting lists for a very long time. Apart from the social and personal costs reported, the current examination tools such as the EEG electrodes and the monitoring camera require the neurologist to be present with the patient precisely when the seizure happens. Constant monitoring of the patient and risk management tasks are necessary before a decision can be made about having a surgery that cannot be done efficiently and adequately with traditional monitoring systems. Traditionally, the video information (via the camera in the room) and the EEG signals that are gathered by connecting EEG electrodes connected to the patient's head by means of small cables are received at the PC where they need to be analyzed the moment a seizure happens. The information on when and where in the brain the seizure happens helps the doctors make the best decision on whether or not surgery could actually help. However, it often takes a long time this way for the doctors to collect as much knowledge as needed about a particular patient which would, in most cases, put the patient at risk of more seizures. In March 2008, the Centro Hospitalar de Lisbon Ocidental and Portugal's Vodafone Foundation tested their remote epilepsy monitoring system.The test used a wireless camera and wireless EEG electrodes connected to the patient's brain. The information could then be accessed anywhere in the world by the doctors to analyze the seizure moments by aid of an alert mechanism allowing them to visualize the EEG signals on their PDAs or smart phones. It has since led to a faster, safer, and

more efficient diagnosis of the cause and has made the process of decision-making for having a surgery a lot quicker and requires the patients to stay in hospital much less than before. An increase of 70% in the number of examinations was reported in 2008 as compared to 2007.

Chipara et al. (2009) is an empirical in-depth study of the feasibility of a wireless clinical monitoring system. The challenges associated with implementing a real wireless monitoring system in general hospital units of Barnes-Jewish Hospital St. Louis in 2009 were investigated during a month-long trial through real reliability measurements. The sensory readings were limited to pulse and oxygen saturation data gathered by the sensors on the body. Dividing the reliability measurements into sensing and networking reliability would distinguish the loss of data due to the sensors' disconnection from the losses associated with transmission-related problems over the wireless channel. It has been represented that only very small networking failures would affect the fidelity of the received data and the rest would be disconnection-induced problems. Similar implementations seem to be a promising approach to apply in future healthcare infrastructures. Their implementation is based on the IEEE 802.15.4 standard, which will be discussed in Section 1.6 in more detail, and is the standard used in this thesis as well. However an 802.11 link has been used for backing up the data from the base station on a local database. The data delivery of the sensor nodes to the base station is a multi hop, which contrasts with the star topology we have used in our experimental (and simulation) set ups. Realistic factors such as the frequency of intervention by nurses or the detection of clinical deterioration, which cannot be measured by mere simulations, have been investigated to learn about the challenges of a real implementation. There was evidence of different unsatisfactory results that had to eventually be excluded from the results for reliability observations. The two main contributors to such malfunctions were the older version of the routing protocol, which had not been upgraded at the time of the experiments, and improperly handled exceptions in the data-collection code at the base station end. Generally speaking, such problems would almost never be encountered in a pure simulation environment, which is evident by comparing the achieved sensing reliability of 80% to the network reliability of 99.9%. Such an increase was a direct effect of the distinguishment between the sensing and networking reliabilities where the issues associated with sensors' disconnections were addressed separately. There is no way of simulating the sensing reliability in any of the current simulators, as only the network functioning components have been implemented, which would only encompass the network reliability if configured well. It is stated in Chipara et al. (2009) that the achieved sensing reliability of 80% is sufficient for health practitioners with the data they need about pulse and

oxygenation, and it even prevails the accuracy of data gathered by traditional manual collection methods. It is obvious that the overall reliability is dominated by the sensing reliability as the main source of causing deterioration to overall system reliability, mainly caused by only a few disconnection incidents, and is emphasized to be addressed in the future of WBAN for healthcare applications. The small fraction of lost packets representing network reliability, which would only happen infrequently and has a very quick recovery time, promises reliable and widely- accepted WBAN implementations in hospital units in the future. A trial period of three days of non-stop monitoring by TelosB sensor platforms was carried out before recharging the batteries.

MEDiSN (Ko et al., 2008) is another empirical study based on a multi-hop structure that has been prototyped at the trauma center of the University of Maryland Medical Center and the Johns Hopkins Hospital Emergency Department. The relay nodes between the physiological sensor node and its coordinator are deployed statically to provide better functionality. The difference between a relay node and a physiological sensor node in their implementation is duty cycling the radio, which is only available for the physiological sensors. Their results show a good resemblance between the experimental test-bed evaluations and simulation evaluations. The authors take advantage of the multi-hop nature of their network to improve the reliability of their test-bed in the high interference-prone environment of a hospital. The two main enhancements performed in their work are at the network layer and in the CTP routing protocol of TinyOS as they carry out their experiments on Tmote Mini sensors by Texas Instruments. In their deployments of a WBAN for the University of Maryland Shock Trauma Center, they were able to cover an area of 10,000 square feet. The placement of the relay sensors was based on an evaluation of the physiological sensors' communication range coverage. Basically, a relay sensor was added whenever a physiological sensor would fall out of the communication range of the previously deployed one. There were eight physiological sensor nodes placed on each patient's body for the experiments to monitor vital signs such as blood oxygen levels and pulse rate. The duration of the experiments for each patient was not more than a few hours as the case study is only addressing the duration of time from when a patient is sent to the operating room, throughout the operation, and after they have been transferred to a post-anesthesia care unit. An average reception ratio of 98.25% was reported for this experiment but it is not clear how many patients participated. In a second experiment carried out in the John Hopkins Hospital Emergency Room, 46 patients were monitored for a longer period of time (over 3 hours) over a 6,500 square feet area of the emergency room waiting section. An

average reception rate of 95.43% was reported for this case study. Since the two experiments reported in this paper were based on a multi-hop structure, the challenges reported were mainly finding the best spots to attach and secure the relay sensors, which in many cases was not an easy task due to obstacles such as thick steel doors, furniture, glass walls, etc. To alleviate the interference problem due to the availability of a hospital-wide WiFi connection, their team identified the channels being used by the hospital, which were channels 1, 6, and 11 of the IEEE 802.15.4 standard, and therefore deployed their implementation on channel 26 of the 2.4 GHz frequency band of IEEE 802.15.4. Although this channel at times also had heavy interference from nearby cordless devices such as mobile phones, the final results revealed that the overall performance of the system, based on the adequate number of relay points employed, was maintained at a very good level.

Scalable Medical Alert Response Technology (SMART) (Curtis et al., 2008) is a prototype implementation of WBAN which was piloted in the waiting area of an emergency department and evaluated with 145 post-triage patients in Brigham and Women's Hospital's Emergency Department, Boston, MA. Although the SMART project has been mainly tested with healthy individuals and as a pilot study for its disaster drill, it has summarized useful facts about the real problems faced during the experiments. The authors assessed the feasibility of their system by a mass casualty drill exercise to measure its reliability in a more realistic way. Like MEDiSN, their implementation is also a multi-hop one but is uniquely designed for mass casualty situations and addressing the scalability of a preexisting WBAN in the emergency unit of a hospital to other units when there is a sudden increase in the number of patients. The ECG and SpO_2 of each patient are monitored by SMART, which provides information to caregivers, based on which they are then able to categorize triage levels. SMART sends out the gathered data in real time to the base station where a caregiver is supposed to act upon specific alerts that are generated based on some abnormalities. The availability of the location information for both the patients and the caregivers is one advantage that differentiates SMART from its predecessors of the same technology. The data from both caregiver and patients must be sent to a central unit (a PC) for processing and make them available for other modules. The use of a smart device such as a PDA is necessitated for both the patient and the caregiver, assuming that the PDA devices are with them at all times during the trial process. Therefore, there are two different user interface modules in SMART, one displayed at the caregiver's PDA and the other at the patient's PDA. The two PDAs of patient and caregiver are connected to each other wirelessly through IEEE 802.11b. Their implementation consists of ten patients and two

caregivers. The system deployed on the PC (or laptop) as the SMART Central has two main operating parts, which together analyze the received real time data from the patients and trigger specific alarms by which relevant information will be dispatched to selected caregivers. New, higher-level data from the raw and waveform data is derived by the SMART Central on the PC and the alarms it generates are based on the evaluation of SpO_2 and heart beat data, which is done by a rule set. The main rationale of the rule set that indicates the capability of their system for mass casualty situations is to send the generated alarm to the next nearest caregiver in the hospital environment. Sometimes, in reality, an abnormality in a vital sign such as atrial fibrillation causes the alarm system to alert continuously, a situation that had to be resolved by giving authority to the caregiver to disable it at any time. Once a caregiver receives the alarm, all the related information of the corresponding patient will be displayed on his or her PDA screen such as vital signs, location, and also a history of previously generated alarms to aid the caregiver to respond efficiently. Another problem associated with the alarm system of the SMART project was excessive generation of false positives or false alarms, which proved to be a distraction to the personnel of the emergency department. The SMART project pilot study was launched in 2006 and was carried out for one year to end in 2007. The technology used then was not as comfortable as the recent ones such as SHIMMER platforms and needed a waist pack to be carried by the patient, from which the ECG and SpO_2 electrodes were attached to the patient by means of small wires, and therefore it was not an absolutely wireless implementation. One hundred and fifty healthy individuals participated in a disaster drill conducted by the SMART group researchers. Patients with life-threatening symptoms were given a pack of sensors as described above and their data was entered into the SMART Central system on the laptop. In reality, the scalability performance of the system could not be well tested as the number of patients never exceeded a threshold, and so a disaster drill was found to be necessary. On the other hand, the system utilization by the emergency room personnel could not be evaluated, as one of the IRB requirements was full monitoring of an individual over the SMART's modules. As mentioned above, IRB approval for the SMART pilot study was conditional on an individual with ACLS training monitoring the SMART's modules'center at all times. Plus the implementation of SMART in the waiting room of an ED of a hospital raised the expectation of a guaranteed efficiency of the system's feedback in terms of detecting abnormalities in the patients being monitored by the hospital authorities, which could even lead to a lawsuit. Such great responsibilities when deploying a WBAN have already been discussed in Section 1.2 as one of the barriers to globally approve such emerging technology in all hospital environments in the future.

Unlike some of the works summarized in this section that encourage the implementation of WBANs in near-future hospital care, there are some other works such as Bonnici et al. (2012) that demonstrate the inefficiency of the current sensor platforms and list the challenges that need to be addressed. Bonnici and colleagues conducted a clinical trial in John Radcliffe Hospital in Oxford that would dispute the reliability of WBAN in medical care units, discussing their main inefficiencies. Thirty-one patients were continuously monitored for 24 hours for their electrocardiogram (ECG) and photoplethysmogram (PPG) signs. Through exploration of some of the mostly neglected issues in many non-realistic and academically evaluated WBANs, their clinical trial uncovered the importance of a few onsite changes that would make great contributions to the delivery of WBANs in real hospital environments. Problems such as "'alarm fatigue", which is barely mentioned in any research paper, and the proposed alternate solutions have encouraged real experimental works as a means to elucidate further the pros and cons of such technology. Although the use of Bluetooth technology has been criticized as a good choice for medical applications due to its high energy consumption, their clinical trial enabled three of the four present sensor platforms in their test to be connected to the hospital WiFi that was available hospital-wide through Bluetooth. Different anonymized sensor platforms were used to test the reliability of four of the current off-the-shelf medical sensor platforms, so it is beyond our ability to further discuss the specific configurations or differences in hardware that might have led to some disappointing results in terms of reliability. A reliability of only 13% has been reported in Bonnici et al. (2012), which is associated with one sensor type, meaning that while a particular sensor has been performing at its highest level, the entire system reliability was let down by transmission, battery-related, and non-intuitive software problems of another sensor type. We saw earlier in this section how reliability is divided into two different categories of sensing and networking reliability (Chipara et al., 2010) to ease both differentiating and addressing the issues related to either transmission of packets over the channel or sensor platform-related problems. Sensing and networking reliability are discussed implicitly in this work (Bonnici et al., 2012) by investigating the packet losses caused by both sensors' battery malfunction and also those related to sending the stored information on the sensors' SD cards to the base station. Sudden and extensive packet loss can take place when certain device adjustment settings have not been met according to those suggested by its manufacturer, which would eventually lead to a massive reduction in their expected maximum battery lifetime. Other losses could be attributed to the method of saving and sending the sensed data to the base station used by the sensors. Sensors set to first save the data on their SD cards

and then send them in batches to the base station are in great danger of losing those data due to the amount of traffic in the hospital's network.

The challenges cited in the above explored works varied from connection-induced problems between the sensor and its coordinator caused by either coverage or wireless channel problems, to finding the optimum spots to place the relay nodes in a multi-hop structure, interference from other devices, and even the high expectations of the hospital's staff in regard to the implemented WBAN and its performance. These real challenges punctuate the significance of test-bed deployments, even though preliminary and basic, as a requirement to lead the academic research works to reality implementation more in the future. Chapter 6 below is dedicated to some performance evaluations of our proposed MAC protocol, which has already been tested through simulations in Chapter 5. The real challenges we met in implementing the same concept (as in simulations) into the selected sensor platforms are discussed in Section 5.2.2, and actually obliged us to make slight changes to the proposed method to be able to efficiently implement the idea from a simulation world to the real experimental world of sensor platforms.

Although the main concentration of this section was on empirical works normally in a hospital environment and specifically for healthcare, there exist threads of other wearable technologies that enclose applications in sport, entertainment, and industry which would instantiate similar challenges. Some of the most well-known ones are: Korpinen, Rakkola, & Ramo (2007) and Gyselinckx,Van Hoof, Ryckaert, &Yazicioglu (2005).

1.4 **WSN vs. WBAN**

Wireless Body Area Networks differ from traditional Wireless Sensor Networks (WSN) in a few senses that we have tried to classify in Sections 1.4.1–1.4.8. A thorough study of these differences is important for a better understanding of the QoS requirements in such networks and also fulfilling those requirements through the layers of the communication stack. Other than differences such as the number of nodes, which is less in WBANs compared to WSNs, and the body attenuation and shadowing effects, there are other factors, as highlighted in Sections 1.4.1–1.4.8, that can have a great impact on implementation in either a positive or negative way. Interference, for instance, is one of the main issues to be addressed, as well as reliability, which has greater significance when it comes to medical data. Security of the transferred data and its confidentiality is another concern, but is not investigated in this thesis as it remains outside the scope of the study. Some differences such as resource asymmetry,

which arises by mostly selected star topology in WBAN implementations, can actually be exploited for a more efficient protocol behavior as also described in Section 1.4.3. The top eight main differences from the reviewed literature are presented as follows.

1.4.1 **Wearability**

There is a need for sensor nodes in a WBAN to be comfortable when fixed on a human body (O'Donovan, Sreenan, Sammon, O'Reilly, & O'Connor, 2009).There is strong emphasis for such entities to be smaller, wearable, and unobtrusive once deployed on the patient's body (Yuce, 2009, p. 119). Therefore many technical issues are still to be resolved to make the sensor nodes as convenient as needed for healthcare applications (Lee & Chung, 2009). Observations made during empirical work by Bonnici et al. (2012) reported that 19% of the patients under study in a hospital unit took off their sensors due to discomfort before the experiments finished. Wearability, also termed "testing comfort" (Bonnici et al., 2012), is evaluated based on different criteria such as the patient's physical condition, their mood, their perception of the device, and also the duration they have to wear it. "Suitable sensors"is another term used (Isikman et al, 2011) in referring to nodes that are small and do not hinder the mobility of the patients and make monitoring as transparent to the patient as possible. Power efficiency can also play a role in the level of wearability of the medical sensors by allowing a smarter design leading to smaller and more wearable sensor platforms. In Burrati et al. (2011), power efficiency is considered to affect the size of the embedded battery greatly, especially in the case of an implantable medical sensor where comfort of the patient is one of the most crucial factors to be looked after. A true example of how wearability would impact the future of WBAN in healthcare is the statistics given in Chipara et al. (2009). One in six patients would not approve of only two medical sensors being deployed on their body. It would cause them difficulty in doing their every day routine. While wearability is more referred to as the level of comfort presented by the medical sensors, it also describes the ease of deployment by the clinical staff when placing them on a patient's body. In Ko et al. (2008), this is especially emphasized for mass casualty events when the demand for medical care has a sudden surge and medical personnel must be as quick as possible.

1.4.2 **Channel Model**

The main difference in the channel model of a WBAN comes from its environment, the human body. It becomes more prominent when the physiological sensor nodes are implanted beneath the skin, but it also affects the signal of those deployed externally and attached to the limbs of

the body. The received signal can be altered as a result of the channel characteristics it has travelled through, which, in our simulation and experiments, is both air and body tissue. The transmitted signal is affected mainly by three factors known as path loss, shadowing, and multipath (Jain, 2007). The main factor contributing to shadowing in a WBAN is the absorption of the transmitted signal by the body limbs' tissue, which introduces the concept of Specific Absorption Rate (SAR). Such an effect becomes more evident when the subject moves and puts the sensors on different spots of the body, sometimes out of a direct line of sight to the coordinator device.As a result of shadowing, a transmitted signal from one specific physiological sensor node on the body may not always be received at the coordinator device with the same power and will vary at all times. Not too much of a multipath phenomenon may happen between the nodes and their coordinator device as the distance between the nodes and the coordinator device is only a meter or so and in many cases even less than a meter. Deep fading and shadowing effects are very common to wireless body area networks, which at times can lead to a connection breakdown of several minutes (Lamahewa et al., 2010). In Zhisheng, Bin, & Chang (2012), deep fading of the wireless channel is reported to be at times as long as 10–400 ms. Different levels of Received Signal Strength Indicator (RSSI) exist in a WBAN both because of body shadowing effects and body movements (Prabh & Haur, 2011). Wireless channel in body area networks is highly variable in time and deep fades make the signal drop below the receiver's sensitivity, which must be taken into consideration especially in a polling-based MAC as the control messages required to achieve polling become vulnerable.

Therefore the use of an appropriate channel model is a key requirement, and for this reason we should consider the variations in the RSS (Boulis & Tselishchev, 2010). In Burrati et al. (2011), extensive measurements were carried out to characterize the time variant channel propagation on the body. Their test bed consisted of four sensors on the body sending out channel impulses to the coordinator node while the patient was moving. A complete model for the on-body channel was then extracted by a statistical description of the fast and slow fading for different scenarios and configurations. There are other research works such as Javaid et al., (2013) that have studied the different channel models when sensors are deployed on or inside the body. The propagation characteristics have been studied in two different categories, the sensors on the body and the sensors implanted beneath the skin on the limbs. A 3D visualization scheme as in Sayrafian-Pour et al. (2010) has been used to provide the possibility of measuring the physical parameters when a sensor node is placed inside the body. Four different communication models are described in the paper to study the path loss calculations in each

case, which are different combinations of a source-to-destination communication between an implanted or on-body sensor node to its coordinator. It is illustrated, through the different models in the paper, how the path loss of a deep implanted sensor node to an on-body sensor is greater than the path loss of an on-body to on-body sensor node. The Castalia simulator research group in NICTA (National ICT Australia Ltd, 2014) claims its implemented simulator to be the most realistic simulator in terms of channel model to fit WSN and WBAN simulations in the academic world (Boulis, 2011, p. 51). Modeling the channel in a WBAN set-up is a difficult task due to the always changing environment (both sensors and the whole body moving in different directions), and there are not as many research studies done at the physical layer, as also declared in Boulis (2011, p. 51), that would offer models to fit the experimental data. For the Castalia simulator, the produced path loss model as well as the temporal variation of the channel are the result of hundreds of thousands of test-bed experiments that are analyzed to create accurate average path loss values between different body limbs of a human subject. Details of the average path loss modeling in the Castalia simulator are presented in its manual (Boulis, 2011, pp. 52–53). In addition to fading effects attributed to the human body, the Clear Channel Assessment (CCA) at the PHY layer must also be realistic (Boulis & Tselishchev, 2010), as a simple CCA is insufficient since it could occur between identical packets (packets of the same message) and would incorrectly report a clear channel (O'Donovan et al., 2009). Replacement of the current Carrier Sense Multiple Access (CSMA)-based scheme in Contention Access Period (CAP) with a mini-slotted ALOHA (Abramson, 1970) has been described in Li et al. (2009) to compensate for the lack of adequate sensitivity in the current CCA mechanism. An additional Clear Channel Assessment for IEEE 802.15.4 CSMA/CA networks is proposed in Ming, Wong, & Hsu (2010) as another solution to the problem. Time-varying fading channels have also been studied in Su & Zhang (2009), where they characterize the variations in channel by SNR. The fading characteristics of WBANs have been studied in depth in Cotton & Scanlon (2009).

1.4.3 **Resource Asymmetry and Star Topology**

What complicates defining a specific standard for wireless body area networks is the asymmetry that exists between sensor nodes and aggregators (Hanson et al., 2009). This asymmetry usually mimics the master–slave nature of cellular networks (Tselishchev et al., 2011) but is always distinguished from cellular networks by low power performance and unique requirements.In cellular networks, a master–slave structure gives the control of a network of devices to a master device that has the role of synchronization and spreading clock

information to its slave nodes. The sensor nodes in a WBAN are often designed to monitor completely different applications running on each node. The data relayed to the aggregator (coordinator) need to be encrypted and received with almost no loss to be as reliable and trustworthy as possible. Several studies have actually tried to exploit this asymmetry so that less energy will be consumed. An example of this in the literature is Lorincz et al. (2009), where the core deriver (controller) is residing on the coordinator to control some features such as downloading, storing, and the amount of data being transmitted on the fly. Star topology is the best topology that goes with such asymmetry between nodes and their aggregator node in terms of resources. Its centralized nature has made it the prominent topology in most of the literature reviewed in this thesis mainly focusing on WBAN structures. In Omeni, Eljamaly, & Burdett (2007), it is described as a master–slave architecture where a central node aggregates the readings of all the other nodes as a master and the topology is not an ad hoc one. Apart from the single point of failure that a star structure has, it represents a simpler MAC protocol approach where an asymmetric QoS design and resource scheduling mechanism would be easily implemented (Zhou, Lu, Wan, Yarvis, & Stankovic, 2008). Almost all the works in the literature have used star as their topology, which eliminates most of the scalability and self organization problems that exist in traditional wireless sensor networks (Omeni et al., 2007; Zhisheng, Y. & Liu, 2011; Li, & Tan, 2005; Li & Tan, 2010; Zhang & Dolmans, 2009; Osterlind et al. 2010; Otal & Alonso, 2009).

1.4.4 **Mobility Model**

Mobility can be introduced to any implemented WBAN the same as traditional WSNs when the environment in which the medical sensors are deployed is mobile. Contrary to most of the WSN set ups, there is a two-dimensional mobility associated with a human's body where the medical sensors will be placed, which imposes new challenges to the designed protocols for a WBAN. The signal strength changes repeatedly due to this mobility and cannot be predictable. Therefore good patient mobility support in the developed WBAN system is an important factor (Chipara et al., 2010). Both "body limbs' mobility" and the "whole body" mobility will be occurring at the same time as the subject moves with sensors fixed on his other limbs. So it should be carefully modeled when simulating and should be well-supported in a real, experimental implementation. A mobility model has been studied by Nabi, Geilen, & Basten (2011), and a mobility support mechanism is proposed in Latré et al. (2007). Other studies such as Prabh & Haur (2011) have also reported significant fluctuations in the received signal strength caused by continuous changes in the relative distance and orientation of the nodes as

a result of human movements. A novel approach has been investigated in Chipara et al. (2010) to handle patient mobility in an empirical study that is a multi-hop set up with relay nodes between the medical sensors and their coordinator. In multi-hop implementations of a WBAN, mobility can greatly affect the routing of the data packets from the physiological sensor node to its coordinator. In Chipara et al. (2010), such an impact is attributed to the communication link between the medical sensor and its immediate relay node. Once the transmitted data has reached the relay node, there will be no expectation of data loss and the data will be transmitted to the coordinator at an optimum reliability. Therefore, the mobility-induced problem in their work is only addressed for the communication link between the sensor node and its immediate relay node. In TinyOS ("TinyOS, an Operating System Designed for Low Power Wireless Devices", 2013), for instance, which is the operating system of most of the available sensor platforms in the market, the subject's mobility would lead to unstable information in the neighbors' table of the collection tree protocol (CTP). This is the routing protocol used by TinyOS and it keeps account of the currently available hops (sensors) within the communication range of the sender. The nearest next hop would be chosen for relaying its data to the next hop but with an unstable channel, CTP would not be able to perform error-free. The mobility-induced problems reported in Prabh & Haur (2011) are summarized as: 1) changes in distance between nodes that influence the path loss and fading; 2) shadowing; and 3) relative node orientation, all of which will lead to different signal strength received at the same distance. In Castalia 3.2, which is the chosen simulator for our simulations in Chapter 4, the path loss for a mobile object has been implementedin the simulator by modeling the space with discrete cells where the value of the path loss is attributed to each cell when the object moves. Therefore, mobility has minimal effect on the outcome of the simulation results in regard to its different path loss model. The smaller the dimension of a cell, the more accurate the path loss will be modeled when the object is moving. However it comes at the cost of more computational overhead. It is stated in Castalia that the choice of the cell size has to be chosen efficiently with respect to the application needs.

1.4.5 **Physical Layer Portability**

The ability to work on different physical layers is also a challenge in WBAN and has been studied in many works. Zhou et al. (2008), for example, provide a virtual MAC layer in order to enable the protocol to operate on different radio technologies such as CC1000, IEEE 802.15.4, and Bluetooth, so that medical sensor devices using different radio platforms will not face a problem. Heterogeneous BSN radio platforms are also reported as a challenge to resolve

in Ren et al. (2011), and it is stated that medical sensor devices often use heterogeneous radio platforms such as CC1000, ZigBee/CC2420 and Bluetooth, as it is indispensable to achieve the performance assurance in a radio-agnostic manner to support platform portability. Inter-operation with other existing body area networks requires standardization of the protocol stack and the format based on which the data storage tasks are done (Hanson et al., 2009).

1.4.6 Self Organization

Another difference between WSN and WBAN is the ability of nodes to self-organize when a node failure happens. Nodes in a WBAN with a star topology do not have this capability but represent less complexity in their infrastructure and do not suffer from ad-hoc oriented problems such as idle listening and overhearing (Omeni, Eljamaly et al., 2007). It is also possible to have a network of coordinators so that the problem of a single point of failure in the star topology will be eliminated (Hanson et al., 2009). We also reviewed WBAN with multi-hop topologies such as the ones in Chipara et al. (2009), Ko et al. (2008) and Curtis et al. (2008) where nodes could automatically find other paths upon a failure to their coordinator when operating on CTP routing protocol in TinyOS. However most of the protocols in the literature relate to one single cluster, which is a single star topology.

1.4.7 Interference

Radio links in a WBAN are more susceptible tointerference from other wide range networks. The choice of medical bands will decrease the interference. The interference problem has been studied widely in WBANs as the IEEE 802.15.4 radio links are short-range low-data rate links, they are mostly exposed to interference from other high data rate bands like 802.11 radios, or even devices operating on the same 2.4 GHz band link. Some relevant references are Miluzzo, Zheng, Fodor, & Campbell (2008), Xi, Guo, & Shu (2011), Kara & Bertoni (2001), and Obayashi & Zander (1998). In some research works such as MEDiSN (Ko et al., 2008), interference can be modeled in a simulation environment in order to more illustrate the challenges of a real implementation before prototyping it. In Ko et al. (2008), the authors take advantage of a noise trace collected at the Stanford Meyer Library (Lee, Cerpa, & Levis, 2007) to add link noise to their simulations when evaluating the effects of interference in a simulation environment. A dynamic approach to adjusting the maximum threshold on the number of retransmissions by the relay nodes at network layer in the WBAN implemented in Ko et al. (2007) is used to address the effect of interference and to improve the delivery ratio of the system.

1.4.8 **Reliability**

Although energy preservation still remains as one of the challenges in a WBAN design, it may not stand as the first crucial task to deal with when it comes to healthcare applications. This makes more sense when, in many of the reviewed pilot studies or short-term real implementations of a WBAN in hospital environments, changing and recharging the batteries of medical sensors remains one of the personnel's responsibilities. Although the research direction for many of the healthcare and sport applications is towards having a WBAN that would possibly need no change or recharging of the batteries, even in most of today's implementations of WBAN for medical use cases, the batteries of the medical sensors will need to be changed at least once or twice a week, depending on the physiological activity they monitor. Reliability, however, stands as the most important factor when dealing with medical data as they are the only evidence to a patient's current health status and are the only clues a physical practitioner has when monitoring a patient remotely. In Huq, Dutkiewicz, Fang, &Ren (2012), WBAN applications are said to be loss-sensitive and delay-sensitive rather than being in stringent need of energy efficiency, claiming reliability as the most important factor in medical applications. The concept of reliability is treated as the level of fidelity or the quality of the received data at the coordinator device, and has been formulated in different ways in different research papers. While we mostly focus on data reliability in this thesis, system reliability could take on a broader meaning and would refer to the overall medical system performance implemented on a WBAN. We will see in Section 5.1.1.1 how we have measured data reliability by averaging the number of successfully received packets from all the physiological sensors at the coordinator device. As also mentioned earlier in Section 1.3, some research works such as Chipara et al. (2010) have categorized reliability based on the main causes of packet loss in the system. "Sensing" and "networking" are the two types discussed by the authors, with "sensing" referring to losses occurred by a link disconnection between the sensor node and its coordinator and "networking" referring to the packet losses imposed by errors related to the wireless channel, which could be collisions at MAC level, buffer overflow, etc. Not much research has been done on sensing-related reliability problems, as most of the works in the literature concentrate on either a simulative or analytical approach to evaluate their implementations.

Reliability may not always be desired at the same highest level for every application. For instance, the level of reliability expected for a simple mobile monitoring system mounted on

the wrist of an athlete that would provide calculations of footsteps or calories need not be at the same fidelity level considered for clinical standards for applications such as ECG monitoring (Bonnici et al., 2012).

1.5 Medium Access Control (MAC) Protocol

A protocol is typically a long set of programs that operate as rules within the operating system of a sensor platform to govern all the communications (data transmission/reception) that happen during the course of a data transfer. Therefore, a protocol can be designed to fulfill a particular advantage that is directly linked to the running application's requirements. A standard would consist of protocols, each ruling a certain layer. Each layer of the protocol stack can be independently improved and enhanced knowing that protocols following the same standard will be compatible with each other. The collaboration between the layers of the protocol stack is maintained by the services that each lower layer is responsible to provide for the layer sitting above it. Therefore there exist interfaces between the layers that act as logical links between them. A medium access control protocol, as its name implies, is responsible for configuring reliable communication links between the nodes and the shared medium. It controls the radio as the most power-consuming component in any sensor node and therefore effective and appropriate techniques could be integrated to boost the performance of the network in terms of its requirements in a MAC structure (Mahtab Alam et al., 2012). Whilst it is actually the PHY layer that turns the radio of a sensor node ON or OFF, it gets the command from the MAC layer in the form of a request and then it is the responsibility of the PHY layer to respond to this request. The sensor node then has less than a 12-symbol period of turnaround time to change its radio state (IEEE, Std. 802.15.4, 2003).A PHY layer contains the radio frequency transceiver with its low level control mechanism (IEEE, Std. 802.15.4, 2003). Since the medium (wireless channel) is scheduled by the MAC protocol, it is indeed the MAC protocol that assigns or dedicates the requested time slots for every single communication to be made over the channel. However, the timing is not always at its optimum in accordance with the network's demands. It has been repeated throughout the literature that proper channel access and resource allocation stands as an important factor in controlling the energy usage and also other performance parameters in the network (Zhisheng & Liu, 2011), which is what a medium access control protocol does. In any wireless communication, the concept of a MAC protocol can be divided into two broad categories: schedule-based and contention-based. The classification suggests the main strategy incorporated for the wireless channel access and the question in any of these categories is: in what order is the channel access granted and how?

Sections 1.5.1 and 1.5.2 discuss these two main categories of existing MAC protocols along with their advantages and disadvantages. In general, different MAC techniques are to make different tradeoffs to suit the variety of situations that may be caused by the existing traffic type of a network and they can also be mixed to create possibilities of handling diverse scenarios for different application types. Therefore the evaluation of each MAC technique should be merely based on what certain QoS requirements that specific MAC is intended to look after, based on which it has been optimized to meet those certain criteria. A table of comparison between TDMA (schedule-based) and CSMA (contention-based) techniques appears in Ghildiyal, Godara, & Amara (2011), where the use of each technique is recommended based on different levels of data priority and required throughput for each application as the main parameters of importance (chosen criteria) for their particular application. It is recommended that pure contention-based techniques would be only suitable for applications of low or moderate priority and a low throughput demand. A schedule-based technique has been identified as a more preferable approach for low or moderate priority applications but with a higher throughput requirement. Since energy consumption can never be considered out of the equation for any sensor network structure, most of the proposed MAC techniques aim to have a hybrid approach in which the two different MAC techniques function in parallel to provide a reasonable balance among target performance metrics. An example of this could be Boulis & Tselishchev (2010), who examine a mixed contention and polling type of access, or as in Liu, Yan, & Chen (2011), where a hybrid of contention- and schedule-based access is proposed. Hybrid protocols are the best possible solutions to respond to a highly variable traffic characteristic, which is very typical in WBANs as well. A full review of such hybrid techniques and their main advantages and disadvantages in different application scenarios appears in Sections 1.8 and 1.9, with a focus on WBAN MACs only. Both MAC protocol approaches are discussed in Sections 1.5.1 and 1.5.2 below to explore their main strategy and characteristics.

1.5.1 **Schedule-Based MAC**

In a scheduled scheme, the channel is divided in parts and interference-free collision-free access, which can be based on time, frequencies or codes. The main characteristic is that the channel will be devoted to potential users regardless of their current need. This makes these types of access mechanisms inflexible for wireless sensor networks and ad hoc networks due to their high dynamic topology changes, and makes them more suitable to be used in a cellular network. In WBAN, a schedule-based access mechanism is well suited for delay-critical

applications and where the medical sensor node is responsible for a fast data relay to its coordinator. One example of such an application could be EEG sensor data for a patient suffering from epilepsy seizures. Abnormal sensory readings of the patient's brain activity will be transmitted to the coordinator device as fast as possible so that appropriate actions such as alarms to the patient to sit or lie down could be taken. Schedule-based techniques are categorized into four major groups based on how they are given access to the shared medium, which could be one of the dimensions of frequency, time, code and space. They are Frequency-Division Multiple Access (FDMA), Time-Division Multiple Access (TDMA), Code-Division Multiple Access (CDMA) (Bechler et al., 2003), and Space-Division Multiple Access (SDMA). Among all the fixed assignment access schemes, TDMA received lots of attention in wireless sensor networks for the possibility of the reserved time slots it offers. In a TDMA access format, time is divided into several time slots while having a common frequency. However, the entire data will not be sent during one single time slot due to the very short duration of a slot, but the time difference between two consecutive time slots is negligible. There can be the probability of data being lost between data frames. LEACH (Heinzelman et al., 2000) is a good example of a TDMA technique in WSN. The nodes inside a virtual cluster have access to the channel on a TDMA basis. Contention-free protocols lack flexibility in accommodating the rapid changes in an Ad hoc or WSN. Channel assignments can only be scheduled to a new node joining the network in the next scheduling period. Topology changes are also not supported. WBAN, however does not suffer from such vigorous behavior in its topology and therefore can exploit a TDMA-based access technique to send time critical and high priority data packets. We will review some of such works in Section 1.8.1.

1.5.2 **Contention-Based MAC**

A contention-based technique basically allows a group of sensors that share one wireless channel to contend for having access to it. The possibility of collisions automatically increases in such a scheme as nodes may find themselves transmitting over the channel at the same time. This has been alleviated by different standards such as IEEE 802.15.4 by incorporating a technique that would make nodes belonging to a network listen to (sense) the channel before they transfer their data over to it. A detailed description of this access scheme appears in Section 1.7.1.1. Contention-based MAC protocols are infrastructure-free, which exhibits great scalability and flexibility. Unlike contention-free access methods, this group of protocols does not work on a pre-allocation basis. The nodes contend for access to the medium as they find themselves with something to send. There is a difference between these types of protocols from

demand-assignment protocols: demand-assignment protocols need a logical controller to assign the channel to them while a contention-based protocol is not in the need of a network controller. Demand-assignment protocols were introduced to refine the inefficient behavior of fixed assignment access methods by reassigning the unused channel (Edgar & Callaway, 2005). However in a contention-based access scheme, the concept of *carrier sensing* is employed for less collision-prone access. The expression *carrier sense* simply means sensing the channel when needed before having access to it and is defined for these kinds of access methods with no sure channel reservation guaranteed for the potential users before they contend. Carrier Sense Multiple Access (CSMA) was the first access method developed for accessing the channel by listening before transmitting. CSMA is further divided into three main categories: 1) non-persistent, 2) 1-persistent, and 3) p-persistent (Kleinrock & Tobagi, 1975). However, CSMA techniques did not perform well in a multi-hop network such as ad hoc or WSN operating on IEEE 802.11 because of the *hidden terminal* problem. The hidden terminal problem is a situation in which a node may cause a collision because it is not able to hear an ongoing transmission from the other node. Figure 1-3 shows this phenomenon. "A and B" as well as "B and C" are able to hear each other while "A and C" are not. Now if C tries to send something to B while A is busy with transmission to B, a collision can simply occur because of C's inability to hear messages from A (Willig, 2006).

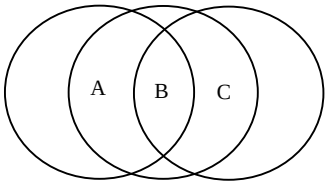


Figure 1-3: Hidden terminal problem

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for IEEE 802.11 WLAN was therefore introduced to solve the hidden terminal problem and is based on a handshake technique between sender and receiver. An exchange of Request To Send/Clear To Send (RTS/CTS) messages can solve this problem by simply letting node C hear the CTS message of node B and making sure that the channel is already in use by a previously sent RTS from another node like A. However, the probability of collisions in these schemes exists, like a collision occurring between two RTS messages. The RTS/CTS mechanism is not available in IEEE 802.15.4 as we will see later in this chapter, but it does not cause a problem in small-scale WBAN with a star topology. Multiple Access with Collision Avoidance (MACA) (Karn,

1990), MACA for Wireless (MACAW) (Bharghavan et al., 1994) added some changes over CSMA/CA, such as a duration field to both RTS and CTS messages known as Network Allocation Vector (NAV). IEEE 802.11 distributed coordination function (DCF) refers to all these three access methods of CSMA/CA, MACA, and MACAW, and supports communications in an ad hoc network along with some added features such as Binary Exponential Backoff (BEB), which also exists in IEEE 802.15.4. CSMA/CA is explained in Section 1.7.1.1. IEEE 802.11 also adopts a distributed polling method for networks enabled with an access point like cellular networks. All these schemes work on a carrier sense basis. The collision avoidance aspect of IEEE 802.11 DCF is the BEB algorithm that generates backoff times to further decrease the probability of two or more users accessing the medium at the same time and thus avoiding more collisions. The concept of backoff time (required waiting time) of any contending node before it is actually permitted to transmit its data over the wireless channel is explained in Section 1.7.1.1, which elaborates on that of IEEE 802.15.4. Backoff times are the standard's resolution to lessen the probability of collisions when several nodes contend with each other to have access to the shared wireless channel. We will see later in Chapter 4 how we have introduced a dynamic and traffic-reflective effect on the length of this waiting time as a means to provide fair access among the physiological nodes of a WBAN.

1.6 **IEEE 802.15.4 Protocol Stack**

IEEE 802.15.4 standard defines the rules and guidelines for the two Physical and MAC layer protocols. IEEE 802.15.4 is the standard currently being used for many WBAN scenarios both in academia and practical deployments. It targets short range, low cost, and low data rate communications and is mostly associated with ZigBee ("ZigBee Alliance," 2014) to build up a complete protocol stack for such networks. However, IEEE 802.15.4 only specifies the physical (PHY) and MAC protocol primitives and ZigBee defines the specifications for network and application layers (Mahalik, 2007, p. 26). The literature works in this thesis are benchmarked against IEEE 802.15.4 as a lightweight protocol that is able to construct energy-aware communications in short ranges. Figure 1-4Figure 1- shows the simplified protocol stack for IEEE 802.15.4 standard together with ZigBee. It shows the block diagram of the IEEE 802.15.4 standard and an overall view of the co-operating services that act as interfaces primarily between the MAC and PHY layers. Each layer has to offer services to the other layer(s) according to the Figure.

Two physical layers existing in the standard (IEEE, Std. 802.15.4, 2003) are 868/915 MHz and 2.4 MHz with a maximum data rate of 250 Kbit/s when operating on a 2.4 MHz band, which is good enough for low data rate communications in WBANs. The two categories of the frequency bands (868/915 MHz and 2.4 GHz) differ in their modulation techniques; the 868/915 MHz frequency bands use Binary Phase Shift Keying (BPSK) whilst the 2.4 GHz band uses Offset Quadrature Phase Shift Keying (OQPSK). With the exception of interference from other devices operating on the same band and also body attenuation, the global availability of the 2.4 MHz band (ISM band) and also its energy efficient operation has made it, by far, the most accepted PHY band almost by all industry vendors in healthcare (Espina, Falck, & Mulhens, 2006, p. 161). The PHY layer accesses the MAC layer through two different service access points (SAPs) that provide interfaces to the MAC layer and are known as Data Service and Management Service.

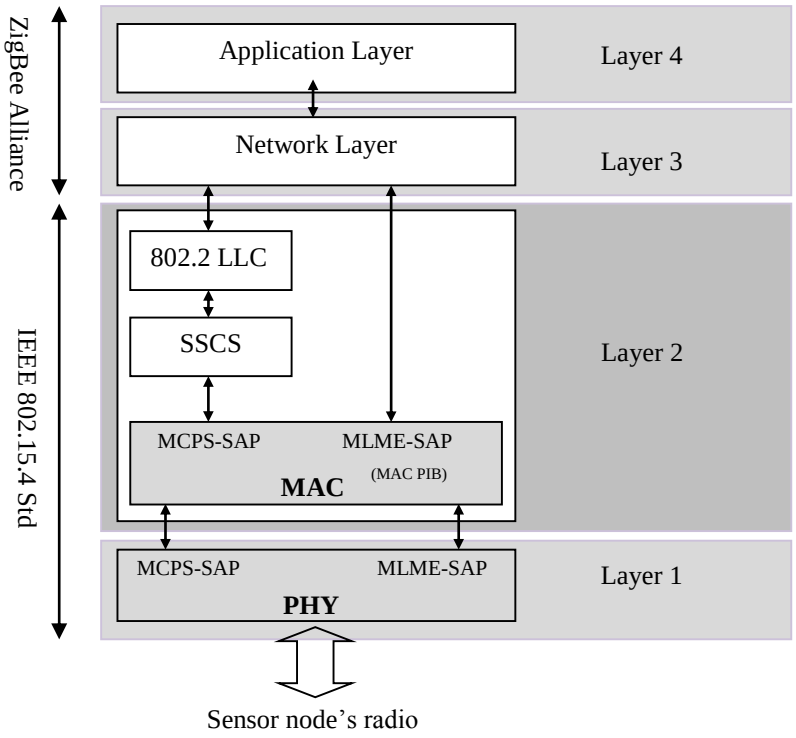


Figure 1-4: IEEE 802.15.4 and ZigBee protocol stack

The PHY data service enables the transmission and reception of the Physical Protocol Data Units (PPDUs) over the wireless channel, whereas the MAC data service enables the transmission and reception of the MAC Protocol Data Units (MPDUs) across the PHY data service ("IEEE, Std. 802.15.4", 2003).

Respectively the MAC layer interfaces with its upper layer, the network layer, through its two access points known as MCPS-SAP and MLME-SAP, which provide MAC Common Part Layer Service (MCPS) and MAC Layer Management Entity (MLME) service. It is through these access points that an upper layer can send its requests to the lower layer. When two sensor devices communicate with each other through their radios, the MCPS unit of the sender liaises with the MCPS unit of the receiver node for the two MAC layers to communicate. Likewise the MLME unit provides all the services in regard to management functions between the two MAC layers and it also has access to MCPS services ("IEEE 802.15.4 User Guide", 2006). MAC PIB is a database created over all the managed objects by the MLME that are related to the MAC layer. The MAC sub-layer provides its services through its two SAPs to the LLC sub-layer so the MAC sub layer is always beneath the LLC sub layer. LLC is responsible for tasks such as error control and retransmission of dropped packets ("FreeScale IEEE 802.15.4 Std/ZigBee", 2010). It acts as an interface to the upper layer, which is the network layer. It does the error controlling by adding some control messages to the data frame coming from the network layer before such data is sent to the next hop, which could be another RFD or the coordinator, and takes control of the sequence of acknowledgement messages. The information added to the frames by LLC is usually called the LLC Header. The importance of the amended header is in the information it carries about the source address of the network layer entity where the message originated and also the destination address of the device intended to receive the message. The information in the header also includes a control field which takes care of the flow of the messages. When network layers of two entities want to exchange packets, the network layer of the source entity needs specific services from its lower MAC sub-layer.

Based on how the source network layer wants to communicate with its peer network layer, two types of connections can be established here. The first type is when the network layers want to communicate without the establishment of a data link level connection that is called "unacknowledged connectionless mode service", and only takes advantage of one service (data transfer) with two primitives of "request" and "indication". Therefore the PDUs get transmitted between the LLCs of the two entities with no data link layer connection and they will not be acknowledged upon reception and there will not be any possibility of error correction or flow control ("IEEE 802.2: Logical Link Control", 1998, p. 23). The second type is called "connection-mode service" and requires the two peer LLC sub-layers to have established a data link layer connection before any PDU exchange. PDUs can then be transmitted between the source and destination entities' LLCs over the data link layer connection, ensuring that an

acknowledgement has been received from the destination LLC. There are five services provided in this type, namely: connection establishment, data transfer, connection termination, connection reset, and connection flow control which take advantage of four types of primitives: “request”, “indication”, “response”, and “confirm”. The use of a service-specific convergence sub-layer (SSCS) is optional. Using SSCS ensures the compatibility of the different LLC sub-layers. The SSCS acts as an interface between the MAC sub-layer and the LLC sub-layer, meaning that LLC can choose to receive services from the MAC layer through an SSCS or directly. More details of these services are really not related to the focus of this research work but before proceeding to the next section we will introduce some general facts about these four layers of the IEEE 802.15.4 protocol stack followed by a more detailed overview of the existing primitives in a MAC layer that are necessary to understand the coding parts later described in Chapter 6, the real BAN implementation. Here is an overview of the MAC sub-layer responsibilities as appearing in the standard (“IEEE, Std. 802.15.4”, 2003). MAC features:

- Beacon management (for the sake of synchronization)
- Channel access (different scan types in the standard)
- GTS management
- Frame validation
- Acknowledge frame delivery
- Association and disassociation

We will see later on in this section how these features are provided in the form of services and their primitives at the MAC sub-layer level. The flow of the primitives between the layers to provide each service is accomplished by exploiting different frame types that change in their fields as they travel through the layers of the protocol stack. Four different types of frames are used: beacon, data, acknowledgement (ACK), and MAC command frames, each with their own specific structure. Layer-specific headers and footers are added as these frames travel through the different layers during the course of a transmission. Each frame type (beacon, data, MAC command, or ACK) is composed of three main sections known as headers (as the prefix to the payload), payload, and footer (as the postfix to the payload). The primary difference of the frames’ structure, as seen in Figure 1-5 to 1-8, lies in their payload section and also the layer they originate from. For example, the MAC data frame differs from the other three in the fact

that it does not originate from within the MAC layer but from the application layer. The acknowledgement frame differs from the other three in the fact that it does not have a payload section as opposed to beacon, MAC, command, and data frames. The frames originating from the MAC sub-layer are considered as MAC Protocol Data Units (MPDUs) that will turn into Physical Protocol Data Units (PPDUs) as they pass through the data link layer to the physical layer. The former MPDU will turn into a Physical Service Data Unit (PSDU) as the payload section of a PPDU. The payload of an MPDU itself is called the MSDU which is a MAC Service Data Unit.

Figure 1-5 to 1-8 show how the data payload of an MPDU converts to the data payload of a PPDU when the packet reaches the PHY layer of the protocol stack. Table I relates each figure to the type of the frame it illustrates along with the description of the terms used in the figures. As shown in the figures, only the payload section of an acknowledgement frame differs from the other frame types and it is none. A more detailed description of the length of these packets and the number of bytes dedicated to each segment appears in the standard ("IEEE, Std. 802.15.4", 2003) and we do not discuss them in this thesis except for parts that are really related. A Frame Check Sequence (FCS) is to check and detect possible errors in every transmitted frame that employs cyclic redundancy check (CRC).

Table I: Packet Types

Figure	Packet Type	Acronyms
Figure 1-5	Beacon Packet	FCS: Frame Check Sequence SHR: Synchronization Header PHR: PHY Header PSDU: PHY Service Data Unit
Figure 1-6	Data Packet	
Figure 1-7	Acknowledgement Packet	
Figure 1-8	MAC Command Packet	

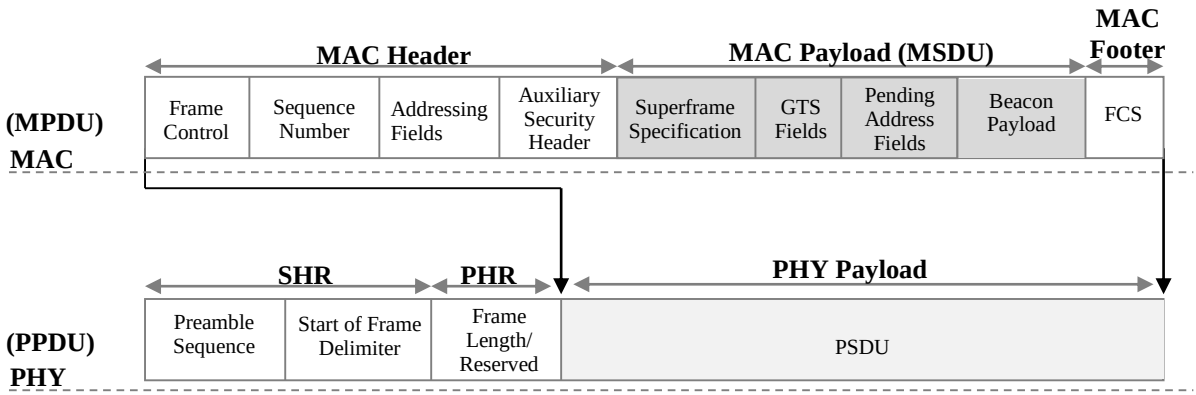


Figure 1-5: Beacon MPDU at MAC Layer Converting to PPDU at PHY Layer

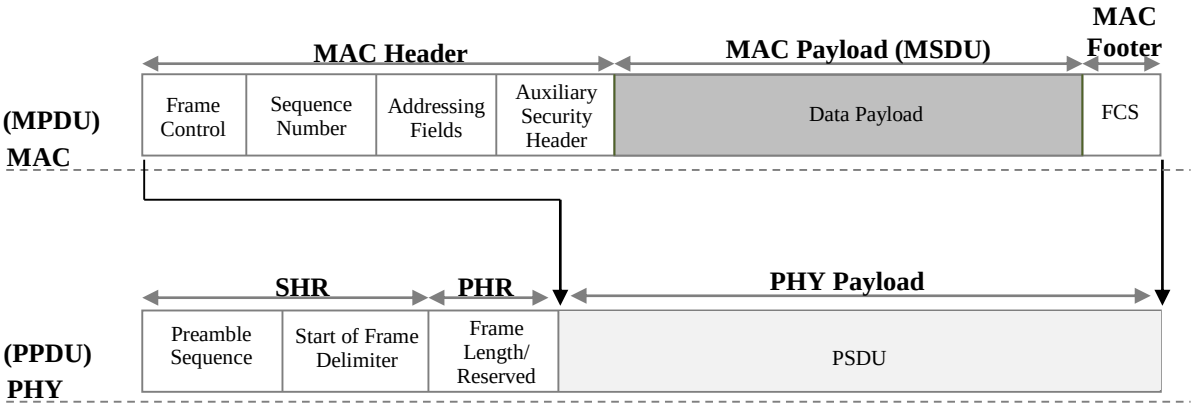


Figure 1-6: Data MPDU at MAC Layer Converting to PPDU at PHY Layer

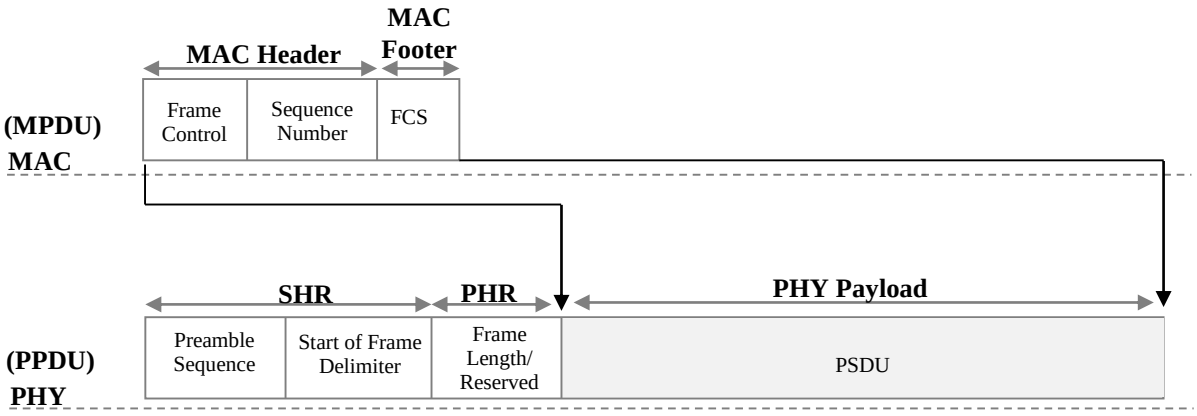


Figure 1-7: Acknowledgement MPDU at MAC Layer Converting to PPDU at PHY Layer

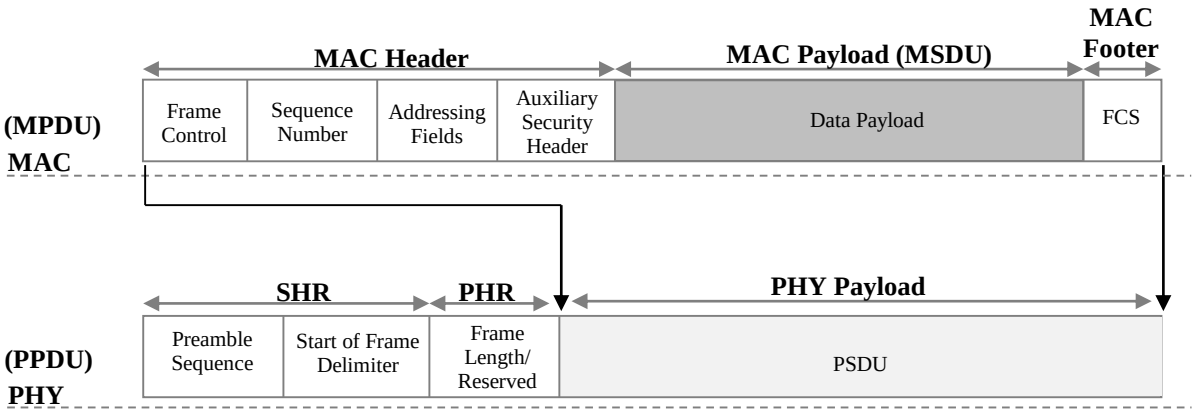


Figure 1-8: Command MPDU at MAC Layer Converting to PPDU at PHY Layer

Figure 1-9(a) and Figure 1-9(b) below depict a simple overview of the sequence of messages passed between a typical RFD node and the coordinator when either of them initiates the transmission. Different tasks carried on by both the sensor device (highlighted as light grey) and the coordinators (highlighted asdark grey) are shown in the blocks. In simple words, the description of two ways of transmissions is as follows:

If an RFD device has data to send to the coordinator, the main steps to follow as shown in Figure 1-9(a) are:

- If an RFD has data to send, it sends a request upon that data to coordinator
- The coordinator acknowledges that request with an acknowledgement frame
- The device will send out the data upon the acknowledgement
- The coordinator confirms the reception of data.

If the coordinator device has data to send to an RFD device, the main steps to follow as shown in Figure 1-9(b) are:

- If the coordinator has data to send, it sends a beacon with a “data message pending”
- If the sensor sees a message pending, it transmits a MAC command to request data
- If the coordinator receives the data request, it sends an acknowledgement
- The coordinator sends the pending data message
- The sensor acknowledges the data reception.

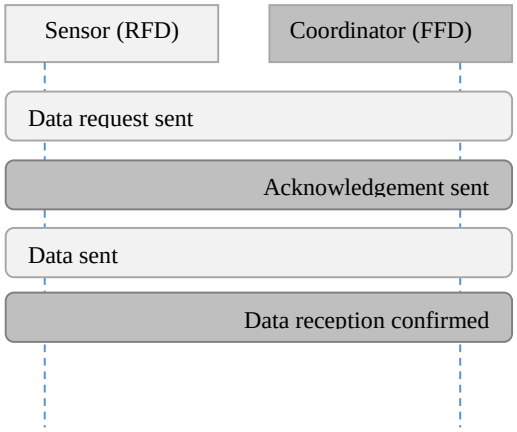


Figure 1-9(a): Data transfer from an RFD to a coordinator device

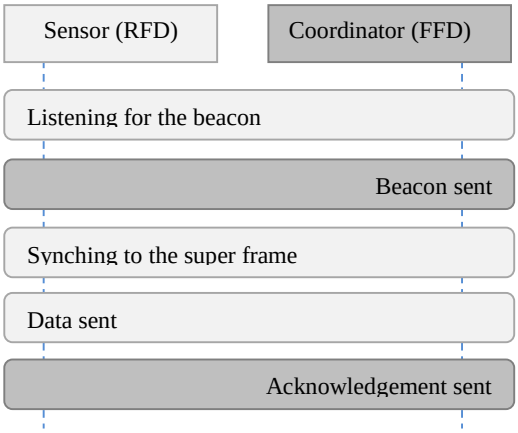


Figure 1-9(b): Data transfer from coordinator to an RFD device

Acknowledgements are optional andcan be sent upon a successful transmission before a short period of time has passed. If an acknowledgement message is not received after a certain timeout, the packet has to be retransmitted, for which there is a limited number of times configured in the standard. In case of a no acknowledgement message, every transmitted packet is assumed to be a successful delivery. It was mentioned earlier how the data transmission between these two types of devices, as shown in Figure 1-9(a) and Figure 1-9(b), would be accomplished by a series of services between the layers, which was described early in this section, and is provided via service access points (SAP). The two services offered by a MAC layer, as the focus of Section 1.7, were declared data and management services that are provided through MCSP-SAP and MLME-SAP access points. Each service as described in the standard has two main features known as service primitives and service parameters. The block diagram of the two services of a MAC layer, data and management, along with their primitives is shown in Figure 1-10 and is explained extensively in the standard, which we briefly review here for a better understanding of the implementation parts in Chapters 5 and 6 where we actually had to include some of the pseudo-codes of the proposed MAC within IEEE 802.15.4 MAC specifications. As can be seen in Figure 1-10, different sets of primitives are provided by a MAC layer to the higher level (application layer) through its two different SAPs. These primitives act as action functions between PHY and MAC. The only two primitives provided by the MAC's data service are MCSP-DATA and MCSP-PURGE, whilst the primitives provided by the MAC's management service are MLME-ASSOCIATE, MLME-DIASSOCIATE, MLME-BEACON-NOTIFY, MLME-GET, MLME-GTS, MLME-ORPHAN, MLME-RESET, MLME-RX-ENABLE, MLME-SCAN, MLME-COMM-

STATUS, MLME-SET, MLME-START, MLME-SYNC, MLME-SYNC-LOSS, MLME-POLL (“IEEE 802.15.4 Std”, 2003).A close look at the services provided by each primitive reveals the four general types of the services available that can be offered by a specific primitive. The four types of services are:

- Request
- Indication
- Response
- Confirm.

All these services are provided by the MAC sub-layer to its next higher layer (network layer) through the LLC sub layer.

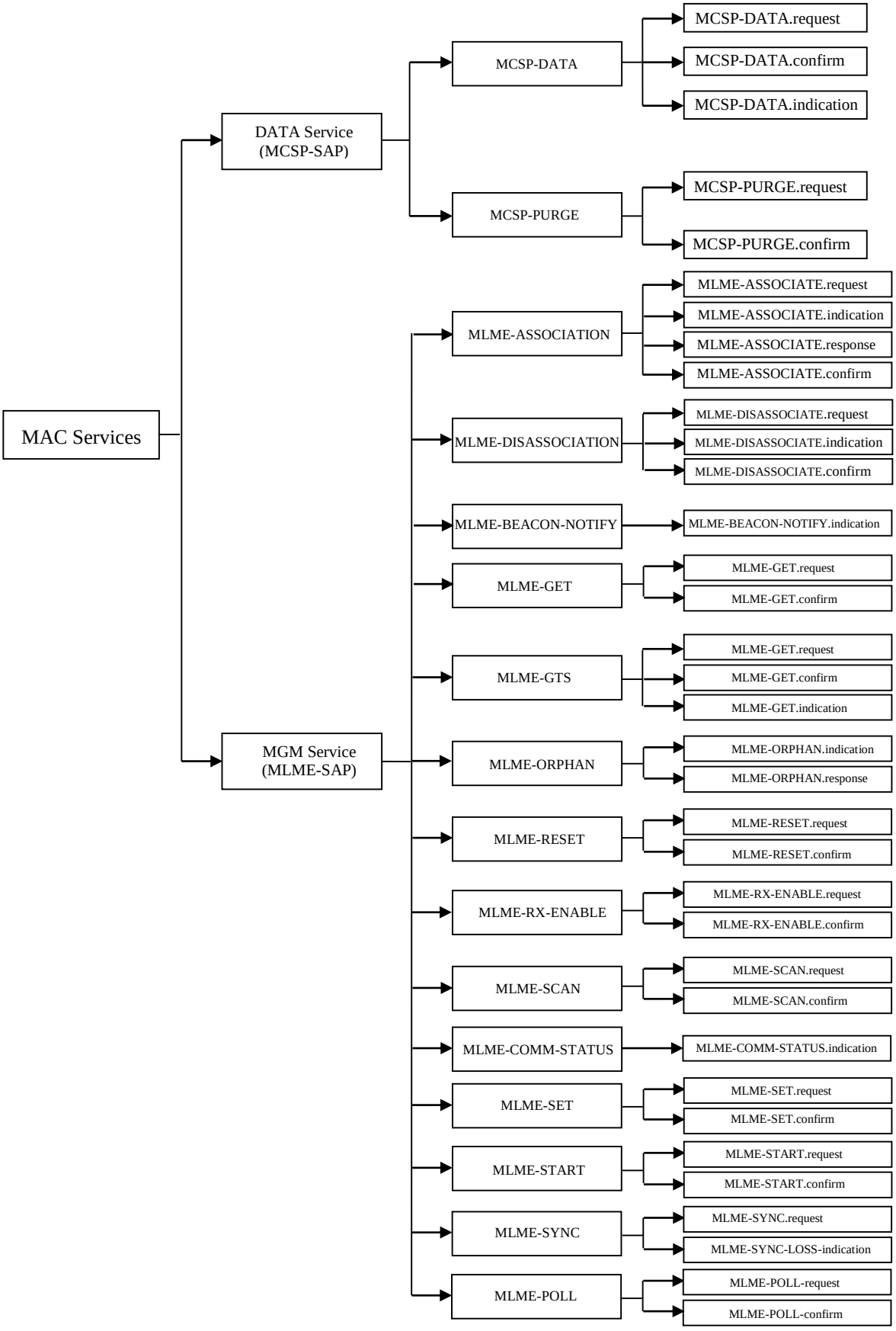


Figure 1-10: Block diagram of the MAC layer data and management services along with their primitives

- MCSP-DATA.request is responsible for the transmission of an MSDU at MAC layer through the SSCS interface. It will take care of the acknowledgement inquiries and also the GTS transmissions if they are requested and so does it take care of the security requirements of the transmitted frame. Frame-length checking is also performed by this primitive to make sure the requested transaction can fit into the CAP or CFP of the super frame. MCSP-DATA.confirm is issued based on the result of all the evaluations performed by the MCSP-DATA.request primitive and is done through the SSCS interface. It could either be a SUCCESS outcome or it will generate a related error message upon the failed transmission request. MCSP-DATA.indication is generated once the data frame reaches the MAC layer of the peer entity through the SSCS.

- MCSP-PURGE.request is used when an MSDU needs to be removed (purged) from the transaction queue at MAC level. The MCSP-PURGE.confirm is issued when the particular MSDU that the MAC sub-layer has tried to purge from its queue has been found in the queue and purged successfully, otherwise no confirmation will be issued and the MCSP-PURGE.confirm will set its status to INVALID_HANDLE. Handles are specific to each MSDU and MSDUs can be identified by means of their handles.

- MLME-ASSOCIATE.request is sent by an RFD device to the coordinator device when the RFD intends to join the network of that specific coordinator. The request is generated from the RFD device's higher layer and passed through to the MLME of the coordinator. MLME has the responsibility of updating the PHY and MAC PIB attributes for its device before generating the association request command. Certain parameters such as the level of security have to be considered upon such request. The association could be processed with or without sharing a key between the RFD and the coordinator to either apply security or not. Sometimes the association request may not get through, i.e. when the channel status is busy, which will generate a failure message. MLME-ASSOCIATE.indication is generated by the MLME of the coordinator device to its next higher layer. The coordinator then issues the MLME-ASSOCIATE.response upon its decision and in response to the indication primitive it received. The successful transmission of an association request by the MLME will result in MLME-ASSOCIATE.confirm which could be with or without an acknowledgement message and is issued to the next higher layer of the RFD device.

- When a device wants to be disassociated from its coordinator, it will send an MLME-DISASSOCIATE.request from its next higher layer to its MLME thatwill be sent to the MLME of the coordinator device and will be indicated to the coordinator's next higher layer by the MLME-DISASSOCIATE.indication primitive. The PANId in the request will be checked against that of the coordinator's to make sure of the coordinator's identity. Once the disassociation request has been processed by the coordinator successfully, the next higher layer of the RFD device will be informed of the result by an MLME-ASSOCIATION.confirm message issued from its MLME which shows a success. Note: Disassociation could also be initiated by the coordinator device which we do not discuss here but the reader is encouraged to see (IEEE, Std. 802.15.4, 2003, p. 92).

- Each RFD device has to be notified of the reception of a beacon message, which will be done through MLME-BEACON-NOTIFY service. When an RFD device receives a beacon, its MLME will notify the higher layer of its arrival. The main purpose is to notify the higher layers of the parameters inside the beacon frame.

- MLME-GET.request primitive is used when a device's higher layer needs to extract information from the PIB database of attributes. If the requested attribute does not exist in the PIB database, the result of such a request will be MLME-GET.confirm but with a status of UNSUPPORTED-ATTRIBUTE. There exist two types of attributes, PHY and MAC, and therefore if the requested attribute is not a MAC attribute but is a PHY one, it will be sent to PHY via PLME-GET.request primitive, which is similar to MLME-GET.request and at the physical layer. The MLME-GET.confirm will be issued once the requested MAC attribute is found in the PIB.

- MLME-GTS primitives let a device communicate with its coordinator upon needing GTS slots for priority data transmissions. The request comes from the RFD device's higher layer through MLME-GTS.request primitive and it will be sent to its MLME and then to the coordinator's MLME for confirmation. The RFD device can not only inquire a GTS allocation but also a deallocation of its previously requested GTS slots. The device will let the coordinator know about the details of its request in the parameters it provides in its sent MLME-GTS.request primitive. Possible errors to occur during such a request are when a device is not permitted to send such a request and when the request does not get through because of a channel access failure. Once the GTS request is sent successfully to the coordinator's MLME, it will expect an acknowledgement message followed by an MLME-GTS.confirm. After the request has been processed, based on the outcome of the request assessment, the MLME-GTS.indication will be issued that contains the allocated GTS and the descriptors and also the RFD device's short address. If all the information matches, the device will then issue an MLME-GTS.confirm through its MLME with a status of SUCCESS.

- MLME-ORPHAN.indication primitive is used to notify about an orphaned device. The indication is sent from the device's MLME to the coordinator's MLME and then to coordinator's higher layer once a device has been orphaned. Upon such indication it has to be checked whether such device has really been associated with this network or not. The coordinator's next higher layer will send an MLME-ORPHAN.response with the parameter AssociateMember set to TRUE when it confirms the device was previously associated with the same network. The RFD device has to then receive the coordinator realignment command via coordinator's MLME, which should not take any longer than *macResponseWaitTime*symbols. The MLME coordinator shall then send an MLME-COMM-STATUS.indication to its higher layer with its related status, which could either be a busy channel or a SUCCESS.

- MLME-RESET service is used when resetting the MAC sub-layer to its default values. The request will be sent from the higher layer through MLME-RESET.request primitive. Upon sending out this request, a parameter called "SetDefaultPIB" will be evaluated, which is of Boolean type. A true value of it will rest the MAC sub-layer. This primitive is always used before either the MLME-START.request or the MLME-ASSOCIATE.request. The result of this operation will be generated in MLME-RESET.confirm primitive, which will issue a Enumeration value with a status of SUCCESS to show the successful delivery of the service.

- The receiver of a device can be enabled or disabled by the MLME-RX-ENABLE service. Certain activities can be done through this service, for example, if deferring an operation to the next super frame is possible or not. It will examine such a possibility by the MLEM-RX-ENABLE.request primitive. The time slot when the receiver is to be enabled and also the duration is specified by this primitive's parameters. The receiver can be disabled similarly by using the same primitive. Once the receiver has been enabled to receive, it may be interrupted

by another operation, which is a conflicting responsibility. The device has to then first perform that operation and then resume its receiving operation only if the duration specified in the request primitive has not expired yet. Upon a successful enabling of the receiver, the MLME will issue an MLME-RX-ENABLE.confirm primitive with a status parameter set to SUCCESS (the parameter is of Enumeration type).

- MLME-SCAN is for determining the level of energy usage and absence and presence of the other PANs in the channel. This is done by the device doing a scan over a list of channels to search for the coordinator it is associated with and also to learn about the energy of the channel. MLME-SCAN.request primitive takes advantage of a list of parameters such as the type of the scan (energy detection, active, passive, and orphan), which channels to scan, the length of the time to spend for scanning, the channel page, the security level to be used, etc. The results of the scan will be reported to the next higher layer by the MLME-SCAN.confirm primitive, in which the type of the scan will also be mentioned. There may be some channels that were requested to be scanned but were not that will be listed as well.

- Indication by MLME to the higher layer about the communication status is done by the MLME-COMM-STATUS.indication, which will inform the next higher layer of two important things: transmission status and security errors. Important information such as the address of the originating device, the destination address that is intended to receive the frame, the communication status that can take 13 different Enumeration values such as SUCCESS, CHANNEL_ACCESS_FAILURE, SECURITY_ERROR, etc., and also some other parameters, will be clarified in this primitive.

- MLME-SET primitive is for when a given value has to be written in the PIB database. The attribute to be written in the PIB database has three key parameters, which will be the parameters of the MLME-SET.request primitive. They are: the identifier of the PIB attribute, the index within the table of the specified PIB attribute, and the value to write to the indicated PIB attribute. Upon successful writing of the requested PIB attribute, the MLME will issue the MLME-SET.confirm with a status of SUCCESS with the other two parameters of identifier and index.

- On two occasions the MLME-START service will be needed: the first is when the coordinator device (FFD) wants to start using a new super frame configuration to initiate a PAN and the second use is when an already associated device (RFD) wants to begin using the new super frame configuration. In either of these cases, the MLME will receive the MLME-START.request primitive from the next higher layer with a set of parameters. There are 17 parameters for this primitive, such as the PAN ID, the logical channel (on which to start using the new super frame), the start time (the time at which to start transmitting beacons), beacon order (BO), super frame order (SO), the Boolean parameter to examine if the device sending the request will become the new PAN coordinator, etc. The result of this sent request will be reported by the MLME-START.confirm which could take different values such as CHANNEL_ACCESS_FAILURE, NO_SHORT_ADDRESS, SUPERFRAME_OVERLAP, FRAME_TOO_LONG, SUCCESS, etc. (discussing these different values falls out of the scope of this research but they are elaborated wherever related).

- Synchronizing with the coordinator will be carried out by the MLME-SYNC service that is available for a beacon-enabled PAN. The loss of synchronization will be reported to the next higher layer by this service as well. MLME-SYNC.request primitive will send a request for

synchronizing with the coordinator by requesting a beacon frame. If it is specified in the parameter list of the MLME-SYNC.request primitive, the device may also track the beacon frames coming from the coordinator, otherwise it will just locate them. The MLME-SYNC-LOSS.indication is to notify about a lost synchronization with the coordinator, which has eight parameters such as the reason of the loss, the PAN ID, the channel the device lost its synchronization on, etc. This indication is generated by the MLME of the RFD device that will be issued to its next higher layer.

- MLME-POLL service is used when an RFD device wants to request data from its coordinator. Such a request can be sent with or without the destination address information that clarifies whether this request is sent to a coordinator device or not. If the request sending fails due to the CSMA/CA algorithm failure, then an MLME-POLL.confirm primitive will be issued with a status of CHANNEL_ACCESS_FAILURE. If the request is sent through successfully, an acknowledgement message will be needed, otherwise the confirm primitive will set its status to NO_ACK. Once the acknowledgement is received, if the Frame Pending subfield is set to one, the MLME will request the data from the physical layer, otherwise the status of the confirm primitive will be set to NO_DATA. Whatever the result of the Confirm primitive is, it will be issued to the coordinator to poll it for data.

The primitives explained above are slightly adapted by the TKN15.4, later discussed in Chapter 6 to match the design principles of the TinyOS 2. We will discuss those in more details in Chapter 6. PHY features of the IEEE 802.15.4 standard are listed below and will be discussed further wherever related throughout the thesis. Clear channel assessment (CCA) for instance is explained in more details in Section 1.7.1.1.

PHY features

- Activation and deactivation of the radio
- Energy Detection
- Link Quality Indicator
- Channel selection
- Clear Channel Assessment (CCA)
- Transmitting and receiving packets across the PHY medium

1.7 IEEE 802.15.4 MAC Structure

IEEE 802.15.4 has been the most popular and used MAC platform for most of the MAC proposals in WBAN (Latré et al., 2011; Timmons, & Scanlon, 2004; Li & Tan, 2005;

Lamprinos et al., 2005; Omeni et al., 2007; Li & Tan, 2010) or at least coupled with other wireless technologies such as IEEE 802.11 WLAN as in Chipara et al. (2010) or Ko et al. (2008).One of the most likely reasons of this popularity can be the availability of the standard's implementation in most of the widely used simulation tools for academic purposes and also the IEEE 802.15.4 radio chips on most of the current sensor platforms in use. Figure 1-11 shows the possible standards in the family of 802.15 standards to be used for medical set ups. Although IEEE 802.15.6 is the most appropriate standard to be specifically used for WBAN, IEEE 802.15.4 has dominated the research world, both academic and in many of the current pilot studies or practical set ups. There are a few reasons for such popularity which will be discussed in this section.

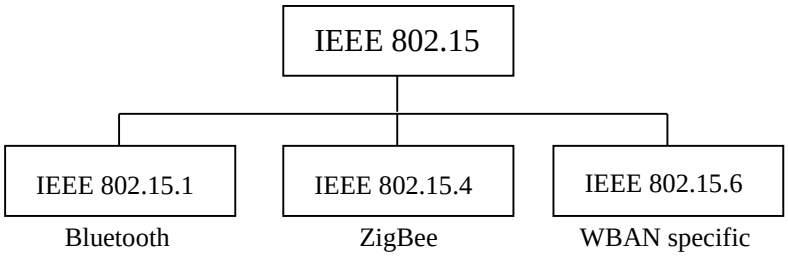


Figure 1-11: Possible standards in the family of 802.15 for medical set ups

Figure 1-11 summarizes the possible standards to be used in a WBAN deployment from the most energy consuming one, Bluetooth, to the most energy efficient one, the IEEE 802.15.6, which is specifically designed for WBANs. Until 2008 there was no standard specifically defined for WBAN (Bilstrub, 2008). Within the earliest and lowest range technologies, which included IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 (often combined with ZigBee), IEEE 802.15.4 presented a lower energy consumption level since it only supports up to a five-meter range of data communication compared to a doubled distance of 10 meters in Bluetooth. Other communication schemes such as Bluetooth could also be considered for the communication links between sensor nodes and the personal server but may lead to more energy consumption. Bluetooth suffers from its protocol complexity, unlike IEEE 802.15.4, which is designed to maximize the energy efficiency at physical and MAC layers (T. J. Dishong et al., 2010). The number of primitives in a Bluetooth protocol stack is much larger than the number of primitives offered by the services in IEEE 802.15.4, which makes it more power consuming to perform on any platform. Bluetooth has a maximum data rate of 24 Mbits/s with a higher startup time and more power consumption during idle times (22 mA compared to 7 mA for IEEE 802.15.4), which reflects the characteristics of the applications it supports. The application area of

Bluetooth is more targeting voice and video, for which it takes advantage of a fast frequency hopping spread spectrum (FHSS) method in order to address high data rates, whilst ZigBee, working in parallel with IEEE 802.15.4, takes advantage of a direct sequence spread spectrum (DSSS) method, which supports short-message applications (Bandyopadhyay, Chaulya, & Mishra, 2009, p. 166). A comparative study (Yan, Zhong, & Jha, 2007) shows why Bluetooth is a poorer choice compared to IEEE 802.15.4 when it comes to wireless body area networks. The higher data rate range support in Bluetooth compared to IEEE 802.15.4 automatically increases the energy consumption rate (Isikman et al, 2011).

Maintaining a simple and flexible protocol is stated as one of the main objectives of IEEE 802.15.4 against the other ones as well as its backward compatibility with devices running on IEEE Std 802.15.4-2003 (IEEE Std 802.15.4, 2003, p. 13). The IEEE 802.15.6 Task Group 6, which introduced the specific standard for WBAN, was proposed in 2007 and released in 2011 and therefore has not yet been completely deployed in many of the current simulation tools nor on the PHY layer of many of the current manufactured chips. Having low data rates rather than other WiFi technologies available in many hospital units is one of the reasons why IEEE 802.15.4 has become so popular for real-time and remote monitoring of vital signs (Chipara et al., 2010). In fact, IEEE 802.15.4 was initially designed for applications that use battery-powered devices, considering that in many cases recharging batteries does not stand as a practical option (IEEE Std 802.15.4, 2003, p. 24). Much of this power preservation can be done by duty cycling the sensor nodes in IEEE 802.15.4, in which synchronized nodes of a personal area network (PAN) or WBAN will turn off their transceivers on a periodic basis and potential sent data can be received by devices periodically listening to the wireless channel. Nevertheless, some very recent comprehensive survey works such as Latré et al. (2011) have come to the conclusion that IEEE 802.15.4 cannot be considered as an ideal choice for all the resource-limited and high-reliability-demanding WBANs in the future. It is expected that in the coming years, research on the MAC protocols specific to WBANs will be centralized around the IEEE 802.15.6 standard as more simulation tools and radio chips provide support for incorporating this newly introduced standard. Having said that, many of the fundamental concepts in the IEEE 802.15.4 standard will still apply to the IEEE 802.15.6 framework and so will most of the protocol-related enhancements done in its MAC and PHY layers, as we will explore throughout this thesis. In this thesis, we base our work on the IEEE 802.15.4 MAC specifications and we benchmark our simulation and experimental results against that of IEEE 802.15.4 for its more comprehensive source of documentation and support tools available at

the time of starting this research in March 2011, but as shown later there is no main hindrance in implementing the concept of our methodology in IEEE 802.15.6 MAC protocol in the future.

Like IEEE 802.15.4, IEEE 802.15.6 defines the PHY and MAC specifications for the standard devoted to only WBAN applications. One of the target issues to be addressed by this new task group was to specify PHY layers that can be completely dedicated to WBAN to decrease the interference caused by devices using the industrial scientific medical (ISM) band. The IEEE 802.15.6 has introduced three different PHY layers of narrowband (NB), ultra wide band (UWB), and human body communication (HBC) layers. The other advantage of IEEE 802.15.6 over IEEE 802.15.4 is its higher security features, offering three different available levels of security to be used based on the application's criteria. The super frame division (as explained for IEEE 802.15.4 in Section 1.7.1) is a bit different but applies the same concept. The contention among nodes for having access to the channel is based on either the CSMA/CA mechanism (described in Section 1.7.1.1) or slotted ALOHA (Abramson, 1970). The ability of contention-free access is provided by two different periods known as random access period (RAP) and exclusive access period (EAP) with each being halved to leave one half for high priority data to transmit. IEEE 802.15.6 has provisioned three different access trials for the applications using this standard. These three groups are: 1) random access scenarios, 2) improvised and contention-free access scenarios, and 3) scheduled access and variants (Kwak, Ullah, and Ullah, 2011). The CSMA/CA mechanism of the 802.15.6 standard differs from that of IEEE 802.15.4 in the sense that the contention window (CW) no longer has a counter nature. In IEEE 802.15.4, CW acts as a counter to keep account of the number of times the channel is sensed as idle by a node, whereas in IEEE 802.15.6 the CW is simply a window of time for which it takes two different default values, CW_{min} and CW_{max}. Scheduled access in IEEE 802.15.6 is provided by the managed access period (MAP), which corresponds to the CFP of the IEEE 802.15.4. If the frequency band in use is a narrowband, the access mechanism for the contention access will be a CSMA/CA, whereas for an ultra-wide band frequency band the contention access mechanism is the slotted ALOHA. One of the main capabilities introduced in IEEE 802.15.6 is the provisioning of the emergency traffic for which the EAPs are introduced and differentiated from the normal traffic in the network which would go through during RAP and CAP periods.More detailed description of the IEEE 802.15.6 is out of the scope of this research work as we base our work on the IEEE 802.15.4 standard.

There are two types of devices with different levels of resource availability that operate based on the IEEE 802.15.4 standard. Reduced Function Device (RFD) refers to sensor nodes or devices that do not have access to all the primitives of the standard and are lower in their resource availability, such as battery power and range of operation. A Full Function Device (FFD) is capable of performing all the primitives of the standard while being richer in resources and either it is main powered or battery powered. It is assumed that its battery can be recharged at any time with no restrictions. In a typical body area network, a full function device is normally assumed as a smart phone or PDA (also known as Body Control Unit or BCU in some research papers (Schmidth et al., 2002), carried by the patient or at least located in the same place as the patient at all times if the application is designed for a continuous monitoring of the subject. The topologies supported in this standard are star and peer-to-peer topologies. Both RFD and FFD devices have a unique 64-bit address to be identified with.

The preferred topology in this research is star, which offers some advantagesin a beacon-enabled mode of access, later described in Section 1.7.1, and used in our proposed method. No concern of beacon messages to collide with each other exists in a star topology as opposed to a multi-hop network (Sun, Sun, & Zou, 2009), plus in a single-hop star topology less delay and consequently less energy consumption is experienced; this is because of the inherent asymmetry that exists between the sensor nodes as RFDs and the coordinator as an FFD. Peer-to-peer communications are more intended for applications that provide a greater coverage for that target application, whilst in a typical WBAN, which is a network of sensors on the body, a big coverage is not needed. In a star topology, a coordinator will have the responsibility to initiate the transmission by first sending a beacon message (discussed later in the next section) to all the nodes residing in the network to inform them about some configurations for the next beacon interval, which is the next round of transmission for all the sensor devices in the network. Direct communications between individual nodes of a cluster, which in this thesis is a single WBAN, is not possible. There are some other works in the literature that introduce a multi-hop wireless body area network such as CICADA (Latré et al., 2007), which is an improvement over WASP protocol (Sathyan, Humphrey, & Hedley, 2011). In CICADA, the authors claim multi-hop communication to be a better choice for WBAN because of the high propagation loss around the human body, which lessens the quality of signals in direct communication links between a sensor and its coordinator. But even in the IEEE 802.15.4 standard documentation it is recommended to use star topology for personal health care applications and a mesh topology is more suggested for applications such as industrial control

and monitoring, asset and inventory tracking, or intelligent agriculture (IEEE 802.15.4 Std, 2003, p. 14). In a star topology, a network is first formed by the coordinator device (the only FFD present in the network) when the device first gets activated. The other RFD devices will then join this network if allowed by the coordinator device and will synchronize their radios to each other by the aid of the information carried in the beacon message broadcast to them via the coordinator.

The IEEE 802.15.4 operates on two different modes, beacon-enabled mode and non-beacon-enabled mode (as shown in Figure 1-12) and discussed in Sections 1.7.1 and 1.7.2. Figure 1-12 shows the hierarchy of them in the standard. The main difference comes from the synchronization capability of the beacon-enabled mode and the ability of the MAC protocol to perform in a slotted manner in which time is defined in terms of super frames and the backoff time of each node influences the backoff times of the other nodes. In the non-beacon-enabled mode, however, each node's backoff time is completely independent of the backoff time of the other nodes (Mahalik, 2007, p. 29). Slotted carrier sense multiple access with collision avoidance (CSMA/CA), as used in the beacon enabled mode of access, encourages a more efficient use of backoff times by means of a channel clear assessment (CCA) mechanism at the PHY layer to more reduce the probability of collisions. No such resolution exists in the non-slotted mode of access later described in Section 1.7.2.

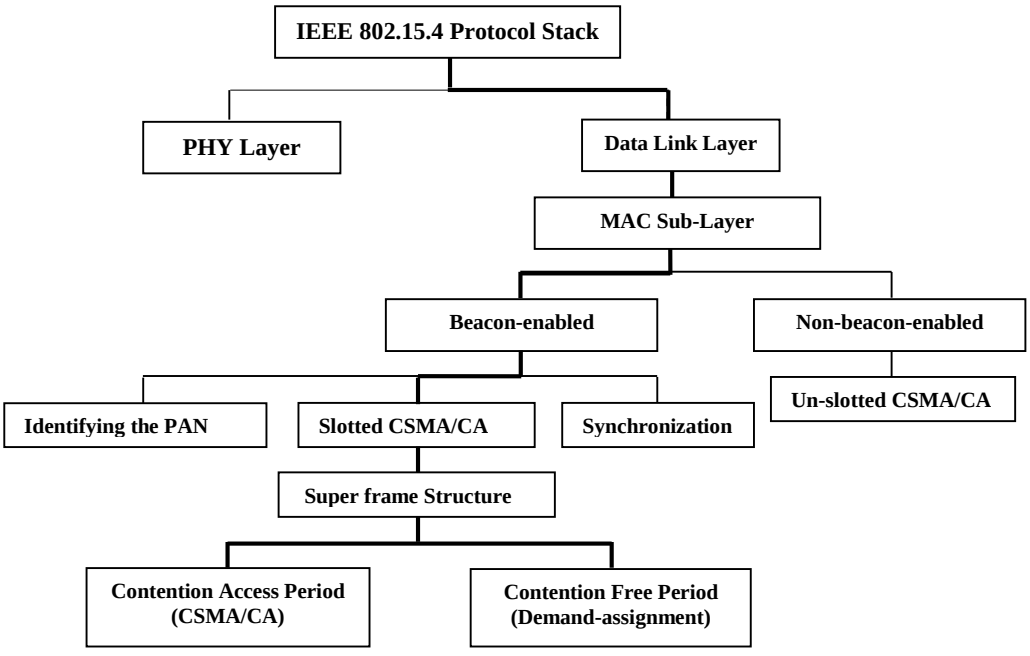


Figure 1-12: Different Modes of Access

1.7.1 **IEEE 802.15.4 Beacon-enabled Mode of Access**

Almost all the works reviewed in the literature are based on the beacon-enabled mode of access. In Koubaa, Alves, & Tovar (2007), pp. 40–42, the two main driving factors behind its selection by most researchers have been identified as a more advanced set of features that gives it more flexibility, synchronization capabilities, and also the guaranteed time slots (GTS) option for high priority and delay critical data packets. In IEEE 802.15.4, three main services that beacons provide in a network are: need for synchronizations, support for low latency devices, and network discovery; however beacons are mostly renowned as a means of synchronizing the sensor devices to their coordinator and carry valuable information to be broadcast to all nodes. In fact, synchronization is the main difference between the two different modes of access discussed in Sections 1.7.1 and 1.7.2, and only applies to a beacon-enabled mode where nodes have to be synchronized at the beginning of each beacon interval to the end of the super frame described later in this section. A coordinator would learn about the PAN (in our case, a WBAN) it coordinates by sending out a beacon frame. A beacon frame is configured by the coordinator device at the beginning of each super frame and is the first frame to be sent to start off the whole beacon interval in which all the transmissions during that beacon interval will take place. The variability of the configuration options that a beacon frame carries gives it huge potential for the researchers to investigate fine tuning the existing protocol in accordance with their own application demands, which is mostly done through dynamically adjusting the lengths of its two different segments, contention-based and contention-free periods.A discussion of some of the related works regarding to the dynamic adjustment of a super frame appear in Section 3.2.1 of this thesis. Other reviewed works in the context of beacon-enabled mode focus on the dynamic methods that would take advantage of other network parameters of a slotted CSMA/CA mechanism, which are discussed extensively in Section 3.1.

For star topologies and where a centralized network performance is preferred, the beacon-enabled mode offers some controlling options by means of its beacon message broadcast. Beacon-enabled mode takes advantage of both CSMA/CA in its CAP (Contention Access Period) and a schedule-based access in its CFP (Contention Free Period) as two main segments of a super frame to transmit and receive data for the duration of a beacon interval. The beacon packets and the acknowledgement packets are not sent based on a CSMA/CA algorithm. The CSMA/CA mechanism in IEEE 802.15.4 standard suffers from a hidden node problem, as the Request to Send/Clear to Send (RTS/CTS) handshake no longer exists to eliminate the

overhead caused by control packets as a means to save energy. In Koubaa, et al. (2007), it is stated that the absence of the RTS/CTS messages does not make any difference to having collisions, as data packets are so small in size and almost as big as RTS/CTS control packets. They also declare RTS/CTS messaging as a very energy-consuming task, which does not comply with the requirements of a WBAN structure. There have been many research works in the literature to compensate for this defect though, such as Hwang, Sheu, Shih, & Cheng (2005) and Koubaal, Severino, Alves, & Tovar (2009), which aim at decreasingthe collision probability in the active period of the super frame, but this is out of the scope of this research. However it is believed not to be problematic for small scale networks (Koubaal et al., 2009) such as a WBAN with only a few nodes deployed on the body and so is not considered as an issue in our BAN implementation. The guaranteed time slot (GTS) capability in CFP provisions the delivery of time-critical data packets normally flagged as high-priority data which would always be transmitted at the end of a super frame before the inactive period with a maximum of only seven time slots to be allocated for GTS purposes. A few devices may have a chance to transmit their data during CFP of the super frame by means of the GTS time slots but one has to ensure its transmission ends before the other one starts and all allocated GTS transmissions must be finished before the end of the current CFP. The CAP and CFP together construct a super frame in which all the communications take place before nodes enter their inactive period in the beacon interval. In fact, every current transaction must have been completed before nodes compete with each other again in the next beacon interval. Figure 1-13 (Sahinoglu & Guvenc, 2011, p. 143) illustrates a beacon interval, which consists of both super frame duration (SD) and an inactive period. A super frame is normally bounded between a beacon frame residing in its first time slot and an inactive period, which is optional to use. If an inactive duration (in which the radio goes off) is not provided, a super frame will end by the last time slot in its CFP. CAP takes advantage of the CSMA/CA mechanism as the standard's resolution to provide more reliable and collision-free communications between the sensors and the coordinator device. No collision concern is associated with the contention-free period of the super frame as it incorporates the concept of GTS, mentioned earlier, on a pre-allocation basis to nodes with more critical data to send. The order of the sent and received control and data messages differs depending on which device is sending and which device is receiving. If an RFD device tends to send a data frame to its coordinator it starts by first listening to receive the beacon frame so that it will be able to synchronize itself to the structure of the super frame commanded by the coordinator. After being synched to its coordinator and the network it belongs to, the device will then try sending its data over the channel using CSMA/CA, which

will be discussed later in Section 1.7.1.1. An optional acknowledgement mechanism would confirm the end of a successful transmission, which is sent from the coordinator to the RFD device. As shown in Figure 1-13, the length of a super frame is divided into 16 equal time slots including some guaranteed time slots (GTSs) that are reserved for time critical applications.

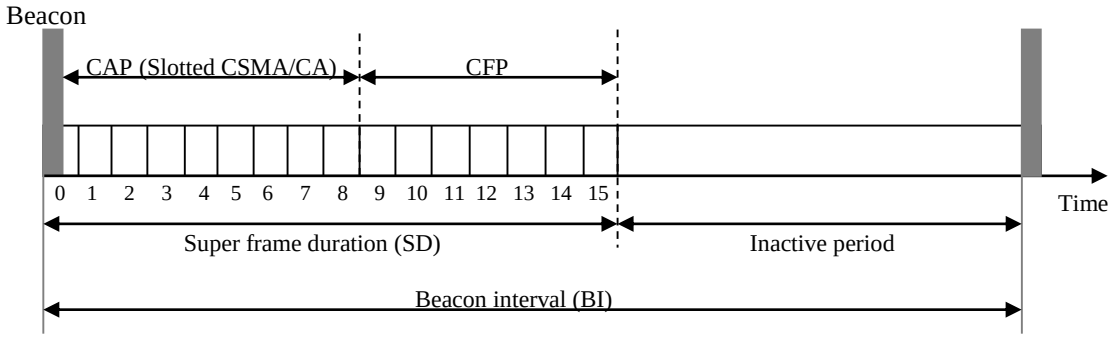


Figure 1-13: Super frame structure in a beacon interval

The elements of a super frame in Figure 1-13 can be calculated as below:

(Beacon Interval); BI=a*BaseSuperFrameDuration*.2^{BO} for $0 \leq BO \leq 14$ (1)

(Super Frame Duration); SD= a*BaseSuperframeDuration*.2^{SO} for $0 \leq SO \leq BO \leq 14$ (2)

The inactive or sleep period is actually optional and depends on the length of the super frame defined by the super frame order (SO). If SO is less than the beacon order (BO), then an inactive period definitely exists, and if SO equals BO there will be no inactive period in the beacon period. If SO equals 0, the super frame finds its minimum duration, which is 15.36 milliseconds (Koubaa et al., 2007, p. 28), considering that each super frame consists of 960 symbols and each symbol equals 4 bits and therefore, with a maximum data rate of 250 kbps, each time slot would have duration of 0.96 milliseconds. If both BO and SO values equal 15, no beacon transmission exists and the standard enters its non-beacon enabled mode of access, as described in 1.7.2. It is also the responsibility of the MAC layer to compensate for any clock drift that may happen between the RFD device's clock and the FFD device's clock. A clock in any device can be defined as a counter that defines time for that device and is implemented using an internal oscillator. Over time, the synchronization between the RFD and FFD clocks will be lost, which will lead to the clock drift phenomenon. Preventing clock drift is crucial for any MAC protocol to maintain its functionality since, for example, for an RFD to enter into its receive mode at the right time it is important to receive the beacon from the FFD device

correctly (Muqattash, 2012). There is a minimum value considered for the length of the CAP period in terms of symbols less than which a CAP duration cannot be but it will grow or shrink occasionally when a CFP has to accommodate more time slots in some cases ("IEEE 802.15.4 Std", 2003, p. 168). If a transmission cannot be completed before the CAP ends in the super frame, its transmission will be deferred to be completed in the next beacon interval. The next section will describe the contention mechanism among nodes in CAP and also how they can have time critical requests for their transmissions in CFP where no contention exists.

1.7.1.1 CSMA/CA Mechanism in Beacon-enabled Mode of Access

We follow the flow chart in Figure 1-14 (IEEE 802.15.4 Std, 2003) to discuss the steps of a single transmission to be made in the context of the CSMA/CA mechanism in the standard's beacon-enabled mode. CSMA/CA is referred to as a mechanism to improve the probability of successful transmissions in the standard descriptions (IEEE 802.15.4 Std, 2003, p. 23) in which different concepts contribute to delivery of such a goal. Two clear channel assessments (CCA) are done in this mode, which are carried out at the end of the backoff time. The channel has to be sensed idle for two CCAs and a random backoff time will be assigned if the node fails to do so (Pollin et al., 2008; IEEE 802.15.4 Std, 2003). Each device maintains three variables when contending to have access to the medium in slotted CSMA/CA mode: NB (number of times the node backs off), BE (the backoff exponent), and CW (length of contention window). CW actually counts the number of times the channel has to be sensed idle, which is set to two by default, and is also known as CCA counter in some papers (Gao, Hu, & Min, 2008). The CCA has to be done by the physical layer and the result of its CS is reported back to the MAC layer. However, BE denotes the waiting time or number of backoff times a node has to wait and is set to min(2, *macMinBE*) if BLE is on, and NB is set to 0 as the default value at the beginning of the transmission.When the node waits for its first number of unit backoff times, it performs the first CCA and if it is successful, the CW decrements by one, meaning that the first CCA has not been interrupted.

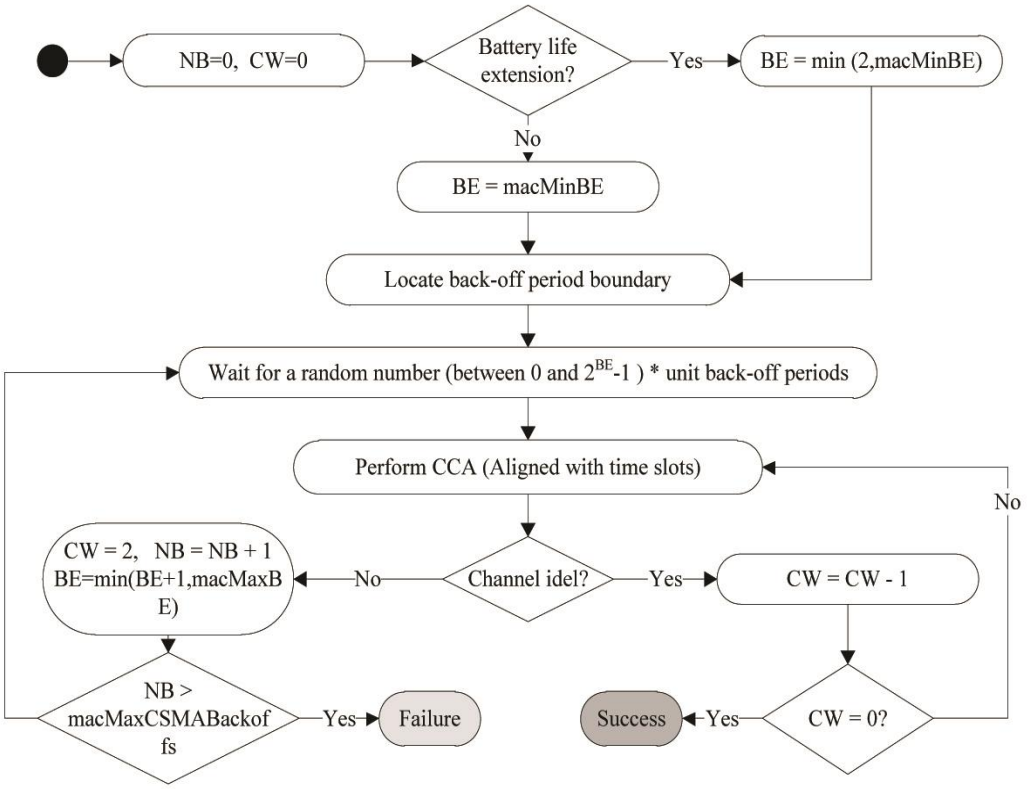


Figure 1-14: CSMA/CA Mechanism in Beacon-enabled Mode of Access

If, however, the channel won't be sensed idle for the entire duration of the first CCA, the NB and BE are incremented by one while CW still remains the same. The process continues until two consecutive CCAs are achieved successfully, meaning that the channel has been sensed idle for the duration of two CWs. A failure is reported if NB reaches its defined maximum (*macMaxCSMABackoffs*), which is listed as an attribute and not a constant in the IEEE 802.15.4 standard, with a default value of four as shown in Table II. This means that if a node has had four unsuccessful trials in performing its two consecutive CCAs to have access to the channel, it will experience a transmission failure, but will still have a few more chances to retry sending its packets, which is attributed in *macMaxFrameRetries* attribute also shown in Table II. The default value for this attribute is three, which allows the node to retransmit its packet three times at most. When the "Battery Life Extension" (BLE) option is set to zero, BE equals *macMinBE*, which is the same initiative as the un-slotted CSMA/CA described later in Section 1.7.2. In this case, the MAC sub-layer has to finish all the transactions before the end of the current CAP. If the number of backoff periods is greater than the remaining backoff periods of the CAP, it has to pause the backoff countdown and defer it to the CAP of the next super frame to be resumed from where it had been paused (IEEE 802.15.4 Std, 2003, p. 171). Where the

BLE option is set to one, there will be a limitation on when the backoff can occur. In this case, the backoff counter can only start on the first *macBattLifeExtPeriods* full backoff periods after the Inter Frame Space (IFS) following the beacon. The frame transmission will be started on one of the first *macBattLifeExtPeriods* full backoff periods after the IFS following the beacon. The BE parameter will be initiated to the lesser of (2, *macMinBE*) in this case. Table II shows the constant values and attributes related to the CSMA/CA mechanism illustrated in Figure 1-14 and also the super frame structure illustrated in Figure 1-13. In the IEEE 802.15.4 standard, the prefix for denoting a constant value is "*a*" and attributes are prefixed with "*mac*", which can be configurable but have their own default values defined in the standard. These attributes in the MAC PIB database of attributes are briefly discussed in Section 1.6 and are required to manage the MAC sub layer of a device (IEEE 802.15.4 Std, 2003, p. 160). Some of these attributes are read-only, meaning that they can only be set by the MAC sub-layer and some are both readable and writable by the next higher layers. The read-only attributes can be read by higher layers by means of MLME-GET.request primitive, which is briefly discussed in Section 1.6 and depicted in Figure 1-10.

Table II: Constant values and attributes in IEEE 802.15.4 specifications

Constant	**Description**	**value**
aBaseSlotDuration	The number of symbols forming a super frame slot when super frame order is 0.	60
aBaseSuperframeDuration	The number of symbols forming a super frame when the super frame order is 0.	aBaseSlotDuration* aNumSuperframeSlot
aMaxLostBeacons	The number of consecutive lost beacons that will cause the MAC sub layer of a receiving device to declare a loss of synchronization.	4
aMinCAPLength	The minimum number of symbols forming the CAP. This ensures that MAC commands can still be transferred to devices when GTSs are being used. An exception to this minimum will be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance.	440
aNumSuperframeSlots	The number of slots contained in any super frame.	16
aUnitBackoffPeriod	The number of symbols forming the basic time period used by the CSMA/CA algorithm.	20
Attribute	**Description**	**Type/Default**
macBattLifeExt	Indication of whether BLE, through the reduction of coordinator receiver operation time during the CAP, is enabled. A value of TRUE indicates that it is enabled. In Figure 1-14 it is shown how this attribute affects the backoff exponent in the CSMA/CA algorithm.	Boolean/FALSE
macBattLifeExtPeriods	The value of this attribute is PHY dependent and is the sum of three values described in (IEEE 802.15.4 Std, 2003, p. 162). In the BLE mode, it is defined as the number of backoff periods during which the receiver is enabled after the IFS following a beacon.	Integer/6-41

macBeaconOrder	Specification of how often the coordinator transmits its beacon. If *BO*=15, the coordinator will not transmit a periodic beacon. The relationship between *BO* and beacon interval is explained in Section 1.7.1.	Integer/15
macGTSPermit	TRUE if the PAN coordinator is to accept GTS requests. FALSE otherwise. In our simulation experiments in Chapter 5 this capability of the MAC layer is ON, meaning that we take advantage of both TDMA and CSMA access modes.	Boolean/TRUE
macMaxBE	The maximum value of the backoff exponent, BE, in the CSMA/CA algorithm explained in Figure 1-14.	Integer/5
macMaxCSMABackoffs	The maximum number of backoffs the CSMA/CA algorithm will attempt before declaring a channel access failure. Detailed explanations are brought in Section 1.7.1.1.	Integer/4
macMaxFrameRetries	The maximum number of retries allowed after a transmission failure.	Integer/3
macMinBE	The minimum value of the backoff exponent, BE, in the CSMA/CA algorithm explained in Figure 1-14.	Integer/3

1.7.2 **IEEE 802.15.4 Non-beacon-enabled Mode of Access**

In non-beacon-enabled mode, the nodes simply transmit based on an un-slotted CSMA/CA (Sahinoglu &Guvenc, 2011; IEEE 802.15.4 Std, 2003). As its name also implies, there are no beacon frames in this mode of access, nor do any super frames exist, therefore the values of BO and SO earlier discussed in Section 1.7.1 for the super frame structure are equal to each other and to the value of 15. There will be no use of CSMA/CA mechanism except for the transmission of the acknowledgement messages and there will be no contention-free transmission possibility such as GTSs. In this mode, the protocol makes the nodes sense the channel for a random backoff time and lets them transmit if the channel is sensed idle for that duration, and if not the backoff time has to be extended. In un-slotted CSMA/CA, the backoff time of one device does not depend on the other device's backoff, meaning that their backoff times are not synchronized. The variation of the slotted CSMA/CA mechanism in Figure 1-15Figure for the un-slotted CSMA/CA is shown in Figure 1-15 below.

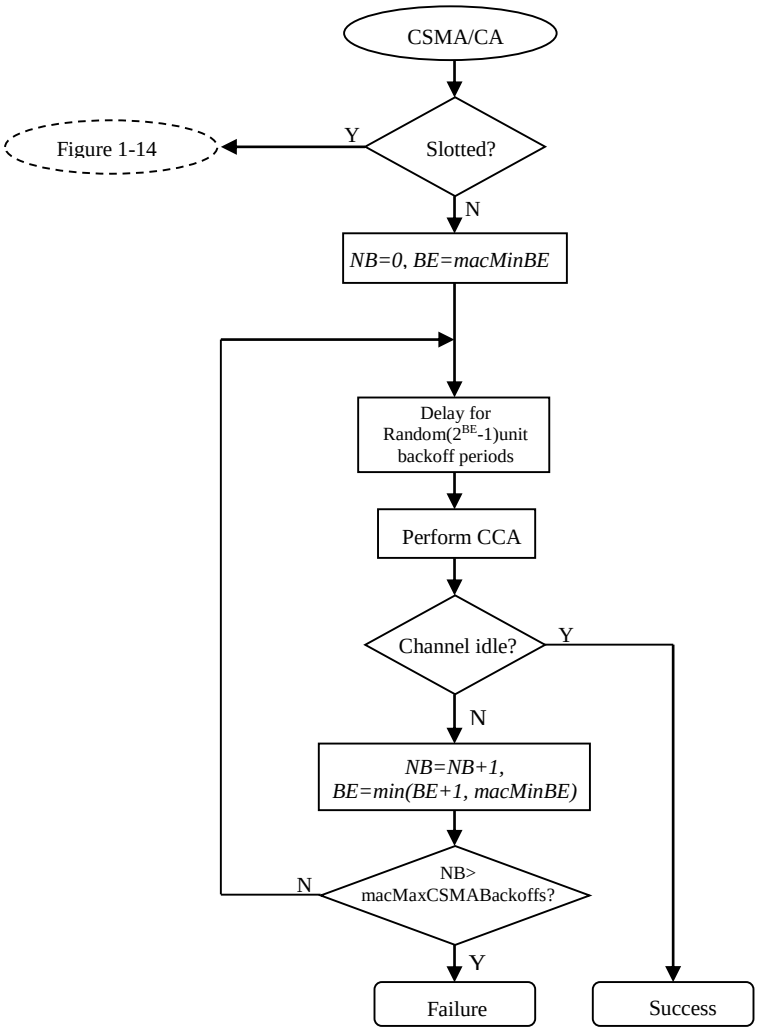


Figure 1-15: Unslotted CSMA/CA

In terms of the variables used, CW is only used in the slotted CSMA/CA and is not used in un-slotted CSMA/CA, as we do not have the concept of two consecutive and successful CCAs as a condition of transmission in the latter. Only one CCA exists in un-slotted CSMA/CA and, if it does not indicate a clear channel, the node enters its backoff time period, otherwise it immediately transmits its data over the channel. Further discussion of the non-beacon-enabled mode of access is beyond the scope of this research. In Sections 1.8 and 1.9 we look at the merits and demerits of each of the schedule- and contention-based MAC approaches specifically for WBAN scenarios and especially in healthcare applications. We then continue with the problem that motivated our research in Chapter 2.

1.8 Synchronous Medium Access Control Protocols for WBAN

Sections 1.8.1 to 1.8.2 give some insight into two different types of schedule-based MAC techniques, namely time division multiple access (TDMA) and polling based access schemes. Some related examples of implemented MAC protocols from both categories are discussed.

1.8.1 Schedule-based (TDMA) MAC Protocols for WBAN

TDMA-based MAC protocols have been identified as the most suitable MAC schemes for WBANs as stated in Mahtab Alam et al. (2012), Marinkovic, Spagnol, & Popovici (2009), Tselishchev et al. (2011), Ghildiyal et al. (2011), and Marinkovic, Popovici, Spagnol, Faul, & Marnane (2009), but they are also said to be power consuming and complex for their synchronization needs. There has been no pure TDMA-based MAC protocol that could solely satisfy all the requirements of a WBAN. In all the reviewed works collected here, the TDMA-based MAC is coupled with one or two more techniques to become absolutely suitable for a desirable MAC performance for WBAN. In fact, the type of MAC protocol chosen so much depends on the current traffic information of the network, which is a real-time issue. In other words, if the traffic goes periodic for a while then a TDMA-based MAC outperforms CSMA, but if the traffic is chaotic then CSMA has a better performance due to its flexibility and non-pre-defined nature. In Marinkovic, Spagnol, & Popovici (2009), however, it is claimed, with reference to the fixed nature of the body area networks, that a TDMA-based MAC outperforms a CSMA approach as scalability no more poses as a problem in a WBAN. CSMA's main advantage over the TDMA schemes is its adaptability to real-time changes in the network topology, which does not happen under a fixed and predefined topology with only a few nodes. The complexity of synchronization of nodes in TDMA approaches, therefore, is said not to be harsh or a hindrance in WBAN since synchronization only becomes problematic in dynamic topologies (Marinkovic, Spagnol, & Popovici, 2009). In general, TDMA approaches suffer from more overhead when distributing the slot assignments to sensor nodes of the network using control packets. Therefore the authors in Marinkovic, Spagnol, & Popovici (2009) use a predetermined slot assignment as a simpler strategy that only applies to some rigid structures. The data rate array defined for the sensor nodes of the WBAN is considered constant, which is not always the case in medical monitoring applications. Although their approach works, it lacks a realistic framework or it would only suit certain healthcare applications in the future. In Zhisheng & Liu (2011), the overhead of the synchronization scheme was decreased by a novel optional synchronization technique for their TDMA-based MAC protocol. Synchronization in

Zhisheng & Liu (2011) only happens if the clock drift becomes large enough to produce slot overlapping. In their approach, a beacon frame may not be sent at the end of each beacon interval and broadcasting a beacon frame only depends on detecting abnormalities in the received data at the coordinator during the previous beacon interval. This approach might have been designed to handle the emergency data as also described in Section 3.2.2, but it also reduces overhead. Other enhanced versions of the TDMA approach have been discussed in the literature to propose solutions to the lesser flexibility of the TDMA techniques in handling instantaneous traffic. A variable TDMA scheduling technique for example is used in Tselishchev et al. (2011), which is an opportunistic approach rather than a predetermined one by considering the instantaneous conditions of the channel. Opportunistic approaches are, however, deemed to be very power-consuming for body area networks as they require continuous reporting of the channel state (Tselishchev et al., 2011). The main motivation for their proposed MAC is, however, the"high volatility" of wireless links in a typical WBAN structure that restrains the pure TDMA solutions to be an absolute remedy due to their static scheme. The existing volatility causes the respective links between a sensor and its coordinator in its pre-allocated time slot to be unstable at the time of transmission due to the channel's instantaneous bad conditions, which means the data cannot always be safely transferred on their dedicated time slot and depend highly on the channel state at the time of transmitting. In Tselishchev et al. (2011), a slot allocation is only done when the channel condition guarantees a successful transmission and therefore, as opposed to a pure TDMA approach, it is not predefined any more, and considers the real time channel conditions. One drawback of their approach could be a limit slot availability that it provides, which is only one slot, which may not be always enough for different traffic loads.

The concept of a TDMA-access mode has always been to provide a capability to the protocol to address time-critical and high-priority data whenever it is requested, but it could never represent the flexibility of a CSMA-based one in sudden and high traffic situations.This context awareness in handling instantaneous traffic is also proposed in Zhisheng & Liu (2011), where slot allocation can be dynamically changeable by giving a higher sample rate to emergency data. Some works such as Omeni et al. (2007), in contrast to Tselishchev et al. (2011), deem this flexibility to be a second concern for WBANs and suggest a centrally controlled sleep/wake up schedule to perfectly match the energy constraints of such networks. Idle listening, overhearing, and collisions could be eliminated from WBANs by means of a CCA/TDMA access scheme as reported in Omeni et al. (2007), where the traffic is centrally

managed among all the sensor nodes by the aid of their master, which is the coordinator. The master node will allocate the slave nodes their specific time slots for transmission, which would not necessarily be the same slots in each transmission time. Their method adds more complexity to the coordinator node as it deals with scheduling all the slave nodes but it is believed not to be a problem for the rich resource master device. There are, however, some limitations to their work that make it only suitable for some specific applications; the network in their work is assumed not to always act immediatelyin response to a certain change, which is an important factor to be considered in healthcare applications. One drawback of the IEEE 802.15.4 MAC approach mentioned in their work is the ability of a slave node to initiate the communication in a non-beacon-enabled mode, which is not even a major concern as almost no work in the literature has considered a non-beacon-enabled communication mode for data transfer. Plus in a WBAN no new link establishmentis normally considered as there is no node joining the operating network as the network is too small and has a pre-defined structure and once deployed is assumed to work for at least a couple of days without any interruption.The "link establishment" process explained in the paper can account for the "association process" that exists in the IEEE 802.15.4 MAC with the difference that there is no sleep-time distribution by the beacon frame in the IEEE 802.15.4 standard; the standard however does specify the length of each beacon interval (the active and sleep periods together) by means of the BO and SO parameters. As declared in the standard, the coordinator (master node) maintains the timing of both the incoming super frame (the currently received one) and also the outgoing super frame (the next transmitted one), which is handled by the start-time parameter in the beacon frame. The length of the active period in a super frame is determined based on the number of allocated GTS slots and also the MAC header containing the MAC payload for the CAP. The rest is allocated for the sleep (or inactive) period. All the slave nodes in the standard do stay synchronized with their master when they receive the beacon frame, of course. They acquire this synchronization by repeatedly sending synchronization requests until they receive the beacon frame from a master node with a PAN coordinator ID. The number of trials for seeking the beacon frame from a PAN coordinator is, however, limited to a specific number in the standard. Once the beacon frame is received, the device will be synchronized to the super frame structure that it describes and the data that it has will be sent using the slotted CSMA/CA procedure. The common sleep schedule described between the slave and its master in Omeni et al. (2007) is then refined by a Wakeup Fallback Time (WFT) concept to compensate for the interruption their "alarm condition" algorithm causes. The alarm condition was introduced to allow nodes with some priority data to communicate with the master node before their wakeup

time. One major problem this solution would impose to the network is the latency that it produces, especially if several WFT happen at nearly the same times. Their work is however one of the few proposed MAC protocols that has been practically deployed and tested on a real sensor platform. With having produced more chances of the slave nodes to spend their time in sleep mode, in their method they managed to have more energy savings. However, the results demonstrated are not contrasted against IEEE 802.15.4 performance, plus only the temperature sensory data was used to validate how energy consumption relies on the length of sleep periods and the number of retransmissions. The research therefore leaves a huge space to investigate the applicability of their CCA/TDMA approach for all the different applications and traffic patterns existing in WBANs. Whilst in Omeni et al. (2007) a centralized controlling is introduced by the master node for synchronization of the slave nodes, having a centralized resource management for assigning the time slots to the slave nodes is not considered a perpetual solution in Boulis & Tselishchev (2010) due to deep fades that signals in a WBAN undergo. Control messages that make synchronization possible for a TDMA-based MAC protocol may be lost when the signal strength drops below the receiver's sensitivity, making the MAC protocol vulnerable to losing a slot assignment for a particular node (Boulis & Tselishchev, 2010). This would question the 100% applicability of the alarm-condition algorithm introduced in Omeni et al. (2007). The time-varying nature of wireless channel in a WBAN is also mentioned as a hindrance to a flawless implementation of a pure TDMA-based MAC (Liu et al., 2011). However, they later proposed an enhanced TDMA-based MAC (Zhisheng et al., 2012), where they investigate scheduling the slots for each node in the next super frame based on channel estimation and claim it is a better technique than the blind interleaving of the transmissions in their earlier hybrid TDMA-CSMA approach (Liuet al., 2011). In Zhisheng et al. (2012), the two main reasons behind the inefficiency of a pure fixed-assignment approach for WBAN MAC design were reported as the inability to deal with dynamic traffic and time-varying conditions of the environment (as also reported in Tselishchev et al. [2011] and Boulis & Tselishchev [2010]), but the drawback of coupling it with a CSMA-based technique would be the inefficiency in channel utilization and also energy in high data rates. In Zhisheng et al. (2012), the number of slots to be assigned in the next super frame is determined based on the QoS requirement of each node in the current super frame to guarantee energy-efficient data delivery. Based on the above reviewed literature, the drawbacks of a pure TDMA approach for wireless body area network MAC design can be classified as:

- Synchronization control messages' vulnerability due to deep fades

- Inflexibility in handing the instantaneous changes in contextual traffic
- Synchronization overhead

In Li & Tan (2010), a solution to the traditional time synchronization is proposed. Their work is novel in the sense that they are probably the first to exploit biomedical signs in a BSN to do time synchronization. In traditional TDMA-based BSNs, the time synchronization is done periodically and by the network coordinator, which wakes up the nodes periodically to make them exchange their timing information with each other (synchronization). Since the synchronization involves extra energy consumption, they try to do it without turning on the sensors' radios periodically. What could possibly replace it? A natural (inherent) feature from which a time slot can be derived where all the nodes exchange their timing information in that time slot. The time slot is perceived to be based on the heartbeat of the human body (the peaks of the wave form). This means the timing information is in the sensor's sensory data and not sent by the coordinator. The topology is star.

1.8.2 **Polling-based (on-demand) MAC Protocols for WBAN**

A polling-based technique is a bit different from scheduled access MAC protocols in the sense that, as opposed to the scheduled access mode, no pre-defined slot allocation is done and the coordinator might decide, in the current frame, to allocate some additional time slots to a node upon request. Boulis & Tselishchev (2010) studied the two MAC approaches commonly known as:

- Contention-based and
- Polling-based.

Why polling access is not an ideal access mode for a WBAN is described as the variability on the channel because of the interference and fades it undergoes. Boulis and Tselishchev show that a combination of these techniques will work the best. Since the channel model changes all the time, they tried to characterize it by studying the RSSI (signal strength). The distribution and duration of fades in the channel are captured and they use their own channel models in their simulations. The traffic type in a WBAN is reported to be sparse and emergency. The ("SMA-WiBAN", 2010) is used as their baseline MAC approach. They refer to how the baseline MAC assigns "scheduled slots" and "polled slots" during a time frame. The scheduled

slots are the ones that get permission to access some slots freely for the “next” frame and the polled slots are those that can be assigned specific slots during the “current” frame so it is called “immediate polled access”. The authors claim that “SMA-WiBAN” (2010) does not describe HOW it assigns the polled slots so they take advantage of the field “moreData” in the MAC header, and when a node is currently transmitting something, in its lastTX slot it has to set its more-data flag, and when the hub receives that it will set its future poll flag to tell the node (by its ACK) that it has some free time slots starting at time t_1, for example. In this way, the node will wake up at that time to have only one free polled time slot. Next time, this time slot will be the last TX slot of the node, so the chances of a node getting a free polled slot increases. The data packet is 128 bytes and it needs one milli second to be transmitted. The time needed for data + ACK + transition times = 1.16 milli second. Twenty-one combinations of contention and polling were tested in their paper and the results presented are very interesting to see the characteristics (reactions) of different access modes i.e. contention and polling. As depicted in their diagrams, the contention-access mode does not act well in high loads, even if its length increases in slots. In a high load, for example 140 kbps, it is better to have a contention period of least length, for example two slots, and assign the rest of the slots for polling-based access. It is deduced that the current implementation of the baseline MAC is that, inorder to give poll slots to the nodes, it first sends a future poll message via ACK that just says when it will send a future free time slot/slots. But it is suggested in their paper that the future poll message could actually both report the free slots and assign them on the fly without asking the nodes to get up at a poll time slot to take it. Some interesting points in their result presentation are that more interfered packets are observed as the contention length increases but when using polled access or contention-free access, fewer collisions happen because the slots are pre-assigned, but then the interesting point is that if a packet gets lost during a contention-free period, since the packets are sent back to back, the probability that a retransmission attempt fails is higher than a retransmission attempt in contention-based access. Because in contention-based access a retransmission is scheduled by random backoff times, for lower rates the contention access is excellent. Boulis and Tselishchev (2010) show that a shorter contention period and a larger polled-based access period would lead to a better performance of a MAC protocol in terms of delay and, in terms of energy, polling-based access dominates contention-based access. However, the evaluation of both MAC schemes relies more on what the specific needs of an application are; for example, for lower data rates of around 40 packets per second, a contention-based access scheme works better in terms of the packet delivery ratio but it does some harm to the energy consumption compared to polling-based access. So there will be a trade off

between “better delivery ratio and less delay” and “energy consumption”. In Liuet al. (2011), a combination of TDMA (schedule-based), and polling-based access is presented. The polling approach is to handle the emergency traffic from a node if the demanded slots have not been considered in the scheduled slots of the TDMA access. The polling scheme is designed to support time criticality by gathering information from nodes upon needing extra slots to send their data. The slot assignment information is encapsulated in the poll message sent by the coordinator to the sensor node before the current frame finishes.

1.9 Asynchronous Medium Access Control Protocols for WBAN

1.9.1 Contention-based (Slotted CSMA/CA) MAC Protocols for WBAN

Contention-based access has high latency and energy consumption due to the high contention it especially causes in high traffic loads among the sensor nodes. However, the contention is less in a small-scale WBAN structure. The predefined nature of the physiological sensor nodes deployed on the body in terms of the required data rate (according to what vital sign they monitor) also helps some of the drawbacks of contention-based techniques to become less visible. There also might be some positive points of CSMA techniques such as scalability, which does not apply to a predefined and small-scale one-cluster star topology. CSMA based techniques are the best to remedy the high traffic load when increased network activity is imposed by certain applications.

Contention-based MAC approaches have also received attention to be used in WBAN scenarios to more efficiently handle the sporadic and emergency traffic that is inherent in WBAN applications (Boulis & Tselishchev, 2010). In fact, the IEEE 802.15.6 baseline MAC, which is designed specifically for WBAN scenarios, encompasses the three modes of access in its protocol body; schedule-, polling-, and contention-based access. The wireless channel in wireless body area networks is highly variable in time, as mentioned by Boulis and Tselishchev (2010), who initiated and motivated the design of a hybrid contention and polling MAC protocol to best suit the WBANs characteristics. The protocol is described in detail in Section 3.3 but here we look at the rationale that caused their contention-based MAC approach to be also considered as a possible solution for WBAN. As highlighted in an interesting case study in Boulis & Tselishchev (2010), it is shown how differently a WBAN network might respond to the received traffic in both its contention-based and schedule-based periods. The protocol has been made more contention-based or more schedule-based by prolonging or shortening its

contention access period. It is interesting what happens in terms of packet breakdown rate for different data rates as the contention interval changes in length. In fact, there is no clear and constant behavior as to what contention period length would best respond for a given data rate. It happens at times when a contention period length of the same size would lead to less packet breakdown for a higher data rate compared to a lower data rate! An absolute explanation to this phenomenon is that beacon packets might get lost at times and if a node loses its beacon packet it will have no activity during that beacon interval and its packets will be queued in its buffer. This happens as a result of deep fades in the channel, making beacon and control messages vulnerable to be received by all nodes. The packet breakdown rate, however, evidently increases when the contention period length approaches its maximum, since the channel-sensing mechanism involved in the CSMA algorithm imposes considerable overhead to the network. More packets may get dropped on their first try if the length of the contention is longer arising from more interference that the packets cause to each other when they contend for the medium through CSMA. But the CSMA nature is a random one where it assigns the unsuccessful trials a backoff time that is generated randomly out of the contention window. For this reason, the chances of having a collision upon a node's retransmission are slimmer. So CSMA can act better in its second trials rather than the first ones if the contention period is quite lengthy. Overall, CSMA suffers from having more interference but, as opposed to TDMA, it spreads the retransmissions and makes them not to be back-to-back and thus reduces the probability of packet losses (Boulis & Tselishchev, 2010) and increases the packet delivery. The consecutive loss pattern in TDMA approaches in case of a deep fading in channel is also reported in Liu et al. (2011), as its duration can be long enough to cover the time needed for at least a few packets to be retransmitted. "Transmission reliability" is treated as a different concept to "having fewer collisions" in data transmissions. While reliability shifts to a more effective way of spreading the retransmission attempts over the next available time slots after a failed transmission, having collisions is more of an always-present issue in the case of a non-scheduled access. In other words, CSMA is considered a solution to having more reliability in handling the failed attempts while suffering from collisions itself (Liu et al., 2011). In Zhishong et al. (2011), the CSMA/CA approach in the beacon-enabled mode of access for IEEE 802.15.4 is not considered acceptable for latency/energy sensitive WBAN as it undergoes a large amount of delay by depending on performing at least one backoff procedure and two CCAs. Therefore, they conducted their research on the beacon-enabled mode of access with a full GTS capability during their simulations. In Ghildiyal et al. (2011), it is described that contention-based techniques act better in low data rate and low priority sensor applications as such applications

do not need to maintain the heavy synchronization with their master node in a TDMA approach. Although we consider various data rates for our physiological nodes in our proposed method, we do not consider emergency and very high data rate applications in our WBAN. This gives more credential to the mixed contention-based and TDMA-based approach that we have followed in designing our protocol, which can act moderately in accordance to the traffic it receives.

1.9.2 **Preamble Sampling-based MAC Protocols for WBAN**

Low power listening and preamble sampling are two techniques to reduce energy consumption in a network. They reduce idle listening by putting the radio of a sensor node, as its most power-consuming component, into periodic listen and sleep intervals. Preambles are a means to reassure the receiver is not in a sleep mode once the data is sent from the sender side. Nodes must check the channel periodically to learn if they can send their data packets. A preamble attached to a data packet must therefore be long enough to ensure the receiver has entered its active mode by the time the packet reaches its side. Preamble sampling techniques do not take advantage of synchronization. Time synchronization is always energy consuming and imposes more delay and overhead to the system by adding guard times for avoiding clock drift, and it becomes even more complex when nodes have different traffic attitudes. Mahtab Alam et al. (2012) propose a method based on a preamble sampling technique and claim it as the best option for variable traffic as opposed to scheduled (TDMA-based) schemes, which are based on synchronization. An adaptation of wake-up interval based on the received traffic is discussed in which every node adapts its wake-up interval dynamically with the amount of packets it receives to minimize idle listening. The wake-up interval length finally converges to a steady state due to both consequent fixed traffic received and also fast variations in traffic received from the physiological sensor nodes. No synchronization is introduced as the protocol is a preamble sampling one and hence makes it compatible with a wide range of applications while reducing the cost of synchronization-induced overheads. MAXiMAC (Hurni & Braun, 2010) is another example of a preamble sampling MAC protocol that presents a good level of quality of service suitable for healthcare monitoring applications. Failure in adapting to the ongoing traffic conditions of the network in previously introduced energy-efficient MAC protocols is the motivation behind their protocol design. MAXiMAC presents great adaptivity to the current traffic without sacrificing latency and throughput requirements. High and sparse traffic conditions have been addressed through the design of their protocol and it has been contrasted against some energy efficient MAC designs to prove its optimal performance. The

design principle in MAXiMAC is the ability of the protocol to adapt to ongoing traffic conditions of the network, particularly in high load situations, whilst being energy efficient. The energy efficiency aspect of MAXiMAC is more evident in periods of less activity by the sensor's transceiver and on the other hand it reaches its best throughput and latency performance in periods of high load in the presence of an event. Their approach takes advantage of the WiseMAC (El-Hoiydi, & Decotignie, 2004) to minimize the length of the preambles based on the wake-up schedules of the neighbors. Two threshold values have been defined for the rate of the incoming packets based on which extra wake ups will be allocated or which will be deallocated if the rate falls above or below these threshold values. If the rate of a source node (transmitting node) is increasing, it can be estimated by the destination node and if it estimates the rate to be higher than the upper threshold it will add one extra wake-up period between the default intervals of the preambles. The wake-up period will be doubled if the estimation passes the second threshold. The waiting times for a packet to reach the destination node is halved once the first threshold is passed and quartered once the second threshold is passed, which shows how effectively MAXiMAC reduces the idle listening to improve reliability by their traffic adaptive preamble sampling-based technique. The source node will then learn about this extra wake-up period of the destination node by the added information to the ACK message it receives. There is also the concept of a further threshold that indicates a point higher than the second defined threshold. This threshold shows that the destination device has entered a state in which it must keep its radio on for a defined time span. This may increase the energy consumption by some amount but is to address the high traffic rates with an energy-unconstrained CSMA technique. Once the data rate drops below the first defined threshold, nodes go back to their basic default interval states in the preamble sampling technique.

1.10 **Chapter Summary**

In this chapter we became familiar with some fundamental concepts of WBAN such as their structure and their stringent requirements that set them apart from other types of existing networks such as WSN. These requirements put new challenges towards the protocol stack design, which will be discussed more in the next chapter. We investigated different classes of MAC protocols such as contention-based and schedule-based, their different types, and the strengths and weaknesses of each in different network scenarios. The IEEE 802.15.4 standard, as the base and benchmark protocol of this research, was introduced and described in detail through its different modes of access. The next chapter elaborates on the main motivation

behind this research and the main application type in WBAN that our proposed protocol will be supporting.

1.10.1 **Thesis Contributions**

The main objectives of this research are to seek for appropriate application- and protocol-specific parameters and incorporate them into tuning the backoff time for individual sensor nodes of a WBAN. Therefore a through discussion of the drawbacks of the chosen benchmark MAC protocol (i.e. IEEE 802.15.4) in terms of its backoff algorithm is required which appears in Chapter 2. In the light of discussions in Chapter 2 and based on the reviewed literature in Chapter 3 which includes many of the traffic adaptive approaches for both WSN and WBAN, a protocol-specific parameter (denoted as: $Channel_{Clear}Rate$) is defined and described in Chapter 4. This parameter is jointly used with each sensor node's individual data rate value in a two-input-one-output fuzzy system to yield to a dynamically adjusted backoff window in IEEE 802.15.4 MAC approach which results in a fair access to the shared wireless channel. The proposed fuzzy-enabled MAC has been evaluated with its both application- and protocol-specific inputs (data rate & $Channel_{Clear}Rate$) in simulation environment and only with its protocol-specific parameter ($Channel_{Clear}Rate$) on real SHIMMER sensor platforms introduced in Chapter 5.

The main contributions of the proposed fuzzy-enabled MAC approach appear in Chapter 7 and are summarized as below:

- Higher level of reliability
- Lower level of delay
- Having no effect on the level of energy consumption for the fuzzy algorithm running on the sensor's side.

1.10.2 **Thesis Outline**

This thesis is organized as below:

Chapter 1: Chapter 1 describes the basics of WBAN and the unique requirements associated with such networks and also the main differences they exhibit compared to WSN. Medium access control protocols and their two main categories along with their capabilities are also discussed. The benchmark MAC protocol of the IEEE 802.15.4 is described and detailed

through its different modes of access. Differenr MAC techniques for WBAN are researched and advantages and disadvantages of each are explored.

Chapter 2: Chapter 2 focuses on the statement of the problem that motivated this research. The types of traffic in a WBAN are discussed and examples are brought from different works reviewed in the literature. The main inefficiencies of the backoff algorithm of the IEEE 802.15.4 MAC are investigated and the main method path along with the limitations and exact scope of the work are given.

Chapter 3: Chapter 3 is dedicated to reviewing some of the recent and mostly related works in the literature in the area of MAC protocol design. The chapter is divided based on what parameters are taken advantage of in enhancing the MAC performance. The studied works are mostly from WBAN but some WSN MAC implementations are also discussed where chosen parameters can easily be applied to a MAC enhancement for WBAN protocol implementation.

Chapter 4: Chapter 4 introduces the main method taken in order to address the problems associated with the backoff algorithm of IEEE 802.15.4 MAC earlier discussed in Chapter 2. The building blocks of the method have been discussed and necessary simulations have been carried out to determine the range of variations of main parameters used in the method. The chapter ends with a summary of the code architecture in the selected simulator tool where the proposed algorithm has to be implemented along with an introduction to the chosen simulator.
Chapter 5: Chapter 5 gathers the simulation evaluations of the proposed MAC algorithm in Castalia simulator (introduced in Chapter 4) against the MAC performance of IEEE 802.15.4. Detailed descriptions of configuring the MAC parameters and evaluation of different QoS metrics are given. The chapter also includes realistic energy measurements of real SHIMMER sensor platforms along with an introduction to SHIMMER technology which is the sensor platform of choice for this research. The complexity of the integrated proposed MAC algorithm is discussed in terms of its energy efficiency before the proposed algorithm is implemented on real sensor platforms in Chapter 6.

Chapter 6: Chapter 6 gathers the experiments carried on real SHIMMER sensor platforms for evaluating the proposed MAC algorithm against that of IEEE 802.15.4 in terms of reliability. The chapter begins with some essential set ups in order to prepare the sensor platforms for some necessary tasks such as reading and writing data from/to a SHIMMER sensor’s memory.

The proposed MAC algorithm is then implemented into the oparting system of the sensor platforms which needs a through discussion of TKN15.4 as the only MAC implementation of IEEE 802.15.4 in TinyOS operating system. Some reliability evaluations and parameter discussions concludes the chapter.

Chapter 7: Chapter 7 summarizes the thesis by reviewing the main contributions of the proposed method addressing the inefficiencies of the IEEE 802.15.4 MAC implementation for low data rate applications of WBAN. Some future directions are suggested as well.

2. CHAPTER 2: MOTIVATION AND PROBLEM STATEMENT

In this chapter we discuss the traffic types that exist in a WBAN. We investigate the traffic nature existing in the WBAN and also WSN. Uunderstanding the generated traffic in a WBAN helps to tune the MAC protocol behavior to perform efficiently in any upcoming traffic situation. The chapter gives a general introduction to the basics of the proposed MAC protocol and the main problem identified in the aggressive backoff procedure of the CSMA/CA mechanism in IEEE 802.15.4. This backoff mechanism is the main focus of this thesis, and is investigated in this chapter through the standard's poor performance caused by either inefficient or identical backoff situations. Different parameters to be used in our method are noted in this chapter as well, but detailed descriptions of them are given in Chapter 4. The aim and objectives of the thesis along with the scope of research and its main limitations are discussed before moving to Chapter 3.

2.1 Traffic in Wireless Sensor Networks

Inspired and driven by military applications, wireless sensor networks (WSN) are large deployments of sensor entities that often perform a monitoring task in a co-operative manner that is barely applicable to wireless body area networks with heterogeneous applications. WSN is scalable to hundreds of thousands of sensor nodes depending on the application, which limits the individual sensor nodes to exhibit severely different traffic characteristics or being completely isolated in handling their own traffic to the base station. When large numbers of sensors are randomly scattered in an area of interest, they all cooperate to do a common task to provide a reliable data transmission to where the data is intended to be used, possibly miles away from where the sensing is taking place. The multi-hop nature of most of the WSN implementations allows for self organization of different routing techniques to ease the data transmission and compensate for failures. Although the reliability of the received data matters in WSN, loss of the data will not be life threatening, in contrast to WBAN. The first and foremost concern in WSN is energy consumption; especially if the vast sensor deployment has been carried out in a remote terrain where changing the batteries to keep the sensors alive cannot be a possibility. Such disparities in the type of environment and the services provided by both WSN and WBAN applications have led to different desired quality of service metrics such as reliability or energy. In WSN, reliability can be considered as a secondary concern when nodes can self-organize and a single point of failure does not stand as a barrier in sending

the sensed data as it can be compensated for by the other nodes in its vicinity doing the same task at the same rate.

It is often emphasized that understanding the traffic behavior of a network (which is directly linked to its supported type of application) would result in a better protocol strategy design (Wang, 2010). Some mathematical models such as Poisson distribution have been used to model a variable traffic as opposed to normal (or periodic) traffic, often represented by constant bit rate (CBR). While it is not always easy to model the network traffic within the context of simulations or even to analytically investigate them, the varying parameters of a network, reflecting its real time traffic, can be exploited to tune protocol functionality towards fulfilling the target QoS metrics. Traffic adaptive techniques have been used for wireless local area networks (WLAN) prior to the advent of WSN. Although the protocol specifications in these two classes of wireless networks are slightly different, most of the traffic adaptive approaches can be applicable to both, with minor configuration adjustments in regard to the protocol stack implementation. Having said that, the desired quality of service metrics also vary from WLAN to WSN, as their application scenarios do not share many similarities. Some traffic adaptive protocols proposed for WLAN, which are based on the IEEE 802.11 distributed coordination function (DCF), take advantage of different parameters, such as average number of collisions in a time interval and average length of idle times (Cali, Conti, & Gregori, 2000), average length of time spent during collisions (Peng, Cheng, & Lin, 2003), data rate and node density (Lv, Zhang, Han, & Fu, 2007), history of recent transmissions (Nasir, 2008), etc. Other traffic-related metrics such as residual energy have been used to reflect the on-going traffic of the network into tuning the contention window (CW) for backoff time generation in the MAC protocol. An example of this is the research carried out for WSN applications in Cho et al. (2006), where the probability of collisions has been reduced drastically by their traffic adaptive approach. Gong et al. (2005) discussed another traffic-adaptive MAC protocol proposed for WSN where the average number of collisions in the network was used as a measure for traffic intensity. The motivation for their work was the dynamic change in the network traffic that leads to the need for a contention window that can take dynamic values according to traffic intensity so that the collision will be reduced by a balanced backoff time generation. Some of the other traffic-adaptive techniques and their fulfilled QoS parameters are discussed in Chapter 3 with a classification of the type of parameters that different algorithms exploit. It is important to note that the design of any protocol must be in close alignment to the application's stringent requirements so that the tradeoffs made will ensure the desired system performance.

Hurni and Braun (2010) explained how the difference in the traffic characteristics of one network to another would affect the design of its MAC protocol from being energy efficient to being more robust in reliability and latency performance. The type of traffic in a network must dictate the design of its protocol stack to fulfill the requirements for such traffic to go through as smoothly as possible. One way of achieving this goal is to exploit and incorporate traffic adaptive parameters and techniques, as mentioned above, into the protocol's body in such a way that the required performance balance is achieved. The choice of the traffic adaptive parameters to use depends on the tradeoff that is to be made.

2.1.1 Traffic in Wireless Body Area Networks

Unlike traditional wireless sensor networks, where all the sensor nodes in the network co-operate to do a common task, wireless body area networks suffer from heterogeneity in the applications that run on each sensor node (Latré et al., 2011). Each application has its own attributes in terms of data rate/sampling rate, priority, reliability, and delay criticality which should be taken into consideration in the structure of a protocol. In other words, a versatile protocol is one that can be resilient to all these inherent characteristics and performs well accordingly, especially under varying traffic conditions. The type of traffic in WBAN is reported to be different compared to WSN traffic patterns in the sense that it is less correlated and sometimes not correlated at all. If we assume the rates at which a WBAN sensor samples the data to be dynamic according to the abnormalities it detects, an ECG sensor may send out the data at a higher rate when, for example, a heart attack happens. If a temperature sensor with a dynamic data rate also exists in such a scenario, it is likely that it will increase its sample rate due to a sudden change in the body's temperature at the same time as the ECG sensor, but that may not necessarily happen. If an EMG sensor is also attached to the body of that patient, it may not detect any abnormalities at all when a heart attack happens and it will not increase its rate. This unpredictability in traffic from each node means that a sensor designed to capture body movement to monitor a fall, for example, will generate traffic at different data rates as the patient moves, sits down, or runs and it is not necessarily correlated with other nodes' on-going traffic. The above example is the case where the data rate (sampling rate) of a sensor node is programmed to alter by every change triggered in the accelerometer readings that exceeds a threshold. It is the same for an ECG node, for example, that is put on a patient's body to detect heart attacks; the traffic that an ECG sensor generates, when programmed to change its sampling rate based on the extreme changes in the ECG signal, is also not predictable (O'Donovan et al., 2009). In Ghildiyal et al. (2011), the traffic of WBANs is however suggested

to be only periodic. It is suggested in some research papers that even in situations with emergency traffic in a WBAN the sudden changes in the traffic pattern of a physiological node only take a little while and transit to the normal traffic quickly (Huq et al., 2012). The contention-free access period of a super frame in Huq et al. (2012), with a guaranteed slot management, is then claimed to be not 100% efficient in terms of using all of its assigned GTS slots during CFP. Data rate is however only one of the many sensor-specific attributes that differs from one node to another. As mentioned earlier, nodes in a WBAN also stand apart in other terms such as the priority of the data they send out to the coordinator or even the delay criticality of the data, which decides how fast the sensory data should travel to the coordinator. This brings heterogeneity to the world of WBAN in contrast to WSN, where the sensors of a network are said to be homogeneous and therefore the nodes' behavior in WBAN in having access to channel is not the same at all times. Some nodes may acquire the channel for a very long time due to the nature of their application and some may have to wait a long time before they can relay their data over the channel. There are many works in the literature that have emphasized such traffic heterogeneity in WBANs (Mahtab Alam et al., 2012; Ahmad, Riedl, Naramore, Nee-Yin,& Alley, 2009) that exemplify the great variations in data rates of different physiological sensor nodes or great differences in data rates in case of periodic traffic. As reported in a variety of related literature, the data rate ranges for different sensors deployed in a typical WBAN vary depending on their applications (Mahtab Alam et al., 2012). Different data rate ranges for different applications of WBAN appear in Latré et al. (2011) where the data ranges differ based on the specific physiological vital sign to be monitored. The data rate can be as small as only 16bps for blood saturation monitoring to as high as 1Mbps for audio monitoring tasks. In a study by Bilstrub (2008), the data rate range for WBAN applications is identified as <1 Mbps for health monitoring applications. A typical ECG sensor may have a data rate as high as 50 to 300 kbps. Although the individual data rates are not expected to be very high for WBAN applications, the aggregated data rate could be in the order of a few Mbps. In Zhen, Patel, Lcc, and Won, (2008) and Isikma et al. (2011), data rates as high as 10Mbps have been reported for some applications in healthcare. In Zhisheng & Liu (2011), data rates of 10 kbps to 1 Mbps are noted, which may be a cause of data rate overload in medical applications. They identified four different groups of nodes based on their data rates in their simulated WBAN model ranging from 1250 bps to 10000 bps. In Ghildiyal et al. (2011), the data rates vary from 10 kbps (for an accelerometer sensor) to only 0.2 kbps for temperature readings. For heart activity readings, they have chosen a 5 kbps data rate. Both accelerometer and ECG sensors have HIGH priorities assigned to them, which signify a substantial disparity

between the assigned data rates of the sensors in their WBAN. In Curtis et al. (2008), the data rate assigned to a medical sensor to send out its ECG and SpO_2 data along with location information to a central unit is said to be about 0.5 Mbps for a total of 10 patients in an emergency room of a hospital.

The data rate might not always be dynamic, but may still vary greatly from one node to the other. The centralized and co-operative traffic in WSN (Ghildiyal et al., 2011) sets them in a different class apart from centralized and non-co-operative traffic in WBAN. Heterogeneity is the first challenge that WBANs bring to the world of conventional WSNs. It relates mostly to the disparate service requirements that each node may have such as data rate, delay criticality, and data priority (Zhang & Dolmans, 2009; Hanson et al., 2009, p. 63), which lead to *traffic diversity* (Li & Tan, 2005), especially when the data rates are dynamically changed, requiring the MAC algorithm to be adaptive to different application categories. Heterogeneity of the application requirement for each node has motivated this work to treat each node individually in which chances of transmission for each node is decided based on a fair channel access mechanism. With a focus on chronic diseases, which do not normally involve high data rates, our approach aims to increase the reliability of the transmitted sensed data to the coordinator by assigning backoff times to each node that reflect the node's past trials in having access to the channel. In our approach, no node will be left in severe need of channel access and no node will acquire the channel for a relatively long time. Such behaviour is achieved through enhancing the backoff approach of IEEE 802.15.4 MAC discussed later. It is described in Chapter 4 and specifically in Section 4.2 how the successful channel access rate for each node is calculated individually and separatelyby taking into account the node's activity in a defined interval of time. The data rate of each node is also a parameter to be considered in our method, which is different from one node to another but, is a static value for each node representing each node's periodic traffic.

The variable unpredictable traffic load also exists in traditional wireless sensor networks with only one certain task to monitor since they are event-driven and only send and propagate the data when an event of interest happens. Fluctuations in the arriving traffic rate of different sensor nodes or the same sensor node of a typical WSN has been mentioned in Ren & Liang (2006), but the severity of various traffic loads is even higher in wireless body area networks where each node is devoted to a different task and the traffic is not correlated anymore. In a WSN, in the event of a change in the monitored phenomenon all the nodes in the same vicinity

tend to have higher data rates than the data rate they had before the event's occurrence but this is not the case in WBANs all the time. With the physiological sensor nodes each having different application requirements, the optimal time slots for packet delivery vary depending on the data rate each node has and also how successful it has been so far in relaying its data over the channel. Boulis and Tselishchev (2010) note that the optimal time slots for packet delivery vary depending on current traffic conditions, and are mostly inspired by the individual sensor node's traffic condition. In this thesis, although the data rate assigned to each physiological sensor node is different from the others, they do not vary over time. We will see later in Chapter 4 that we have used a static array as the input data rates for our network configurations. The data rates assigned are, however, based on low to moderate values that would represent different type of vital sign each node is set to monitor and will be used as one of the two measures used in our traffic adaptive MAC structure. In general, the traffic in healthcare monitoring can be categorized into two main classes:

- **Sporadic:** mostly associated with diseases like epilepsy seizure, heart attacks, or sudden falls associated with motor disabilities like Parkinson's disease. In such cases we may have a sudden change in the signal being sampled from the sensors on the body. Despite the fact that the default sampling rate is fixed at some level when first deployed, the sensors can actually be prompted to change their sampling rate and consequently the data rate received at the sensor, based on exceeding predefined thresholds.

- **Periodic:** mostly associated with periodic monitoring and non-delay critical applications such as temperature or for vital sign monitoring in case of chronic diseases.

Although this research is not about modeling or analyzing the type of traffic in the WBAN, it does take advantage of some parameters closely linked to the traffic behavior of the network. We have used constant but different data rates for the sensor nodes of our simulation and test-bed experiments which illustrate their different application characteristics. On one occasion only, we did show the effect of low to high data rates (averaged for all nodes in the network) to see how the network would respond to a varying load but unfortunately at the time this research had started there was no accurate model for precisely modeling the bursty traffic in the WBANs. In addition to that, we also take advantage of another average parameter (later described in Section 4.2) that has traffic-reflective values as the time goes on. The burstiness

of the traffic present in a WBAN has been closely attributed to mobility variability and the degree of spatial correlation (Wang& Akyildiz, 2011). The hybrid nature of the IEEE 802.15.4 standard exhibits a decent performance level with its contention-based and schedule-based scheme. With dynamic data rates, great fluctuations exist in the individual data rates of each physiological sensor node, referred to as "application dynamics" in Mahtab Alam et al. (2012), which not only highlight the difference between a node's single data rate but also its individual real time variations.

2.2 **Low Sample Rate Vital Sign Monitoring**

Since the traffic behavior in a body area network is not of one specific type and even controlling it is not an option under a realistic framework, understanding the generated traffic would help to tune the MAC protocol behavior to efficiently perform in any upcoming traffic situation. What matters most in a body area network is a timely and flawless response to the vital signs and a great deal of reliability. This requires lesser packet delivery delays endured when sending a data packet to the coordinator device and also fair access to the medium among all the sensor nodes. In the previous section we discussed the heterogeneity of applications in the network for a WBAN and even a dynamic change in data rates was stated wherever the sensor platform would support a real-time variation in data rates in the event of an abnormality. Medical applications can be classified into generally low data rate and high data rate. We assume a periodic traffic for a low data medical set up for our experiments that would best apply to a chronic disease type, such as diabetes or most of the aging-related illnesses such as general every day monitoring of an elderly person. We assume five to ten different physiological sensor nodes (only five nodes for our real experimental set up described in Chapter 6) with pre-defined but diverse and application-specific data rates. The main goal of this research is to propose a highly reliable MAC protocol for low data rate applications but we also test higher data rate sensor nodes in the network. For moderate periodic traffic behavior with reliability in mind as the main concern, a fair access to the shared wireless channel contributes greatly to the overall system's reliability performance. This fairness makes channel access possible for all the heterogeneous sensor nodes in the network while examining their individual data rate value and their success channel access rate. The successful access rate for each sensor node is considered to be our main traffic-indicating parameter, which is derived in the continuing loop of the CSMA/CA performed in the CAP by each node. The motivation for deriving this parameter is the aggressive backoff mechanism in the CSMA/CA mechanism of the IEEE 802.15.4 standard, which would sometimes lead to unbalanced waiting times assigned to the

nodes of a WBAN. The two main methods of calculating this parameter are discussed in Chapter 4.

As discussed earlier in Chapter 1 and also shown in Figure 1-14, we conduct our research on the beacon-enabled access mode of IEEE 802.15.4, which incorporates both a TDMA-based access in its CFP and a CSMA/CA-based access in its CAP. Such a medium access control protocol is therefore able to employ the advantages of both access modes in different upcoming traffic conditions. Most of the research works in the literature have conducted their study over this access mode in IEEE 802.15.4 but there has been, very recently, the emergence of IEEE 802.15.6 that is initially designed for wireless body area network applications but remains out of the scope of this research. We did not conduct our proposed MAC within the context of IEEE 802.15.6 since to this date (2012, the date of our experiments) it had not yet been finalized and only a proposal of it existed ("IEEE, Std. 802.15.4", 2003) and therefore it is not fully implemented in any simulation tool or even used as widely as IEEE 802.15.4 on many of the current chips. We first try to illuminate the problem in more detail in the repeating loop of CSMA/CA algorithm in CAP with a focus on its backoff algorithm.

Referring to the explanations given in Chapter 1 and in Section 1.7.1.1 whatever is seen in Figure 1-14 is happening for *every* transmission. If a node is at the state of transmission in this flow chart then the other nodes are definitely in the state of backoff with different random values assigned to them, otherwise a collision occurs. The event of collision is for when the current transmission ends successfully and two (or sometimes more) of the other nodes finish their backoff state *at the same time* and start the CSMA/CA process together. The idea of backoff in the standard was basically to solve this problem of simultaneous transmissions by generating random backoffs with the BE exponent. This however will still lead to a high probability of collisions and thus less achieved reliability if the random backoff algorithm fails to reduce the probability of generating identical backoff times (identical backoffs: backoffs that end at the same time slot) in certain applications. Identical backoff periods that lead to an event of collision, however, are not the only source of concern in an IEEE 802.15.4 CSMA/CA mechanism. Whilst the standard introduces the backoff as a resolution to less probability of collisions, it does not necessarily address the efficiency of the generated backoff time in non-identical backoff times when no collision actually happens. This shifts the problem to *inefficient backoffs* rather than identical ones. The CSMA/CA algorithm mostly relies on incrementing the BE parameter (which gives and exponential increase to the backoff window)

upon a failure or decrementing it when there is a success. This means that, while having less probability of collisions is a milestone behind a backoff procedure design, it is not the only requirement we look for in an efficient and high performance protocol. In other words, a successful transmission (no-collision situation) may happen with a long backoff time before it is made, which does not suit a body area network application and neither does it for many WSNs. When no traffic-indicating parameter is considered in generating the next backoff period, each node conquers the channel differently from the other nodes. This leads to an unfair channel access among the sensor nodes of a network. A traffic-aware tuning of the backoff period that would consider how successful each node has been in its previous attempts in having access to the channel would help in maintaining a fair access, and would also increase the reliability of the received data at the coordinator device.

The problem of such unfairness gets even worse in high congestion situations. In a highly congested network, the probability of two or more nodes generating the same backoff time is higher and therefore a collision has a higher probability, but that also affects the problem of inefficient backoff time generations in a successful transmission; when the traffic is high, nodes' frequent attempts in finding an idle channel fail, in which case they must backoff to retransmit the packet, but there is a limit in the standard more than which a node cannot backoff and the packet fails. Therefore, a failure is not always reported because of frequent identical backoffs that lead to a collision but also because a node cannot backoff more than a certain amount of time and none of its previously generated backoffs was good enough to let the device transmit.

The question is: can a proper backoff algorithm lead to more reasonable backoff times so that, despite having fewer collisions, more packets are transmitted before they reach their allowed limit of number of backoffs? The answer lies in the ability of that backoff algorithm to generate backoff times that would not keep nodes waiting more or less than needed, and neither would it let them access the channel at the same time as another node. We believe that, if the assigned backoff times are more closely reflecting the current traffic going on in the CAP, the probability of having both an identical backoff and an inefficient backoff decreases.

Identical and inefficient backoff times happen due to a diverse range of reasons that can be *protocol* or *application* related. Examples of application-induced causes of such a phenomenon are high numbers of nodes, high data rates, and different priorities the sent data have. Whilst

data rate and the number of nodes involved in a network are good descriptives of an application scenario and its characteristics, in a very realistic implementation they cannot be considered tunable themselves as they describe an application's requirements. Nevertheless such parameters can be exploited into tuning *protocol-related* metrics that make CSMA/CA algorithm function for each transmission. Protocol-related parameters are other parameters influencing an identical/inefficient backoff phenomenon that is well tunable with respect to application-specific parameters' characteristics and other traffic parameters in the network. Such parameters are more controllable within the confines of the standard and can be tuned based on both the application's needs and the on-going traffic being handled by the MAC protocol. Figure 2-1 shows a summary of different categories of what possible sources of an identical backoff and inefficient backoff problem exist. The list of parameters on both sides, however, is not limited to what we have shown here, and a wide range of other metrics, specially the protocol-specific ones, can be drawn based on the research's needs.

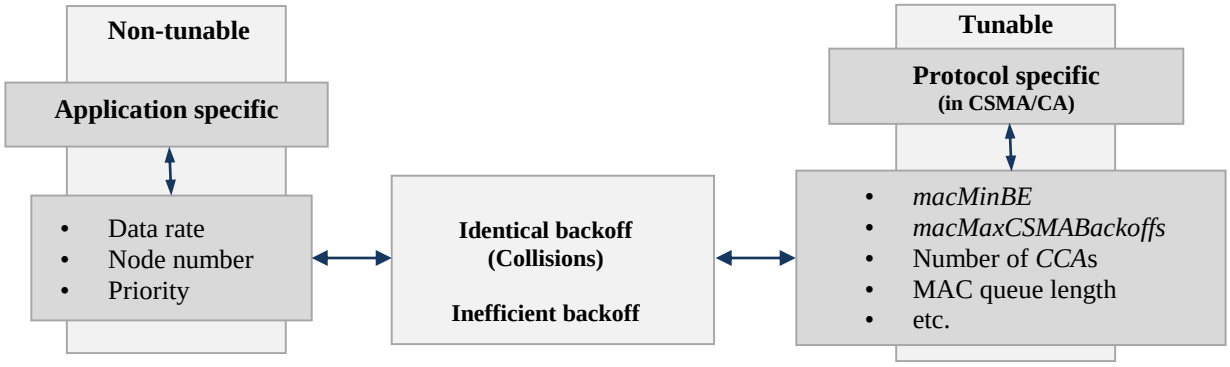


Figure 2-1: Identical/Inefficient Backoff Sources Classification

Figure 2-1 shows that backoff time is actually influenced by many metrics, both application- and protocol-specific. The influences for the application-specific sources might not be explicit and direct but they do lead to having a more or less congested situation in which the standard's generated backoff times may not satisfy a fair access to the channel. With a more congested network, for example, we will have more backoff times generated and more nodes spending their time in waiting periods as the number of collisions increases. However, as mentioned before, in a realistic implementation you cannot dictate or alter the real values of data rate for the sensors or the number of nodes deployed in a sensor network. On the other hand, with the protocol-specific parameters we have an explicit and direct impact on the values generated for a backoff time. Therefore we shift our focus onto modifying the protocol-related parameters

that we can but we also try to consider one application-specific parameter, such as data rate, for its implicit impact and because it can be exploited in the designed MAC algorithm together with our protocol-specific parameter defined and later explained in Chapter 4.

2.3 **Intended MAC Algorithm Placement**

The intended MAC approach for this research, as represented in Figure 2-2, is based on beacon-enabled mode of IEEE 802.15.4 with a star topology to take advantage of the resource asymmetry that exists between the coordinator and sensor nodes. The proposed MAC algorithm acts as a slight but effective change to the CSMA/CA process in the CAP period of the super frame. The loop of CSMA/CA algorithm was discussed in details in Section 1.7.1.1 where our fuzzy-enabled MAC algorithm will be implemented in. The highlighted parts in Figure 2-2 are explained more precisely in Section 4.2. As can be seen from Figure 2-2, a fuzzy-based algorithm is used for the calculations of the protocol-specific parameter in our method that acts as one of the inputs of the fuzzy system. The other input, as discussed before, is the data rate of the node which, together with the successful channel access rate (attributed in $Channel_{Clear}Rare$ variable), will be fed into the fuzzy system to decide a traffic-adaptive range for the backoff time that is to be randomly generated and assigned to each node during its backoff mechanism. This traffic-adaptive range gets updated every specified interval of time, which is defined in terms of a number of super frames, as discussed in Chapter 4. The fuzzy system resides on the sensor's side and we will see later in Chapter 5 how we have simplified it to be almost no threat to the battery lifetime of our sensor platforms.

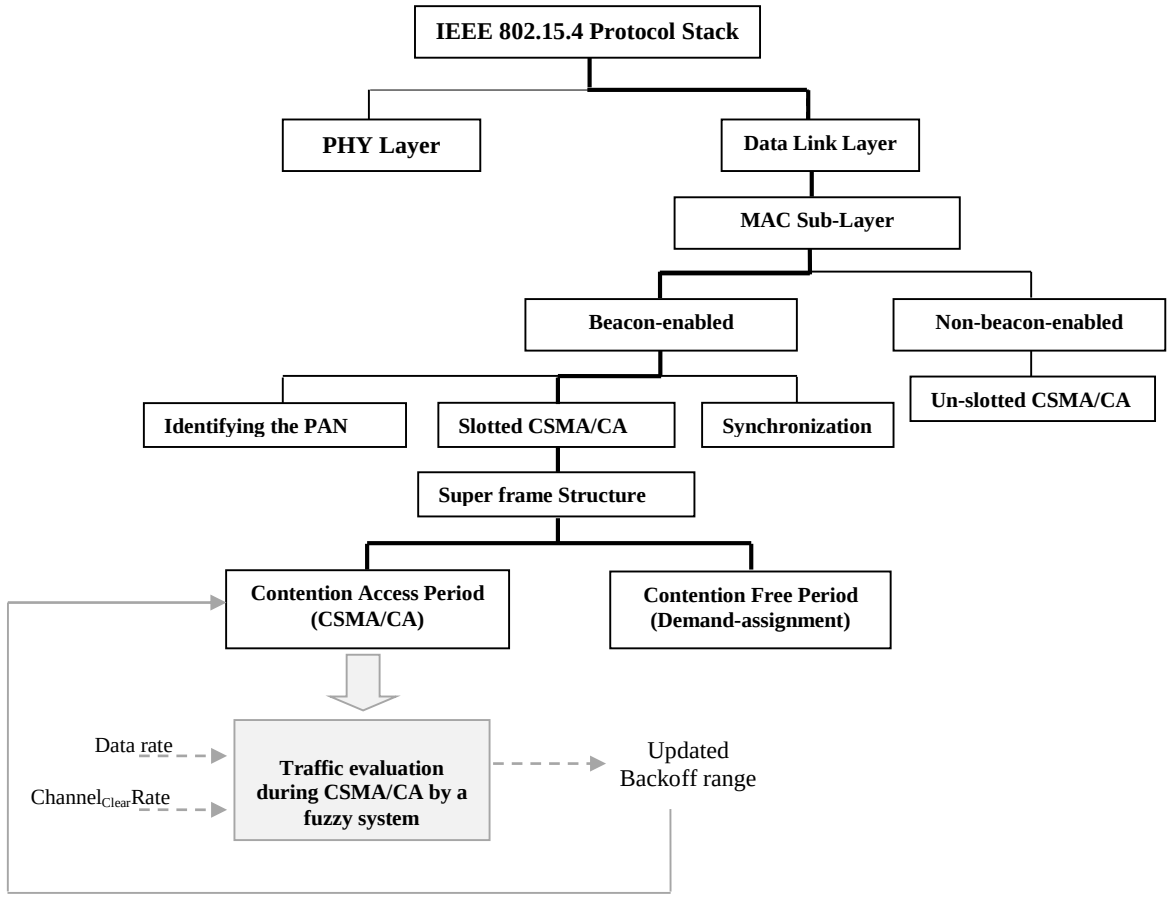


Figure 2-2: Intended MAC algorithm placement (highlighted)

2.4 **Research Aim and Objectives**

As mentioned earlier in this chapter, to address the heterogeneity of the application requirement for each node, which has motivated this work, each node is treated individually in terms of the backoff time assigned to it in the CSMA/CA algorithm. Therefore chances of transmission for each node are decided based on a fair channel access mechanism. With a focus on chronic diseases, which do not normally involve sudden high data rates, our approach aims to increase the reliability of the transmitted sensed data to the coordinator by assigning backoff times to each node, reflecting the node's past trials in having access to the channel. We have designed a traffic-adaptive MAC algorithm on top of IEEE 802.15.4 that can organize a fair access among the physiological sensors of the WBAN.

Following the aim of the research discussed in the previous section, the primary steps to take towards thatare:

- Designing a dynamic fuzzy-enabled backoff window and
- Integrating the designed fuzzy algorithm into the CSMA/CA of IEEE 802.15.4 backoff procedure that exploits the output of our fuzzy logic system.

These two steps are highlighted in Figure 2-3. Due to dissimilar and various characteristics of each biomedical sensor node in a WBAN, no versatile MAC algorithm would exist as each proposed scheme in the literature takes on a different target QoS metric to improve and not all of them have the same application scenario. When designing a MAC protocol, different criteria could be set and considered that put different weight factors on each of the target application's requirements, be it power consumption, transmission delay or throughput (Lamprinos et al., 2005). The variety of data ranges, for instance, is one of the driving factors behind different MAC strategies required in different situations, as also described in Ghildiyal et al. (2011), who propose a MAC protocol that takes advantage of the merits of two various access modes: TDMA and CSMA, which suggests a hybrid technique to be the most suitable for dynamic MAC designs.

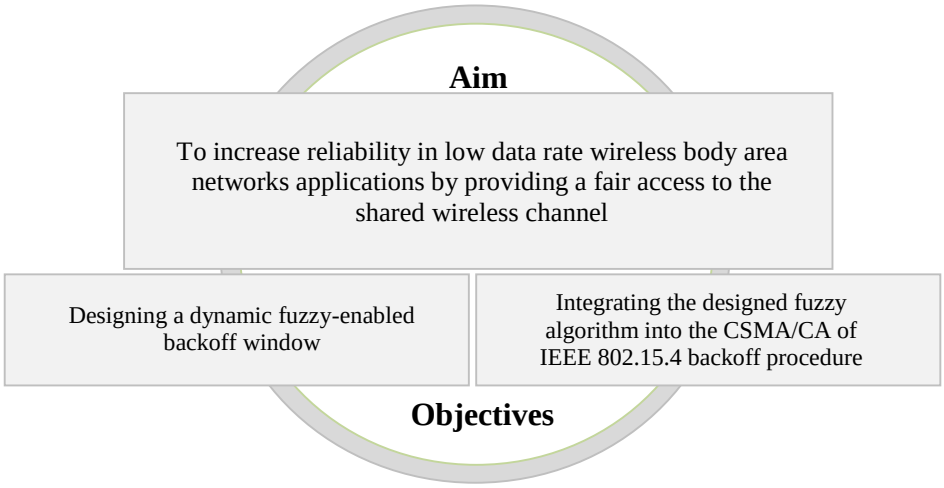


Figure 2-3: Aim and objectives

In the next chapters, we explain how we have tried to fine tune the current MAC algorithm in IEEE 802.15.4 based on the observations of the current traffic, which mostly relies on each node's waiting times, to achieve a reliable and dynamic behavior based on both contention- and schedule-based techniques. Having discussed the advantages and disadvantages of each of the TDMA- and CSMA-based techniques, a hybrid approach that exploits the best of each technique seems to be the best option for a highly volatile yet energy- and reliability-sensitive structure as evident in any typical WBAN. Many of the reviewed works such as Ghildiyal et

al. (2011), Boulis & Tselishchev (2010) have consolidated the best of two methods to create a desirable MAC behavior for WBAN.

2.5 **Research Scope**

According to the proposed classification of different types of communication in a WBAN in Latré et al. (2011), this research only focuses on the *intra-body communication* links that concentrate on the data communication between the on-body sensor nodes and their coordinator. The research is limited to the MAC sub-layer of the data link layer as the closest sub-layer to the physical layer, thus no detailed description of the logical link control (LLC) sub layer under 802.2 Standard appears in this thesis. The achieved simulation and experimental performance results have been benchmarked on IEEE 802.15.4 MAC specifications as the standard for the MAC sub-layer of the protocol stack. Therefore no routing concept or protocol has been discussed in this thesis and all the mechanisms relating to it are considered out of the scope of this research. The detailed discussion of the responsibilities of a MAC sub-layer in Section 1.5 gives a better understanding of the research's confines. The CSMA/CA channel access is the focus of this study which is more discussed through MAC protocols for WBAN in the literature in Chapter 3. The pros and cons of both TDMA and CSMA approache are identified and discussed wherever related. The topology taken is star and for the simulations works the effect of interference, body attenuation, and the different mobility models of the body have not been taken into consideration. Therefore, we have not assessed the effects of shadowing or attenuation on the received signal at the coordinator device. Although we use the Castalia Simulator for its better and more realistic channel model for a WBAN implementation, we have not investigated such effects beyond the already existing path loss models that Castalia has provided. Although the use of actuators as complementary devices mounted on the sensor nodes is described in some research papers (Latré et al., 2011), in our research we have not considered any actuator device to act in response to the data gathered from the sensor nodes. No energy-harvesting techniques are presented in the proposed protocol such as obtaining energy from body temperature or vibration. No energy-scavenging technique is used for the SHIMMER sensor nodes that we used for the experimental phase of this research. The sensors merely rely on their batteries. The sensor nodes only collect the data and send it over to the assigned coordinator for data analysis. No node-to-node communication is considered and sensor nodes only relay their data to the coordinator in a single hop manner. Likewise, no network of coordinators has been considered as a potential situation to occur and, even if a few patients are to be placed in the same room, no communication among the

coordinator devices is likely to happen. Ghildiyal et al. (2011) have described this as an "uplink" manner for the nature of traffic in WBAN. The communication link can only be initiated by the coordinator as we closely study the MAC approach in the beacon-enabled mode of access. Castalia 3.2 (Boulis, 2011) is used for the simulations, thus the channel model is as close to real experiments as possible. After the simulation stage, the proposed MAC protocol was tested on real SHIMMER sensor platforms to compare the effectiveness of the method with the results obtained during simulations in terms of reliability and energy efficiency. For this stage of the work, we did not seek ethics approval as it was just an experimental test of the performance and not really implemented in a hospital or clinical environment. However, it is confirmed that all SHIMMER sensor platforms are CE marked, which relaxes any regulatory hurdle for them to be used in a real health care environment. The simulation scenarios are carried out with 5–10 nodes and on one occasion the effect of a larger number of nodes has been tested with 35 nodes. In real experiments, however, we did not have the possibility of more than five nodes in our set ups which is also a fair number for a typical WBAN. Therefore the reliability measurements are tested with only five nodes as compared to the simulations but within the same time frame. The results of the experimental stage might be different on diverse sensor platforms produced by different manufacturers but discussing the differences is out of the scope of this research and it is expected that the proposed algorithm will only be tested on one sensor platform, which is chosen to be SHIMMER ("SHIMMER Sensing Techology", 2014) nodes. The design of the controlling modules on the base station where the physical practitioner is, as well as the user interface design on the smart phone carried by the patient, is out of the scope of this research. As the work is specifically carried out for wireless body area networks, the range of communications in terms of meters is no more than five meters as also declared in Bin (2008), however in our case, as we are only considering a network of five nodes when doing the experiments with SHIMMER nodes, the range of communication between each node and its coordinator does not exceed one meter. The carrier used to do the experiments is the 2.4 GHz ISM band and the research does not consider using the specifically WBAN designed bands such as MICS and WMTS. The main target application category for our proposed MAC algorithm in healthcare is chronic diseases, which do not have an urgent sudden load of high priority data generation nature. Therefore no priority is given to the physiological nodes of the body area network described. As declared in the standard draft, no security measures are considered in the scope of the IEEE 802.15.4 standard and it is assumed to be the higher layers' job to establish and maintain the trust relationships between the devices ("IEEE

802.15.4 Std", 2003, p. 24). The symmetric-key cryptography is, however, used in the standard that takes advantage of the keys calculated by higher layers.

Chapter SummaryWireless Body Area Networks have been proved to be a powerful tool in providing a means to monitor patients remotely, giving them needed safety in the comfort of their homes. The rapidly growing population of elderly people around the world is a big motivation behind the deployment of such networks as it eases the care systems to be provided at the comfort of their homes. Wireless body area networks have also been used to control obese and disabled children (Wong et al., 2009, p. 6576) as well as in the sport and entertainment industries. However, healthcare applications have attracted a lot of research in recent years in the area of body sensor networks in order to mitigate the costs of care giving and also leaving more rooms in hospitals for more critical situations, especially in mass casualty events. Emergency situations demand more hospital personnel to be involved in the situation for a short period of time where other patients suffering from chronic diseases might be neglected for a while. WBANs have emerged as a possibility to provide real-time caring to such patients even when they cannot be accommodated inside a hospital unit. When dealing with a person's life, it is crucial to be able to keep certain QoS parameters at a certain desired level to ensure data fidelity, timely arrival of data, and as little energy consumption as much as possible during data communications. This research has studied these parameters through a novel MAC protocol aiming at providing a traffic-adaptive technique that is able to describe the current traffic of the network and uses this information to dynamically tune some key parameters of the base MAC. The proposed technique briefed in this chapter will lead to a high level of reliability in transmitting data to the coordinator as well as improved packet latency. Identical and inefficient backoff phenomena have been discussed in this chapter and two of the disadvantages of the original IEEE 802.15.4 standard which could endanger the reliability factor in a WBAN design have been discussed. A brief introduction to the proposed method along with its parameters has been given in Section 2.3. The motivation behind choosing these parameters has been discussed as well as the target QoS performance parameters to achieve. The main confines of the thesis are discussed in Section 2.5 as the scope of this research work.

3. Chapter 3: RELATED WORK

This chapter reviews the related works noting the dynamicity of their approaches with respect to the different network parameters they engage. The related literature has been divided into six main categories with an emphasis on the first three (3.1, 3.2, and 3.3) as the most popular methodologies taken by the researchers around the world. Table III summarizes the reviewed work in this chapter. The first category of the reviewed literature (Section 3.1) skims over some of the traffic-adaptive protocols that take advantage of the varying MAC parameters during a super frame structure within its CAP and CFP periods. These MAC parameters are the parameters that have to be maintained for each transmission that happens within the beacon interval; namely CCA, BE, NB, macMaxBackoffs, BO, SO, etc. Dynamic techniques where certain network parameters are influenced by the on-going situation of a network provide the possibility for a MAC protocol to act according to the real-time network condition to fulfill specified QoS parameters that reflect the application's requirements. The second category of the related literature (Section 3.2) takes on a broader vision on other existing MAC parameters that are not necessarily assigned to a CSMA/CA process but are effective on target QoS metrics. Such parameters could be the specific duty cycle, the MAC layer queue, or the structure of a super frame. PHY layer parameters such as link quality and received signal strength (RSSI) can also be incorporated in some of the decisions and actions made in the MAC layer, as we will see later in Section 3.3 as the third category of dynamic methods presented in this chapter. The related work ends by inspecting some of the related researches integrating fuzzy-logic algorithms in Section 3.4.Therefore the main classification of the studied works in this chapter is based on the specific network parameter to be tuned (coming from different categories), which also enlightens their effects on the different performance parameters under study. Not all the studied works are related to WBAN scenarios but they suggest similar enhancements that can be applied to a WBAN MAC layer when being fully aware of the tradeoffs to make and also the specific scenario of the application under study. The backoff scheme in IEEE 802.15.4, which is the main focus of this thesis for example, has slight differences from the backoff in CSMA/CA algorithm in IEEE 802.11 MAC implementation (as we have referred to in some of the reviewed works) but the related dynamic approaches for WLAN are occasionally mentioned here as they can also potentially apply to CSMA/CA behavior in IEEE 802.15.4 MAC protocol especially in regard to its backoff behavior.

Table III: Reviewed works

Problem	Author	Method	Contribution
Exploiting MAC Parameters in Super Frame Duration (CAP & CFP)			
Dynamic Evolution of CSMA/CA Parameters in CAP			
Lack of efficiency in generated backoff periods	Hadid, Guitton, & Misson (2009)	Dynamic change of *macMinBE*	Better goodput and delay performance
	Park, Fischione, & Johnsson (2010)	Channel state inspired backoff/optimized *macMinBE*, *macMaxCSMABackoffs* & *macMaxFrameRetries*	Energy optimization/Improved reliability
	Prakash, Rao, & Marandin (2006)	Dynamic traffic-adaptive change of *macMinBE* range	Reducing the probability of identical backoffs
	Golme, Cypher, and Rebala (2005)	Change of *macMaxCSMABackoffs* and BE	Better delivery ratio
	Koubaa, Alves, Nefzi, & Song (2006)	Dynamic values of CW_{ini}, *aMaxBE*, *macMinBE* based on different data priorities	Improved delay performance for time-critical applications
Dynamic GTS Allocation in CFP			
Lack of efficiency in bandwidth utilization for emergency data	Li, Hao, Zhang, Liu, & Li (2011)	Early transmission of GTS requests	Efficient GTS allocation
	Huq et al. (2012)	Dynamic allocation of listening windows in MAP	Higher throughput
	Cheng, Bourgeois, & Zhang (2007)	Using smaller time slots	More efficiency in accommodating suddenly increased GTS demand
	Ying Lei, Choi, Park, & Rhee (2012)	Prioritizing the physiological sensors' traffic	Higher throughput
Dynamic Super Frame Order and Beacon Order Values			
Lack of energy efficiency/High delays in handling emergency data	Li & Tan (2005)	Prioritizing Sensors based on their residual energy/Dynamic adjustment of super frame parameters	Reducing energy cost
	Ghildiyal et al. (2011)	Rapid data transmission at the end of sleep period	Better energy efficiency performance
Exploiting General MAC Parameters			
Dynamic Super Frame Structure			
Lack of efficiency in addressing different traffic types (TDMA/CSMA techniques balance)	Zhang & Dolmans (2009)	Different channel access schemes for data/control traffic	Providing differentiated QoS to different applications (prioritizing medical applications)
	Liu et al. (2011)	Dynamic adjustment of CAP length based on occurrence of collisions	Increased throughput
	Zhisheng et al. (2012)	Adjustment of transmission order in super frame based on channel information	Better energy performance
	Zhuo, Song, Wang, & Wang (2012)	Variable TDMA period in super frame	Better throughput
Dynamic Duty Cycling			
Energy-data rate balance/addressing sudden traffic changes	O'Donovan et al. (2009)	Dynamic duty cycling based on patient's position information/Transmission interrupts based on message priority	Better delivery ratio/prioritized data transmission
	Zhisheng & Liu (2011)	Dynamic duty cycling and data rate	Improved latency for time-critical applications
MAC Layer Queue			
Lack of traffic awareness	Ghaboosi, Pahlavan, & Pomalaza-Raez (2011)	Considering MAC queue utilization/Residual energy	Less delay endured
	Zhou et al. (2012)	Dynamic duty cycling based on queue length	Better throughput
Exploiting PHY Layer Parameters			
Changes in RSSI Levels			
Lack of PHY layer parameters incorporated into MAC layer data scheduling	Prabh & Haur (2011)	Packet transmission during high RSSI windows	Better throughput

Sections 3.1 to 3.3 summarize some of the traffic adaptive methods exploiting diverse parameters both at MAC and PHY layer into a dynamic approach with different target QoS metrics to achieve.

3.1 **Exploiting MAC Parameters in SuperFrame Duration (CAP and CFP)**

This section summarizes some of the works in the literature that have done extensive analysis on the CSMA/CA algorithm of IEEE 802.15.4. It is an important part of this study to realize and clarify the effect of different traffic parameters or configurations of the MAC protocol on the performance of the whole network in different terms such as reliability, energy, or delay. This section of our literature review has given us the best impression for choosing the inputs of our fuzzy system later in Chapter 4.The effect of each chosen parameter here and the effect of each on different network performance parameters are studied. In other words, the application requirements may vary from needing less delay in time-critical situations to having less energy consumption no matter how late data packets may arrive at the destination, or sometimes its requirements could be a combination of both criteria emphasizing both time criticality and energy at the same time. In some prolonged and low data rate monitoring applications the main concern will actually be shifted to manage a fair access among the few nodes of a small WBAN to provide optimum reliability/delay and energy behaviors. Organizing a fair access to a shared channel can become most helpful in achieving the target performance parameters, especially where nodes have been deployed with their predefined application types that do not vary much over time. This has been more elaborated in Chapters 4 and 5. Whatever the desired trade off is MAC parameters can adapt themselves uniquely to the application's specific needs. Different methods of MAC parameter configuration distinguish the different tradeoff they are meant to make but they may be common in the MAC parameters they exploit.

Our aim is to illuminate the significance of the chosen inputs for the fuzzy algorithm in Chapter 4 and we categorize this section based on the main MAC parameters in the CSMA/CA algorithm which were earlier classified as "protocol-specific" parameters in Chapter 2. The studied literature in Section 3.1 does not include the non-beacon-enabled mode of access so all the studied works focus on the slotted access, which incorporates the beacon-enabled-mode of access. Later in Sections 3.2 and 3.3 the effects of general MAC parameters (out of the scope of CSMA/CA) and PHY parameters (communication links' characteristics) are explored.

Section 3.2 takes a look at different methods that might exist in both TDMA and CSMA-based MAC protocols in the recent literature.

3.1.1 Dynamic Evolution of CSMA/CA Parameters in CAP

The two main problems originating from the BE parameter in the CSMA/CA process can be summarized as:

1) With a small backoff exponent (BE), the amount of random generated backoff duration is proportionally small; this therefore fits well into low traffic situations whilst it increases the probability of producing more identical (equal) backoff durations for a high traffic.

2) With larger backoff exponents, the energy will become a more serious concern since, although the backoff durations are generated less identical, they are bigger in length and make nodes spend more time in a backoff state when they are prompted to enter one.

In Muneer Bani, Marwa, Wail, and Khamayseh (2012), the random nature of the binary exponential backoff (BEB) mechanism of the CSMA/CA in IEEE 802.15.4 is said to be blamed as the main source of collisions that originate from an insufficient distribution of numbers in a small range. Whilst the authors take away the randomness from the BEB algorithm by replacing it with a Fibonacci Increment, in our MAC approach later discussed in Chapter 4 we improve the algorithm's performance by dynamically adjusting the length of the interval that the random backoff value is going to be selected from. In some other works the minimum and maximum values for the backoff exponent are dynamically adjusted so thatfewer collisions occur in the CSMA/CA algorithm. In Ko, Cho, and Kim (2006), for instance, the default minimum value of BE (Attributed in *macMinBE* variable in the standard) is suggested to vary based on the changes in the network traffic. The range of changes for the *macMinBE* variable in the standard is revised from a minimum of three to a variable minimum between one and three. In other words, the *macMinBE* values will range from one to three. The state of the node could be any of "no data", "post data", and "send data", and that is why their method is called a *state transition scheme*. The value of macMinBE takes on a value from one, two, and three based on the state transition of the node. Although the simulation results show a better performance compared to the standard's original MAC, the protocol cannot outperform the standard in high loads; this is because setting the *macMinBE* variable to smaller values will of course generate

smaller backoff times, but the disadvantage reveals in higher traffic congestions where such a small generated backoff could be a source of more collisions to happen.

In Hadid, Guitton, and Misson (2009), a wise variation of the *macMinBE* value is presented in which the consistency of the MAC protocol's performance is looked at both for low and high traffic loads by a dynamic reasonable change of *macMinBE* over the length of the CAP from the start of the interval to its end. In their method, an adaptive CSMA/CA algorithm is proposed to compensate for the high level of packet loss and collisions occurring in the contention-access period. The authors claim that the packet queue for each node keeps on growing mostly in the inactive period of the beacon interval in the slotted access and will impose more traffic pressure on the active period. The main contributions of their work are a better goodput and delay performance. The state of the node's queue, i.e. its mean behaviour, has been used as one of the main parameters in configuring the CSMA/CA. The node queue is assumed to have a capacity of not more than 12 packets. The main difference between this work and the works studied in its literature is that they assess a beacon interval in which both active and inactive periods exist, unlike many analytical researches where only the whole beacon interval consists of an active period. When only an active period exists, the super frame order (SO) and beacon order (BO) are equal and the maximum load of 250kbps can saturate the network. In order to make this interval consist of both active and inactive periods, they devote only 50 percent of the whole interval to each period with a maximum load of 125kbps. The evolution of the node's queue is studied when the traffic is meant to start only at the beginning of the sleep (inactive) period so that all the generated packets get queued. In a small number of consecutive time intervals of a length of 1/40 second, the "mean number of failures" in accessing the channel for two different data rates of 64kbps and 128kbps was studied. This shows the different distribution in this traffic adaptive variable in both low and high loads. In the case of a low load, the distribution is not uniform, meaning that it starts with a peak value and takes on a decreasing trend as the node reaches the end of the CAP. However, this trend for a higher load does not vary much since the distribution of the channel failures is almost the same. This concludes that in higher loads the demand for having access to the channel always stays high whilst it has a decreasing manner for lower loads as nodes find nothing more to transmit once they approach the CAP's ending and therefore they undergo fewer failures. Hadid, Guitton, and Misson (2009) integrate the "state of the node's queue" into initialization of the parameter BE. The CAP of the beacon interval is divided into five distinct and disjoint intervals with indices 1–5. The transmission time for a packet at any given time will fall into only one of these

intervals, the closer the interval is to the end of the CAP, the smaller the value of BE will be.That is because it is initialized according to: $BE_{initial}=7-i$, where i is the index of the intervals. This tuning of the initial BE makes nodes not back off if they are approaching the end of the CAP in a low load. However, this trend is not useful for high loads as the queue is full with data packets but, as time passes to the end of CAP, the BE values get smaller and act weakly to serve the queued packets. The second enhancement on the original CSMA/CA algorithm is that they put a condition before every time the BE needs to be incremented in case of a busy channel; the BE is only incremented if the channel is busy and the queue state is not empty. In case of an empty queue the value of BE gets decremented. Here is a quick look at the algorithm (Figure 3-1):

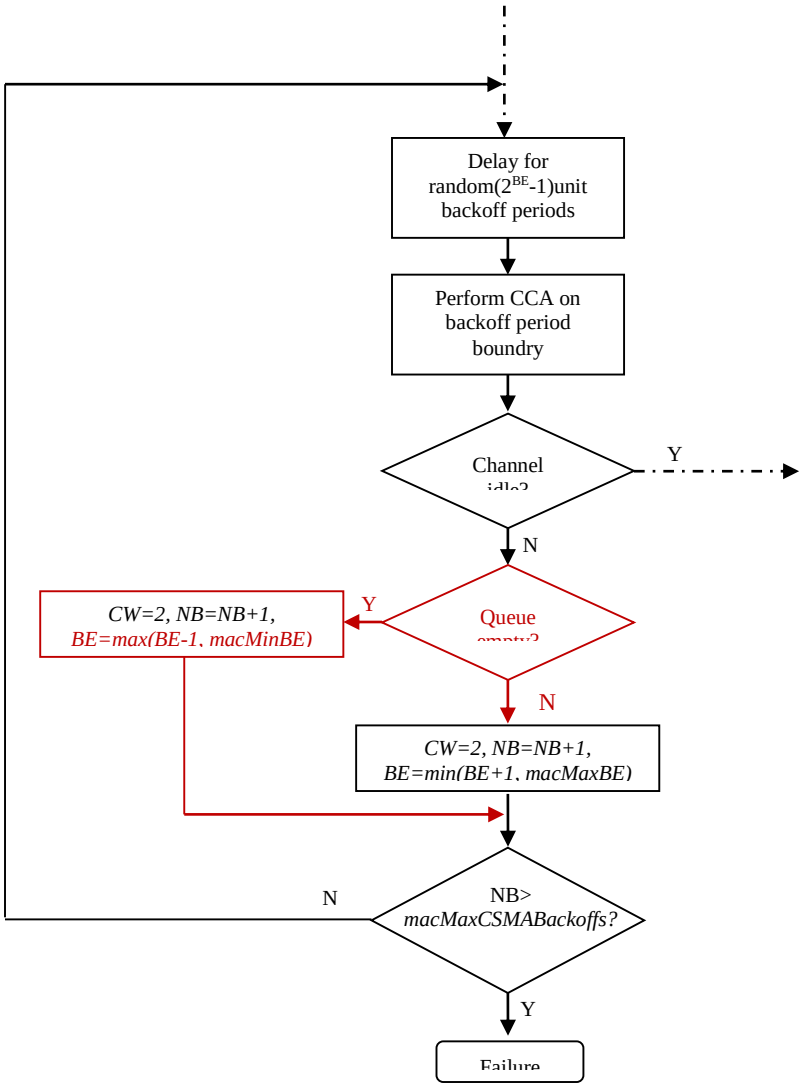


Figure 3-1: Additional steps added to the algorithm (highlighted in red)

Park, Fischione, and Johansson (2010) targeted minimization of energy consumption as well as a reasonable level of reliability and delay constraints. The Monte Carlo simulation results are compared to the analytical results based on the MAC model in Bianchi (2000). Two scenarios of transient and stationary conditions were considered for the simulative part, and they discussed the real implementation of the improved MAC as well. The relationship between the protocol parameters and power consumption, reliability, and delay was clarified through extensive analytical validations and the optimal operating points were characterized using simple approximate relations. Power consumption was considerably decreased without sacrificing other important parameters reliability and delay. An unsaturated traffic condition is utilized for their Markov chain model leading to a more precise modelling of reliability, latency, and energy consumption. The optimization problem is declared based on the evaluated Monte Carlo simulation results; formulation of the optimization problem is based on the energy consumption as the objective function. The scheme is being stated as a channel state based method where the estimation of busy channel and channel access probabilities are merged into the backoff mechanism to produce a self-adaptive contention MAC algorithm in which energy is minimized without affecting reliability and latency requirements. The amount of random backoff is reported to be dependent on data traffic, MAC parameters, and the network topology. A simple closed loop (shown in Figure 3-2) is depicted for each sensor that integrates two main variables of "application's needs" and "channel's state estimation" to produce the optimized MAC parameters. Application needs range from reliability to delay and battery life time. The channel states could be the probability of a busy channel or the probability of sensing the channel based on which the optimal MAC parameters are computed by using an optimization problem with power consumption as its objective function.

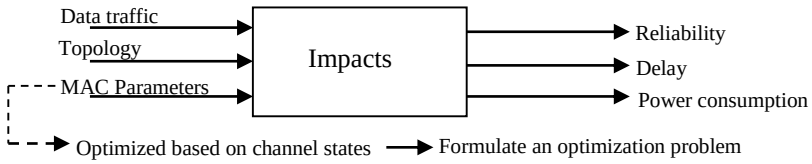


Figure 3-2: Closed loop for providing optimized MAC parameters

The MAC parameters to be optimized are *macMinBE*, *macMaxCSMABackoffs*, and *macMaxFrameRetries*. These three MAC parameters must be optimized in the optimization function in their work as mentioned earlier, therefore the optimization problem consists of three mathematical expressions of the energy consumption (as the object function) and delay and

reliability. The paper by Park, Fischione, and Johansson (2010) discusses in detail the formulation of these three parameters with mathematical non-linear equations. After the three parameters of energy, delay, and reliability are formulized they can be substituted in the optimization function to yield to optimized values of energy, delay, and reliability. However, the evaluation of performance metrics by means of complex computations is not a feasible approach to be processed on the scarce resource sensor nodes and hence they recommend approximated expressions to be evaluated by sensor nodes rather than the complex non-linear equations. These approximations are carried out as local measurements on sensor nodes to estimate the channel states and evaluate the performance metrics. Figure 3-3 summarizes their technique:

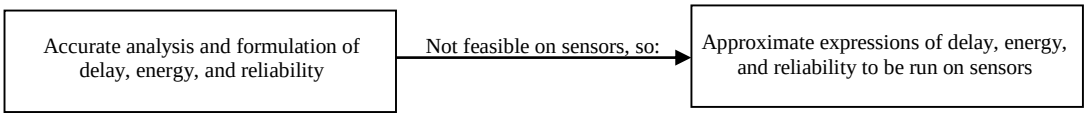


Figure 3-3: Achieving optimized values of energy, delay, and reliability

The paper shows the results of Monte Carlo simulations of only the approximated expressions of delay, energy, and reliability that are done by the sensors. The effect of MAC parameters *macMinBE*, *macMaxCSMABackoffs*, and *macMaxFrameRetries* is investigated on delay, energy, and reliability levels through different traffic conditions and number of nodes. Better results are obtained for low traffic than high traffic. It fails to match the simulation results in higher loads and with a larger number of nodes.

In Prakash Rao, and Marandin (2006), the inefficiency of the protocol is described by producing short backoff periods in high congestion situations. They reveal an exponential increase in the number of packet drops for higher data rates. The small range of backoff exponent which only varies from three to five is reported as the primary reason behind identical generated backoffs. Their method contributes to a more reasonable distribution of the BE values over the beacon interval by choosing larger BEs to reduce the probability of producing identical backoff durations. This, however, is done by carefully taking into account the energy consumption of the protocol so as not to let the bigger BEs deteriorate the energy efficiency behavior. The *macMinBE* variable of the standard (the minimum value of BE) is designed to be variable as time goes on; the devices imposing more traffic to the network will be assigned bigger *macMinBE*s and the devices contributing less traffic will have to pick smaller

macMinBE values. The analysis of the traffic imposed by each node is done at the coordinator's side and the decision for the next taken minimum value of BE for that node is also made by the coordinator. The coordinator is, however, prompted to analyse each node's contributed traffic during a cycle of time named as "analysing cycle", which for their implementations is assumed to be three consecutive beacon intervals. The main metric here to be counted by the coordinator during an analytical cycle is the "number of packets received per node". Although the method works well, it might only be suitable for applications where there is no need for data prioritization and where time criticality of some data streams does not matter. The only metric used in the calculations of the next assigned *macMinBE* variable is the "number of received packets", which dictates the next *macMinBE* variable regardless of its time-critical or high-priority data. Although sporadic time-critical data might gain fast access to the medium by the aid of GTS slots, for two reasons a GTS allocation might not always perform its best:

- First of all, the GTS requests for time-critical data packets will be processed during the current beacon interval but allocated during the next beacon interval
- Secondly there is limited number of them available for each interval.

The research of Golmie, Cypher, and Rebala (2005) answers some useful questions in regard to the concern of enhancing the performance of a WBAN through different protocol adjustments. The protocol parameters that can be used to improve the performance were investigated as the backoff exponent and also the *macMaxCSMABackoffs* defined earlier in Table I in Section 1.7.1.1. The main difference of this work is that it was conducted on the unslotted version of IEEE 802.15.4. The reason for that has been claimed as being less complex and also having less overhead due to the absence of beacon packets. The paper shows how reducing or increasing the maximum default for the number of permitted backoff trials for each node (*macMaxCSMABackoffs*) has an effect on the delivery ratio of the protocol. In the CSMA/CA algorithm of the IEEE 802.15.4 it is shown how lesser values of this parameter are suited better to a low data-rate WBAN, especially when the number of sensor nodes increases in the network. Even when the parameter is set to zero, which actually means each node can only have one attempt to have access to the channel, the protocol performs better than a default value of three for this parameter. The effect of the values assigned to the backoff exponent parameter is also investigated in the paper. The minimum and maximum values assigned to BE in the standard are three and five respectively. The paper suggests a BE value of one to

outperform the default value of three in the standard in terms of goodput. Once BE is set to zero, the best delay performance is achieved followed by BE values of one and two as compared to three. The paper also concludes how limited utilization of the medium is a result of the improper default parameter initialization. This can be addressed by tuning certain parameters of the protocol to reflect the requirements and characteristics of the application, such as smaller values of BE for a low data rate WBAN.

Another example of parameter enhancement in IEEE 802.15.4 CSMA/CA is (Koubaa, Alves, Nefzi, & Song, 2006), which is specifically designed for time critical applications. A queuing strategy was used for data and command traffic, which are defined as low and high priority traffic types respectively. The protocol parameters to boost the CSMA/CA performance are CW_{init} (which is the same concept as NB), *aMaxBE* (maximum default value for BE), and *macMinBE* (the minimum default value for BE). Different values of these parameters are assigned to the two different classes of low and high priority data. The classification of the different traffic types is done by a priority queuing strategy technique to differentiate these two types of traffic from each other. The same as the concept of our fuzzy system later described in Chapter 4, longer backoff times are assigned to low priority data as they can afford a longer waiting time in contrast to high priority data. The CW_{init} is set to larger values when the data sent is classified as low priority. The "responsiveness" of the protocol was improved by letting *macMinBE* of high priority data (denoted as $macMinBE_{HP}$) be smaller than *macMinBE* values for low priority data ($macMinBE_{LP}$). They evaluated their differentiated service approach in OPNET ("OPNET Simulator v11", 2014) which is also the simulator used for evaluations in Golmie et al. (2005).

3.1.2 **Dynamic GTS Allocation in CFP**

Most of the works reviewed in this section propose solutions to the bandwidth utilization problems of the current GTS allocation mechanism in IEEE 802.15.4 which resembles the fragmentation problem in an operating system's memory allocation. When an operating system's memory is fragmented, the unused memory is broken into small parts where no individual part is large enough to help execute a task. The same phenomenon can happen in GTS allocation of a super frame structure. Since the allocated GTSs will be in the order of units of times (a time slot), no allocated GTS can be less than a time slot. This causes bandwidth utilization problems when the generated traffic is so low that the entire allocated time slot is not consumed. The rest of a time slot might not be enough to execute a task but the sum of

these partially unused time slots could possibly address the bandwidth requirements of a node with emergency data. The reliability of GTS allocation in IEEE 802.15.4 is therefore mostly questioned for not efficiently servicing the emergency traffic.

There have been many studies done in this area but we limit our discussion to a few selected ones here in this section. In Li et al. (2009), two major problems of IEEE 802.15.4 discussed are its low support of simultaneous data transmissions and its low level of reliability, which have forced the need for a WBAN-specific standard ("IEEE P802.15.6/D01", 2010). The average packet delay in transmitting each packet is reduced by not postponing the GTS requests to the next super frame but sending them in the current superframe. The dynamic part of the superframe in their approach is the duration of the CFP, which changes according to the traffic.

As stated in Section 1.7, two topologies that IEEE 802.15.4 supports are star and peer-to-peer (cluster tree is also a type of peer-to-peer). The MAC operating modes are beacon-enabled (slotted) or non-beacon-enabled (non-slotted). We do not have RTS/CTS but we have ACK, data, beacon, and MAC command frame. Nodes in IEEE 802.15.4 can communicate in three different ways: direct, indirect, and via GTS. If a node needs a GTS, it must send the request to the PAN in the CAP of the current superframe and the PAN, if it has enough GTS to allocate, will allocate them by the next beacon interval, not the current one.There are two major deficiencies in the above method of IEEE 802.15.4 as Li, Hao, Zhang, Liu, and Li (2011) claim: 1) constant lengths of active and inactive periods; for example if there is no GTS traffic, then the CAP will continue till the end of the active period (where the CFP ends), and sometimes even if not needed or sometimes CFP seems to be very small if there are many GTS requests; 2) current GTS requests are postponed till the next beacon of the next superframe. In their approach, such requests are transmitted by the current notification frame rather than the next beacon frame so GTS allocation information is not included in the beacon frame. They simulated their work based on ideal situations where there were no propagation delay or bit errors in the channel or interference considered, which is not realistic for WBANs. The number of nodes is another issue.We normally do not put ten nodes on a patient's body, probably 3–5 would be enough to monitor a patient's vital signs: some sensors even do multiple sign monitoring. Li et al. (2011) introduced two different scenarios, each scenario taking a different response time to the GTS by the coordinator. Since "CFP cannot operate independently and is always integrated with CAP", if a node wants a GTS, it sends its REQ to the coordinator during CAP and then the coordinator answers that request in a broadcast frame called "notification",

which is sent in the current superframe. So the GTS information will not be sent in the next beacon frame. If there had been no GTS traffic beforehand, the notification frame allocates the GTSs it has immediately after the CAP, but if there are some GTS slots beforehand, it will allocate the GTSs it has after the pre-existing GTSs. This work is similar to that of Boulis and Tselishchev (2010), where a poll slot is assigned in the current superframe by the future poll message without postponing it to the next frame.

In a recent study, a new MAC protocol for WBAN is introduced as MEB MAC (Huq et al., 2012), which is based on the IEEE 802.15.6 MAC specification. The protocol is mainly designed to address emergency traffic for medical applications while maintaining other quality of service parameters such as throughput at a desired level. The authors claim that the GTS allocation in CFP of a super frame (which for IEEE 802.15.6 is managed access period (MAP) for high priority data transmissions) is not always optimal. A rapid transition of emergency traffic to normal traffic exists when an event (normally an abnormality in the received signal) occurs in a vital sign monitoring. The emergency traffic usually happens for a short timeand therefore the pre-allocated time slots to a specific sensor node might not always be completely used. This phenomenon leads to less throughput of the system and can be prevented by a dynamic allocation of listening windows in managed access period (MAP; as introduced in IEEE 802.15.6) wherever applicable. The data in a WBAN was prioritized into four different groups, and different backoff lengths were assigned to each group. Shorter backoff periods would be given to nodes of a greater priority to be relayed on channel at the earliest possible time and vice versa. The deficiency of the IEEE 802.15.6 MAC is the longer delay it goes through when it has emergency data to deliver during MAP periods (MAPI and MAPII). The longer the super frame length is, the longer the produced delay when handling the emergency traffic will be for the IEEE 802.15.6 approach. The MEB MAC approach to utilize the unused bandwidth by adding listening windows into the MAPs (acting as guaranteed time slots) reduces the delay in channel access and consequently increases the throughput performance, especially when emergency traffic is frequent. More emergency traffic can be addressed by the added listening windows in their proposed algorithm.

In Cheng, Bourgeois, and Zhang (2007), the inefficiency of the GTS allocation in IEEE 802.15.4 is explained firstly through the highly dynamic traffic presented by specific applications. This will lead to some pre-assigned GTS allocations not being sufficient to accommodate the suddenly increased GTS demands. On the other hand, in some occasions of

extremely low traffic it could also lead to a waste of bandwidth when there is no traffic during the GTSs assigned. The GTS allocation in cases of no dynamic traffic (for example CBR traffic) has also been declared as inefficient since the allocation of them is based on units of times and not all the time will the entire allocated GTS slots be used. The two main issues arising from the unfriendly bandwidth utilization of the current GTS allocation in IEEE 802.15.4 are summarized as: 1) variable bandwidth demand among all the sensors in the network; and 2) variable bandwidth demand of each sensor from time to time. One way to address the partially used time slots in the CFP is to use smaller time slots compared to the base time slots used in the original IEEE 802.15.4. For this purpose, the GTS descriptor format of the base standard must be altered accordingly. Therefore, whilst the GTS allocation in the IEEE 802.15.4 standard is only composed of five time slots (two time slots for the first GTS and three time slots for the second GTS), their approach divides the CFP into 16 time slots, each being smaller than the time slot length of the IEEE 802.15.4 CFP. The 16 time slots in CFP provide three GTS allocations each with a different number of time slots.

In Ying Lei, Choi, Park, and Rhee (2012), the first-in-first-served (FIFS) nature of the GTS allocation in IEEE 802.15.4 is believed to be problematic in servicing emergency traffic. A possible shortage of GTS slots (also referred to as GTS starvation) is deemed to happen mostly for higher data rate applications, which will lead to data being lost. A reliable GTS allocation mechanism is therefore proposed for emergency traffic that focuses on applications in WBAN. Different physiological sensor nodes are classified into three classes (very low, low, and high) based on the data rates assigned to them. For example, ECG/EEG/EMG, the accelerometer, and the gyroscope have been classified as high data rate applications compared to temperature and humidity as very low traffic applications. Two new fields in the GTS descriptor in their method are added that carry information about the emergency data and the class (level) of the data rate. The emergency data is basically a field that can be set to either zero or one to represent normal or emergency data traffic type. The lower data rate nodes will be asked by the coordinator to surrender their GTS slots if a high data rate node requires one based on the GTS request that they have sent to the coordinator. This will only take place when the emergency data field is set to one. An OPNET simulator was used for their experiments, which show a better throughput performance by a GTS allocation algorithm. The late arrival of the emergency data does not deprive the nodes with emergency data from having prompt access to the GTS slots, as opposed to IEEE 802.15.4 which basically keeps the nodes waiting until the coordinator processes their request in the coming beacon intervals.

3.1.3 **Dynamic Super Frame Order and Beacon Order Values**

Li and Tan (2005) propose an adaptive MAC protocol based on feedback from sensors. The information that feedback packets carry is used in a closed-loop control. The topology used is star, but it also supports peer-to-peer topologies. Li and Tan assume that the coordinator (smart phone) is not power- or resource-restricted (as so close to reality). They prioritize (categorize) the sensors based on their residual energy and tune the MAC accordingly to come up with a MAC protocol that can save energy on nodes. The interesting concept of this paper is that they use "feedback information from sensors to form a closed-loop control of MAC parameters". The fuzzy-logic system of our thesis will act similarly as a controller in our system discussed in Chapter 4. However in their paper, they analyze some parameters to give different priorities to different sensors. Contrary to the fuzzy logic used in our method, they take advantage of the feedback mechanism in control theory. The feedback information is carried by the request (REQ) messages to the coordinator, the coordinator evaluates them (Pmac, RBI, RBN) and will change the SO and BO and CRBN accordingly so the superframe structure is changed dynamically. The MAC priority is formulized based on some real-time parameters, such as remaining energy in each node, remaining buffer, time criticality, energy level of the node. There is a tradeoff considered between delay and energy on sensor nodes. Based on the priority, the MAC gets its superframe parameters dynamically adjusted. The priority that is gained for each sensor will be sent to the coordinator by the REQ packets that are sent from sensor node to the coordinator asking for a data slot to transmit. So once the coordinator knows the priorities of the multiple requests that it has at hand, it will listen to the one with the highest priority and may change its beacon interval accordingly.

In summary: on each sensor, the MAC priority based on the sensor node's specifics will be calculated and then sent as a REQ to the coordinator. The coordinator will choose the highest priority one and will change the superframe parameters (number of beacons and the length of beacons for the current request) accordingly. One interesting interpretation of their results is that "using larger beacon intervals and packet sizes can reduce the energy cost dramatically; this is because the overhead of control packets and extra beacons is reduced".This concludes that sensors with a smaller buffer will consume more energy than sensors with a larger buffer when they want to transmit data of the same size.

The “determination of frame parameters”, such as beacon interval, is proposed in Ghildiyal et al. (2011). The paper aims at MAC protocol design specifically for implantable WBAN scenarios to address priority, latency, and throughput. The effect of beacon interval length on the amount of produced delay for high priority packets is discussed. The appropriate beacon interval length for medical applications manifests itself more when making a balance between two main performance parameters: latency and energy. While a larger beacon sleep interval (as part of the beacon interval) favors energy efficiency, it will definitely add more to the experienced delay. Therefore the authors considered a “rapid” transmission of data at the end of the sleep period. Generally, the longer the duration of a super frame is, the better the energy efficiency performance of the MAC protocol will be, and on the other hand a short duration of a super frame would lead to a better latency performance at the cost of more energy consumption (Huq et al., 2012).

3.2 Exploiting General MAC Parameters

3.2.1 Dynamic Super Frame Structure

In Zhang and Dolmans (2009), the data rate for medical applications normally ranges from 10kbps to 100 kbps and the traffic is periodic, not streaming or bursty. In their approach, they split the channel into a “data” channel and a “control” channel. The access scheme in the control channel is “contention-based” and in the data channel is “schedule-based”. The performance of their proposed approach is compared to that of IEEE 802.15.4. The requests for the GTSs in the CFP are sent during the CAP and based on a CSMA/CA scheme. The parameters in a slotted CSMA/CA are BE, NB, and CW. A node that needs to occupy the channel must choose a backoff time in between zero and 2^{BE}. When the backoff time is finished, it must perform the CCA for the duration of CW number of backoff times. (So CW is actually the number of backoff times the channel has to be sensed as idle.). The superframe structure is modified by placing the CAP in between two CFPs. The first CFP (CFP1) is reserved for “periodic” traffic and the second CFP (CFP2) is reserved for “bursty” traffic. Having two channels actually means that the data will be sent during the CFP and the control messages will be sent during the CAP. The control channel part is further divided into Active Channel1 (AC1) and Active Channel2 (AC2). The control channels work based on ALOHA, and nodes randomly select a time slot in that to send their requests. Each of the AC1 and AC2 control channels is reserved for different applications. AC1 is only used for medical application requests and AC2 is only used for consumer electronic (CE) application requests. The

coordinator is programmed to identify the application category. So they add algorithms on the coordinator to provide differentiated QoS to different applications. That means that the coordinator will give a higher priority to medical traffic if the traffic becomes high. Data channels are allocated on demand, based on the type of the traffic (traffic characteristics). The main benefit of their approach is that they do not let the periodic traffic resource allocation be influenced by the other types of traffic in the network. For this reason, they make the resources allocated to the periodic traffic remain intact while the control channels (AC1 and AC2) are adaptive to the traffic. The main difference of their approach to the IEEE 802.15.4 is that the control channels (because they are application-specific) take care of data channels. So control channels are traffic adaptive to make data channels collision free. Tradeoffs made on the length of control channel are as follows:

- Fewer time slots on the control channel will lead to more collisions
- More time slots on the control channel will lead to waste of resources.

And tradeoffs on the number of backoff times are as follows:

- The more the backoff times, the less timeslots we get on the control channel
- The less the backoff times, the more timeslots we get on the control channel (more resource efficiency).

The concept based on which the length of the control channel changes adaptively with regard to traffic is that the coordinator has the knowledge of the total number of nodes in the network, by knowing that and some other parameters such as the traffic arrival rate and the duration of a superframe, a coordinator will be able to calculate the probability of a successful contention and a successful access. Therefore, it can monitor the access contention on the control channel and be aware of the traffic load. This is actually based on the rate of collision or idle times on the control channel that the coordinator changes the control channel length accordingly. Simulations are done by Monte Carlo in Matlab and the topology is star.One of the disadvantages of their proposed work as also reported in Zhisheng & Liu (2011) is that it has a complex super frame structure that does not also consider the emergency traffic.

In Liu et al. (2011), the length of the contention period is evolved based on the number of consecutively dropped packets and the time spent in collisions. Collisions actually represent the inefficiency of the CSMA/CA algorithm and therefore are used as a measure to reduce or increase the length of the contention period together with the number of dropped packets. The number of dropped packets, on the other hand, represents the inefficiency of the TDMA approach in handling the failed transmissions in a deep fade affected channel. This

phenomenon has already been described in Section 1.9.1 and here the method is examined more closely. All nodes are to calculate a parameter individually during their current beacon frame, which acts as a measure to describe the proper length of the contention access period feeding the traffic each node has been through. The coordinator then collects this parameter from each sensor node, makes an average of it, and makes a final decision on the length of the next contention access period of the next beacon interval. Putting together the trade offs of both access techniques, TDMA and CSMA, the authors came up with a locally calculated parameter on each node that makes a balance between the unpleasant consequence either TDMA or CSMA may exhibit in different network conditions. The calculated averaged parameter in the previous beacon interval is also taken into account for the calculations of the next averaged parameter. By doing so, they can investigate the chances of having an unchanged contention length in case of very slight changes in the network traffic. There is an option of emergency traffic coming through from nodes, which makes the coordinator generate a new beacon packet with the demanded slot allocations in the next frame. One of the main issues with the work is that the deep fading in the channel is not modeled realistically. Deep fading periods are chosen randomly and there is no particular explanation on how the length of the deep fades in channel has been set. The maximum fading duration is 50 ms and the smallest is set to 30 ms.

In Zhisheng et al. (2012), the same group of authors tries to eliminate the need of combining a CSMA approach with a TDMA one as described in Liu et al. (2011) so as to provide the best energy performance for WBAN. A thresholding scheme is proposed to adjust the transmission order in the super frame. It is based on the channel status that enhances the delivery quality for a node. The number of slots that can be allocated to each node for its data delivery is also controlled and optimized by a scheduling algorithm based on its QoS and energy requirements. This slot allocation in the super frame is done at the beginning of the beacon interval and is carried out by the coordinator, which looks into each node's QoS demands and its most recent channel history for slot assignment. Disregarding the transition times when nodes change their modes from sleep to transmit or vice versa makes the measurements not so reliable in their method. The channel state has been modeled with a two-state (good or bad) Markov chain in which nodes can only send if the channel state has been assessed as good, otherwise it will be considered a fail. Transition probabilities are then defined for the sensors based on their previous transmission state (a successful or a failed transmission over the channel). The probability of being in a good or bad state after an interval of time based on the previous status is then determined. It is shown in their model that there will be a monotonic increase in the

state transition of a node from a bad state to a good state after a time interval (denoted as τ). The longer the interval τ is, the higher the delivery probability will be. Channel condition assessment in their method is, however, an estimation rather than obtaining real channel information. At the end of the current super frame, therefore, two different sets of nodes will be seen in the network: nodes whose last state had been a bad one and will be transmitting to a good state, and nodes whose last state had been a good one and will be transmitting to a bad state. The behavior of each of the two types of transiting nodes will be monotonic, which means either a monotonic increase or a monotonic decrease in their transition probability. Higher delivery probabilities are actually introduced for the nodes whose very last trials in the previous super frame had been unsuccessful but had been in a good state at the initial stages of the super frame. This is because they are now considered to transit to a good state again more possibly than the nodes whose initial state in the previous super frame had been good. These nodes will be scheduled in the former part of the next super frame as they most probably will be transiting to a good state at this point. A threshold concept is defined to make sure that all the transmissions fall into the scheduled time slots. The total number of allocated time slots to the node to transmit its data can also be optimized based on its data rate. In fact it should only be big enough to satisfy the node's traffic characteristics.

A dynamic super frame structure has been proposed in Zhuo, Song, Wang, & Wang (2012) that takes advantage of a variable TDMA period in its super frame structure. The protocol incorporates the number of packets in the MAC buffer to decide about the number of guaranteed time slots to offer to highly loaded nodes. The protocol is discussed further in Section 3.2.3.

3.2.2 **Dynamic Duty Cycling**

Duty cycle specifies the length of the sleep period for sensor nodes' radio and is defined as the ratio between the listen interval and the frame length. The longer the sleep period is, the smaller the duty cycle will be. The length of the duty cycle has a direct effect on how the network handles delay and how fast the data packets can travel throughout the network to their destination (Mouzehkesh & Zia, 2011). Duty cycling is a technique that was invented to control the idle listening phenomenon in WSN that was identified as the dominant source of energy consumption compared to overhearing, collisions, and control packet overhead (Ye, Heidemann & Estrin, 2002). However, the main problem with all the duty cycled protocols is the ability to handle the sudden changes in the current traffic. Although duty cycling may

reduce the power consumption to a reasonable extent, it is also said to be of finite advantage when too much data is buffered during a node's inactive period in its normal duty cycle if its data rate is high (Ghildiyal et al., 2011). In this case, the node's MAC queue becomes saturated during sleeping periods and a longer active period would be needed to send out the buffered data, which imposes more latency. The comparison of the average current drawn for two different sensors of high and low data rates (in their case PEA sensor with data rate of 10 Kbps and temperature sensor with a data rate of 0.2 bps) shows how increasing the duty cycle has different effects on the energy consumed. For a data rate as high as 10 kbps increasing the duty cycle would only continue to decrease the energy spent up until a certain duty cycle, whilst in case of the lower data rate assigned to the temperature sensor, the energy continues to decrease to even a highest duty cycle of 100 seconds (100 seconds in sleep mode). This illuminates the fact that tuning the duty cycle also highly depends on the data rate of a sensor. Although it is less visible in applications of WBAN since the sporadic traffic is less experienced, it still applies to some specific classes of medical vital sign monitoring where the associated traffic is highly unpredictable. Sudden fluctuations in high sporadic traffic require the sensor nodes to stay awake for possible reception of emergency traffic, which leads to idle listening and consequently more energy consumption. TDMA approaches are deemed to be free from idle listening (Ghildiyal et al., 2011), which makes it perfect for WBAN implementations but only at the cost of synchronization, which also imposes some energy spending. In general, when the traffic behavior of a network is bursty or sporadic, a low duty cycle will not function best. This has motivated many dynamic and traffic adaptive duty cycling techniques to support sudden traffic changes of such applications.

O'Donovan et al. (2009) discuss three different management techniques: 1) priority—emergency sensor readings have higher priority to be sent than other sensor readings; 2) aggregation—some messages will be aggregated and will be sent in one message; and 3) adaptive duty cycling. When a change happens in the body, the sampling rates of the sensors will change accordingly. The traffic is dynamic. The low-duty cycle is for when the patient does not do anything, and when a patient changes position or has a higher pulse, the "data rate" and the "duty cycle" change. It can also change the sampling rate so that more data get logged. In O'Donovan et al.'s implementation, two nodes are equipped with accelerometers and when they detect a change in position the sampling rate gets altered. Their CSMA-based protocol in use has three major characteristics: 1) it performs adaptive duty cycling; 2) it has opportunistic data aggregation; and 3) it has message prioritization. So the node's features are message

rendezvous as explained in the paper, high priority interrupt, opportunistic aggregation, and adaptive duty cycling. First of all, they correct the defect in the CCA mechanism by using a "pre-send listen". By comparing some attributes represented in the messages, a waiting node that wants to transmit will decide if it has to interrupt an ongoing transmission or not. They make interrupts in the transmissions based on the message priority but it is not explainedin the paper how this priority is given to them. While it is mentioned that the messages have time constraints attached, the authors claim that it is really hard to schedule the messages' delivery based on priority, especially in an event driven network. The messages are scheduled on the order of their priorities and an interruption concept is used to interrupt those with lower priority. Later an aggregation method is described that is regarded as "packet stuffing". Data aggregation for networks with bursty traffic is strongly recommended. The adaptive duty cycle works based on the changes in the number of messages generated, if the number of messages is increased then it will reduce the sleep interval to accommodate them. Data generation of each node should be consistent with other corresponding nodes. Two accelerometers are attached to two sensor nodes (Leg and Torso) so if a change is detected in position, the state gets RED, sampling rates of these two nodes are increased for two minutes, and the duty cycle of the central node (ECG)/gateway is increased (less sleep delay). Other nodes suspend sampling. They tested their implementation with Moteiv Tmote Sky and attach those sensors to their Moteiv Tmote nodes. The correlation among the decisions made in this paper is similar to that of Lorincz et al. (2009), which is all about when to suspend and resume the accelerometer and gyroscope readings, when to do throttle download and all these controlling functions.

Zhisheng & Liu (2011) proposed a dynamic duty cycle and data rate to improve latency requirements according to the current traffic. The authors emphasized the timely delivery of highly prioritized data rather than ordinary data as a means to guarantee patients' safety in medical applications. Therefore two states of *normal* and *emergency* have been considered for the proposed TDMA-based MAC protocol. A reallocation of the super frame slots is introduced, which is only triggered if an abnormality has been processed by the coordinator in the data received from the physiological sensor nodes. Nodes of interest will be able to send data on a higher data rate or even by being assigned higher duty cycles for the next beacon frame or frames if they are prompted by their coordinator. An optional synchronization is also implemented for the protocol where a comparison of the nodes' local clock and the master node's local clock helps to decide whether a synchronization process has to be done in the next

TDMA frame or not. This way, the overhead due to unnecessary synchronizations will be eliminated automatically. The simulation model of 20 nodes seems, however, to be quite a large WBAN scale, as normally 20 sensor nodes deployed on the patient's body cause too much inconvenience. When they make the data abnormality happen at second 180, they assume the emergency state to last for almost four minutes, for which there is no justification in the paper. Great savings on latency are achieved when a node goes into the emergency state as it makes some nodes cease transmitting or entering their sleep periods at the same time to give way to the emergency data.

Osterlind et al. (2010) explained a technique where a low duty cycle can be applied to networks of a sudden traffic characteristic. For sudden changes in traffic, the offered duty cycle must change from low to high. It is to help transfer the buffered data, but Osterlind et al. (2010) propose a low duty cycle technique that can also handle the peaks in the traffic through a contention resolution mechanism. The topology described in the paper is a star one hop topology. Nodes must send requests of random lengths to the coordinator (or "receiver" as stated in the paper) upon having data to send in a contention round. The coordinator will receive the request messages and check for the lengths and will make a decision on granting the channel access to the node with the highest request length. A collision resolution mechanism was introduced in the case that two or more nodes send out requests of the same size, which would let them send again in the next contending round. A failed channel access in such a case is depicted in the acknowledgement message that is sent out to all nodes at the end of each finished transmission. This is a distributed contention mechanism based on drawing straws (requests) that would let several nodes simultaneously send their request upon having data to send. Unlike the traditional RTS/CTS mechanism, the contention here is not initiated by the sender but by the receiver or the coordinator device. The average contention rounds have much shorter lengths than those of the binary exponential backoff mechanism in the standard which, with a protocol on a relatively low duty cycle, will perform better in terms of collision probability as the nodes manage to finish their transmissions faster. There are, however, some inefficiencies with their proposed technique, such as no priority guaranteed approach, which lets the nodes just pick a straw of any length and be selected randomly by the coordinator device, which does not present a good level of fairness and might endanger reliability at times.

3.2.3 **MAC Layer Queue**

In Ghaboosi et al. (2011), the best route from the sensor to its coordinator (or to the coordinator of a coexisting WBAN in its neighbourhood) is selected based on a stochastic routing mechanism. Some key parameters of the network such as maximum queue utilization factor were considered. The connectivity of the sensor nodes in one WBAN is not limited to their local sink and they can send their data via establishing routes to the sink nodes of the other co-existing WBANs. The route by which the data is relayed to a sink must be chosen based on three parameters: outage probability, instant queue utilization factor, and the remaining battery of the sensors that will be sent to the hub of their local WBAN and shared among the neighbouring WBANs. These parameters will be used to determine a route for a sensor upon its sent request and will be shared among all the hubs of all the neighbouring WBANs at the same time as they share their routing tables with each other. The proposed algorithm performs a route categorization algorithm based on these parameters that uses a counting-down mechanism of a few parallel counters, each corresponding to a certain route. The first one to reach zero is the chosen route and the rest of the counters will stop immediately after a route has been chosen. The random placement of the sensors in the WBAN scenario is unusual in their paper, as normally the WBAN deployments cannot be based on a random scattered structure. In Zhuo et al. (2012), the duty cycle of the nodes in a hybrid CSMA/TDMA structured MAC is dynamically adjusted based on the information of its MAC queue length, which is described as a traffic adaptive metric in their implementation. The problems associated with CSMA poor performance in high traffic loads and also the scalability and overhead problems associated with pure TDMA approaches motivated this research. A fixed duty cycle is also described as a barrier to an efficient MAC performance in times of bursty traffic. The authors propose a dynamic duty cycled MAC protocol where the load of each node is reported to its cluster head so that the heavily loaded nodes (based on their MAC queue length) will be granted more TDMA slots. The duty cycling is achieved according to the traffic conditions of the network (both low and high) to resolve the conflict that exists between bursty traffic and low bandwidth (low duty cycle). To the Payload of the MAC frame is added a queue indicator that reflects the load imposed to each node. These queue indicators of the received packets, when piggybacked, are used to achieve a variable TDMA period performance in the super frame structure. The sensor nodes with a higher load will have more chances of conquering the medium to send their data packets. The paper by Hok Limand Qiu (2001) is another example of dynamicity using queue length information.This is discussed more in Section 3.4.

3.3 Exploiting PHY Layer Parameters

3.3.1 Changes in RSSI Levels

Link quality indicator (LQI) and energy detection (ED) are the two main parameters that indicate information about the wireless channel (Papadimitratos, Mishra, & Rosenburgh, 2005). Network load and the quality of the wireless channel are believed to play an important role in the MAC sub-layer performance. The paper encourages considering the wireless channel errors in a MAC protocol cross-layer design. Two different identifiable data flows in the network were given a different amount of access to the wireless channel. The two data flows differed from each other in terms of how well they could have claimed an access to the channel or how error prone their access had been so far. The node's wireless link quality was used as a measure in determining the backoff time to assign to it in the CSMA/CA procedure. The wireless link quality of a node is defined through its error performance, and shorter backoff periods will be assigned to a node with a lower error rate in its wireless link. However, this will be an aggressive approach and not a balanced one in terms of channel utilization and fairness among nodes since some nodes with poor link quality may undergo long delay time and never be granted access to the channel.

Prabh and Haur (2011) stated that body movement makes fluctuations and dynamic changes in the RSSI that in fact affect any type of measurement in different scenarios including our MAC approach. Having performed the experiments on three different platforms, SHIMMER, Telos, and Micaz, the authors believe there is not much fluctuation in the signal between internal and external antennas in a WBAN. The idea in the paper is to transmit the packets during high RSSI windows. The RF interference is negligible in their work and this was verified by measuring the noise-floor periodically. They measure the RSSI for the closest node to the sender while the subject is walking and they draw the fluctuations and conclude that it fluctuates by around 20 dB, but this fluctuation is even more for a node on back because there will be more noise. The authors try to schedule the packets such that they are transmitted when the signal strength is high. Considering the fact that signal strength amplitudes (100s milliseconds) are long lasting enough compared to the airtime of packets (a few milliseconds), their proposed scheme would be feasible. When a node (sender) starts sending packets to all the other nodes, they passively listen and they do not send any ACK and they keep the statistics of the correctly received packets and the associated RSSI (they do this by the first eight symbols following the start of the frame delimiter). The RF interference is assumed to be negligible in the test area. Changes

in the body movement manifested in the RSSI as periodic fluctuations. They captured the RSSI levels for the sender for only 10 seconds in a graph and revealed that the RSSI has fluctuations of around 20 dB every second (ranging from -60dB to -80dB). So the goal is how to send the packets when the RSSI value is at its highest. One precondition needed is that there would be enough fluctuations in the RSSI signal. Therefore, they monitored it during the 25% and 75% of the readings during the five-minute experiment. They did the experiments with different platforms, and they also changed the data rate from one packet every five milliseconds to 20 packets every second. In their methodology, they state that if they want to exploit the fluctuations in the RSSI value they must be able to predict the RSSI pattern.

Ideally, the RSSI pattern must be periodic and have durable amplitudes so that they would be able to schedule the packets within the peak times of the RSSI. This way the packet loss will be reduced. A time interval that has a high RSSI is called the OTW. The problem was that an RSSI pattern could never really be periodic as human movements are often not periodic and so are unpredictable plus there was noise in the environment too. Therefore, in order to locate the OTW, a relation between the dominant peak in the RSSI pattern and the speed of the object is found.

Besides all the works described throughout Sections 3.1 to 3.3 that discuss the research done into the lower layers of the protocol stack such as MAC protocol, there are also some works such as Mercurry, a platform designed in Lorincz et al. (2009) that is an interesting practical example of a real wireless body area network implemented at Harvard University for patients suffering especially from neuromotor diseases. The work presents enhancements at the application layer of the protocol stack for two application categories: 1) patients suffering from Parkinson's disease and 2) patients suffering from epilepsy seizures. Unique techniques have been incorporated in the system architecture where effective trade-offs can be made according to different application requirements. Mercurry may not be the closest work to our MAC research in this thesis but integrates intuitive methods to alleviate the consumed energy through different controlling operations in downloading, data storing, and enabling/disabling certain features of a sensor node under specific circumstances. There is a narrow interface between the small modules running on each sensor and the application core driver that runs on the base station (assumed as a laptop). The application core driver is programmed to have full control over the activity level of the patients by running some sophisticated analysis of the data it receives. Unlike most of the other works in the literature where the experiments are done in

laboratory settings, Mercurry relaxes this assumption by testing its platform on real patients who are not necessarily within the range of the base station (laptop). The main challenges set to address in their work are:

- Saving battery lifetime
- Tunning the operation of the network in face of variations in bandwidth
- High quality data

The programming interface is flexible since the applications differ in data rate and requirements. For every application based on the current sensory data, certain actions take place such as disabling the gyro, or downloading feature data rather than raw data, or even disabling the data storage. A wise trade-off is the direct result of such behavior. The needed data to be passed to the application core driver to make a decision is sent through request packets (heartbeat packets) from sensor nodes.

3.4 **Fuzzy Logic Control for Dynamic Approaches**

Extensive research has been focused on using fuzzy logic in Connection Admission Control (CAC) applications. Hok Lim and Qiu (2001) proposed two efficient traffic control schemes by means of fuzzy logic in CAC. The first fuzzy logic system takes three traffic parameters (arrival rate, average length of queue, and average traffic load) as the inputs to come up with a weight factor that will be incorporated into calculating the estimated available bandwidth that can be offered. The second fuzzy logic system serves as a traffic predictor to produce a predicted traffic parameter (PCR). PCR will be compared to the first fuzzy logic system's output. The result of this comparison will contribute to the outcome of a CAC to be a reject or an accept. A distinctive high bandwidth utilization is reported to be the advantage of their method over the conventional traffic descriptors (*a priori* and measured non-predictive) and the authors declare the fuzzy logic predictor as an accurate and soft estimator of the network traffic.

Otal & Alonso (2009) presented a realistic medical setting that offers two solutions to a WBAN MAC protocol: 1) anovel cross-layer fuzzy-rule scheduling algorithm, and 2) energy-aware radio activation policies.In critical situations, i.e. when a packet's delay goes high or the residual energy in a sensor's node is too low, a node can demand a "collision free" time slot. Similarly, they may not send anything if the channel is in bad conditions. They borrowed the idea from a work originally done in 1993 (Lin & Campbell, 1993). The idea is to have a

distributed queuing random access protocol by dividing a time slot into two different parts: one for the user-request access and the other one for the collision-free access. Two types of queue were considered for each of these divisions. The idea is how each node in the network uses these queues to have access to the two divisions of a time slot in a distributed way, i.e. they find out the position of each node in these queues and the number of stations in each queue. The topology chosen is star.

The difference between their proposed method and the base Distributed Queing Random Access Protocol (DQRAP) is that, for the DTQ, the serving order is not FCFS manner but a "cross-layer fuzzy-rule based scheduler" does this. It works as an intelligent MAC protocol that adapts itself to the traffic load, channel link quality, and QoS requirements. Fuzzy logic algorithm can well model some dynamic and unpredictable parameters that affect a WBAN communication link: the variations in signal quality (due to interferences), the residual battery lifetime, the packet waiting time, etc. So they actually dictate the demand or refuse of the next frame collision-free time-slot scheduling. Since they are changing and uncertain, the use of fuzzy logic theory seems to be appropriate for such implementation (this is because we have a system that has inputs of a diverse nature). If a system exists with only its inputs as the varying-in-value parameters, the cost of its implementation would be much less than the scenario in which something has to be re-implemented for every different input. For the fuzzy system inputs, they pick three different variables that show the system behavior best (Signal-To-Noise Ratio, waiting time in the system, and residual battery lifetime), which they would convert or fuzzify by linguistic variables.

The increased network dynamic in Munir Saad, Bin, Biao, and Jian (2007) is one of the motivating factors to use a fuzzy logic system that has proved to best sort out the traffic and minimize the packet loss ratio. Desired QoS metrics are maintained by controlling the congestion at node level, which is implemented by a fuzzy logic based mechanism. It is stated in the paper that fuzzy logic provides a proactive approach towards achieving efficiency in a high dynamic scenario. The queuing model located in the node's buffer classifies the traffic into three different types: Source-event driven, source-continuous data, and sink-query based, with the event-driven type having the highest priority as seen in most of the WSN applications. The state of congestion is identified by following the current network state. The congestion is associated with buffer congestion, and different conditions of congestion have been assigned to such a state by means of a fuzzy system. To identify the buffer congestion level at any time through the fuzzy system, two main parameters of net packet inter arrival rate and current buffer

occupancy (percentage level of buffer fullness) were considered. Three levels of low, medium, and heavy congestion levels are attributed to the fuzzy set variables.

In Rahman, Kennedy, Simmonds, and Edwards (2008), precise representation of the network traffic is deemed to be time consuming and costly. It becomes especially impossible for high-speed broadband networks where the volume of data is high. Therefore, a traffic modeller is proposed by the aid of a fuzzy system to analyse the network traffic. Building a traffic modeller from the realistic measured data gives the advantage of no strict assumptions to make as the model gets built by the real data rather than the data being fed into a pre-existing model making it more adaptive. Fuzzy logic was chosen in their work to deal with data of an imprecise nature. In terms of network traffic with a given range for its load, a precise description of its data rate cannot be made. The term HIGH for example can be interpreted in different ways and thus makes fuzzy logic the most suitable for their system implementation. The complexity of the mathematical tasks required in an analytical model is declared to be reduced greatly through input variables' range reduction, which is what fuzzification does in a fuzzy system design. Different fuzzy sets defined for a fuzzy input variable range categorizes this input range (which could be a long one) into different categories; therefore it decreases the number of values to examine for each output value. The instantaneous fluctuations of traffic in the network with a varying load were captured by their traffic model.

The paper by Zuo, Xin Ng, and Hanzo (2010) is another example in the area of routing protocols where the imprecision and inaccuracy of routing information in making a decision for a particular route is mitigated by using a fuzzy based system that would give the possibility of feeding parameters of two different layers. It is shown how the overall system throughput is improved using a fuzzy based dynamic source routing (DSR) rather than a normal DSR. In situations with high mobility, the routing information stored in the routing table becomes old quickly and frequently due to high dynamics in the current stored routes such as route breakage.A fuzzy logic system is said to perfectly suit such dynamicity with a great deal of imprecise data and also the real-time routing situation's dependency on several parameters at the same time. A link's lifetime is predicted by a model that takes into account different information about each node's mobility, such as its velocity, direction, and position, to decide how "stable" a route is. This is attributed as a "stability" parameter as the output of the fuzzy system. This parameter gives certain weight values to each possible route for how "desirable" a route is at the moment to be used. But, as mentioned above, the highly volatile nature of the

route table needs a “timeout” period to be set for each desirable route, after which such route will no longer be perceived as stable.

Salcic (2014) used fuzzy logic to control traffic for asynchronous transfer mode (ATM) networks simulated in Matlab. An ATM network is defined as a network that does data transmission and switching of information in fixed length packets such as 53 bytes. There are two major functions created to avoid congestion in such a network: connection admission control (CAC) and usage parameter control (UPC). The first function relates to all the activities that will be done in order for a call request from the user to be accepted or rejected, and the second one has the responsibility of monitoring an established connection. Defining accurate traffic enforcement for UPC led Salcic to conclude that crisp logic evaluations are not suitable for characterizing the traffic model in ATM networks as they employ fixed thresholds. The fuzzy logic’s capability to formulize approximate reasoning motivated Salcic to characterize the policer (“policing” is the task of ensuring that each user complies with the traffic parameters negotiated) based on fuzzy logic. The degree of a source’s compliance with the agreed traffic parameters is evaluated through a fuzzy system and is no longer a matter of true or false only. Such a fuzzy policing model is built based on the requirements of an ideal policer described in details in the paper.

3.5 **Chapter Summary**

This chapter summarized some of the most related techniques in regard to dynamic approaches taken in a MAC protocol design. The categorization of the methods, which was based on the exploited parameters used in the reviewed works, gives a good insight into some of the most commonly used methods by researchers around the world. Sections 3.1 to 3.3 covered these main techniques through different parameters of interest and different target QoS metrics to achieve. Section 3.4 summarized some of the dynamic methods that have used fuzzy logic and discussed the advantages of using fuzzy logic as a tool to fulfill their design goals.

4. Chapter 4: Fuzzy-enabled MAC Overview

We saw in Chapter 3 how tuning different parameters of the CSMA/CA algorithm would affect the performance of a network in different terms. We also elaborated on the fact that the target QoS metrics to achieve must dictate and be the main motivation in choosing which key parameters (application/protocol-specific) to consider and tune. If fairness and optimum reliability among the sensor nodes are the main QoS goals to achieve, it must be made sure that the backoff mechanism of the CSMA/CA is not performing aggressively in terms of the produced backoff period lengths assigned to each node. An understanding of each node's history of past trials in having access to a channel is therefore necessary, and can be considered in the form of a parameter (called "$Channel_{Clear}Rate$" in our approach and described later) to be engaged in the node's next access trials' backoff mechanism. Therefore no node will be waiting too long nor will they be aggressively given access to the channel. The main flow of the idea behind our method is depicted in Figure 4-1.

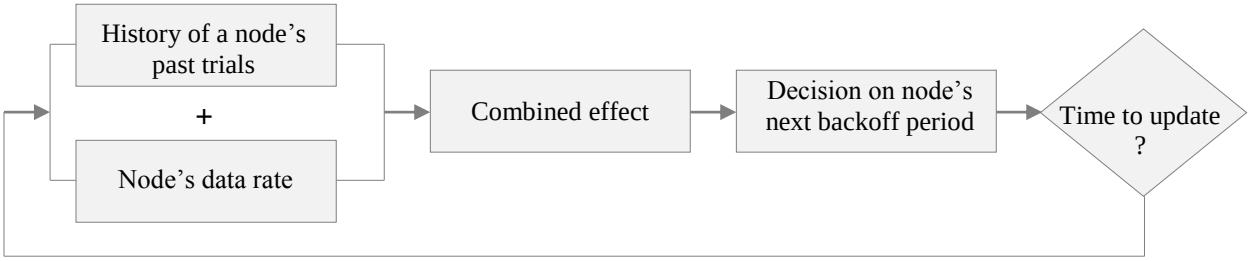


Figure 4-1: Method flow

The flow starts with two main inputs that will have a combined effect on deciding the length of the backoff period assigned to a node when it enters the CSMA/CA procedure. The combined effect of these two parameters was calculated by designing a fuzzy logic inference system that operates based on a set of rules to output a scalar (defuzzified) value. This scalar value is then used as a measure to tune the length of the backoff window from which the CSMA/CA procedure randomly chooses a backoff period for the node attempting a channel access. The tuned backoff window will sustain its length for a period of time before it updates it with new calculated values of $Channel_{Clear}Rate$ (which will be discussed shortly). This chapter will describe in detail the fuzzy inference system design along with a discussion of the input parameters.Two input parameters of each class of parameters described in Chapter 2 (application and protocol-specific)are chosen. As discussed in Chapter 2, we assume a periodic traffic for a low-data medical set up for our experiments that would best suit a chronic disease type such as diabetes or most of the aging-related illnesses such as general every day

monitoring of an elderly person. We assume 5–10 different physiological sensor nodes (only five nodes for our real experimental set up described in Chapter 6) with pre-defined but diverse and application-specific data rates. The dynamicity in data rates is, however, examined by varying the data rates of all nodes continuously as the simulation goes on, which puts the average of all data rates in an ascending fashion. This has been tested on one occasion to show the effect of a dynamic ascending data rate (later described in Chapter 5). The main goal of this research is to propose a highly reliable MAC protocol for low data-rate applications but we also test high data-rate sensor nodes such as ECG in the network, which has a relatively higher data rate than the other sensors in the network. For moderate periodic traffic behavior with reliability in mind as the main concern, a fair access to the shared wireless channel contributes greatly to the overall system's reliability performance. This fairness makes the channel access possible for all the heterogeneous sensor nodes in the network while examining their individual data rate value and also their successful channel access rate. The successful access rate for each sensor node is considered to be our main traffic indicating parameter, which is derived in the continuing loop of the CSMA/CA performed in the CAP by each node. Data rate as discussed in Chapter 2 and in Section 2.2is a static array, which has been tested for a varying behavior in Chapter 5, when the average data rate of all the nodes in the simulation is increased in each simulation run to examine the effect. The performance of the proposed scheme is benchmarked against that of the IEEE 802.15.4 MAC protocol, which has been widely used in medical applications throughout all the reviewed works in this thesis.

4.1 Why Fuzzy Logic?

Autoregressive models can be used as a means to predict the next values of a time-varying variable but may not act as efficiently as fuzzy logic concepts in many ways (Hok Lim& Qiu, 2001). Fuzzy systems are said to be knowledge based, which can provide a systematic procedure to translate human knowledge into non-linear mappings (Wang, 1997, p. 6). What differs fuzzy logic from the other existing methods in engineering is the ability of such systems to not only perform based on analytical measurements that rely only on physical laws but also to engage human expert knowledge that is normally described in normal language (Wang, 1997, p. 2). For these two different knowledge sources to combine, though, there should be a way to formulate the human knowledge into mathematical descriptions. Such transformation of human knowledge into mathematical formulas was made possible thanks to the introduction of fuzzy systems to the world in 1965 by Lotfi A. Zadeh (Zadeh, 1965). Although the application range for fuzzy logic is quite wide, most of the focus in current research has been

on the use of such systems into solving complex problems of control systems. Fuzzy logic can effectively deal with defining sets over input parameters that do not represent precise boundaries in their degrees of membership to a particular set, yet can be modelled at utmost precision. This took the world of mathematics and binary sets, which was hitherto unable to define sets with non-clear boundaries, to a whole new level.

As discussed later in Chapter 3 and Section 3.5 fuzzy logic has been used in diverse control problems in the area of wireless networks. Some of these use cases focus on incorporating a fuzzy decision system in Connection Admission Control (CAC) applications such as the one explained in Hok Lim and Qiu (2001). A traffic control scheme is discussed by the authors which is more detailed in Section 3.5. Otal & Alonso (2009) is another example of using a fuzzy system to present a realistic medical setting that offers solutions to a WBAN MAC protocol. The increased network dynamic in Munir Saad, Bin, Biao, and Jian (2007) is also one of the motivating factors to use a fuzzy logic system that has proved to best sort out the traffic and minimize the packet loss ratio. Controlling the congestion at node level is achieved by a fuzzy logic based mechanism. It is stated in the paper that fuzzy logic provides a proactive approach towards achieving efficiency in a high dynamic scenario. In Rahman, Kennedy, Simmonds, and Edwards (2008), precise representation of the network traffic is deemed to be time consuming and costly and a traffic modeller is proposed by the aid of a fuzzy system to analyse the network traffic. The paper by Zuo, Xin Ng, and Hanzo (2010) is another example in the area of routing protocols where the imprecision and inaccuracy of routing information in making a decision for a particular route is mitigated by using a fuzzy based system that would give the possibility of feeding parameters of two different layers. Salcic (2014) used fuzzy logic to control traffic for asynchronous transfer mode (ATM) networks simulated in Matlab. All the mentioned works here have been illuminated more in Section 3.4.

The main focus of this chapter is to explain a fuzzy system that can decide about a node's waiting (backoff) time based on two of its characteristics, namely data rate and $Channel_{Clear}Rate$, which is a history of a node's successful channel access trials. Therefore the designed architecture presented here is generic and could be used and tested with other parameters of interest, which indicates the flexibility of our proposed method for future researchers working in different standards' confines such as IEEE 802.15.6. The fuzzy system described in this chapter is a normal fuzzifier-defuzzifier fuzzy system with no dynamicity

introduced, which means there will be no feedback of the output to affect the inputs. We described earlier in Chapter 2 that specific requirements of an application are the actual dictators of how a MAC protocol should act such that it will always yield to a trade-off between the performance metrics of interest and the ones with less priorities. The fuzzy logic described here makes it possible for different strategies to be taken on the go as nodes are performing their transmissions. Different patterns of waiting times happen for individual nodes as they traverse the super frames with their successful and failed transmissions to the end of the testing time period. This happens regardless of their priorities of the application type they perform. But this adds no concern to the general design of the protocol as additional inputs such as a "priority" can be added to the fuzzy engine with no extra cost since priority for a physiological sensor node can be always predefined with a constant value. An ECG sensor node, for instance, can always be assigned a higher priority compared to a temperature or a respiratory sensor. Adding priority in a low data rate MAC design, however, would only add to the complexity of the implemented fuzzy system in terms of number of inputs and the rule. The taken approach does take advantage of both contention-based and contention-free periods during a beacon interval. As explained in previous chapters, we conducted our research on the beacon-enabled mode of access in IEEE 802.15.4 MAC where access techniques of both CSMA and TDMA are incorporated into the super frame structure. A discussion of different MAC techniques and the advantages and disadvantages of each in Chapters1 and 2 has motivated the use of a hybrid MAC protocol for our method.

4.2 Fuzzy Logic Traffic Descriptor (Inputs and the Output)

4.2.1 $Channel_{Clear}Rate$ Averaged over n Super Frames

We use fuzzy logic to interpret our input (traffic indicating) parameters since it gives us the possibility to make our MAC protocol aware of both an application-specific parameter and a protocol-specific one. In a pure design of the standard, of course no exploration of the application-specific (yet effective) parameters on the backoff behavior has been taken into account. In other words, we are in need of a common thread that would be able to connect different causes of an inefficient or an identical backoff together for a decision to be made about the next backoff value at the beginning of every transmission. The designed fuzzy algorithm is implemented on the sensor's side as all the information we needed to calculate the output value resides on the sensor's side and not the coordinator. We will see later how the complexity of the algorithm has been alleviated to least affect the battery lifetime of the sensor

devices. Other research work such as BSN-MAC (Li & Tan, 2005), a MAC protocol designed for body sensor networks, have also shifted the computations of key parameters to the sensor's side as the required information for such data was associated with the sensors. BSN-MAC, as earlier reviewed in Section 3.1.3, lets the entire calculations of their defined priority parameter be carried out by the sensors. This priority parameter is then utilized to set the length of the beacon interval for the node that has requested it. This method can be effective if the greatly reduced overhead, associated with the control packets, comes at the cost of only some affordable computational tasks. Similar to their method, our fuzzy algorithm, as explained later in Chapter 5, has been simplified by a history-based technique to prove efficient on real sensor platforms.

The chosen inputs of our fuzzy decision maker are data rate of each sensor node and the $Channel_{Clear}Rate$ described later in this section. A transmission attempt during the CSMA/CA procedure ends either by a failure or a success and, in either case, the backoff times will be recalculated with the current *BE* value in each super frame according to the original standard's algorithm. With our approach, the value of the backoff time is updated based on the traffic in the last "*n*" super frames. $Channel_{Clear}Rate$ is an average parameter made over every *n* super frames paved from the beginning of the simulation up to the current time of the simulation, so it only considers the last *n*super frames for making its averages and updating the previous one. An alternate way of calculating this parameter is explained in Section 4.2.2, where the average value is not made over *n* super frames but over the entire trials (successful or failed) during *n*super frames. In other words, for making the average value in the second method ($Channel_{Clear}Rate$), the number of second successful CCAs is divided by the entire number of failed and successful CCAs during *n* super frames. We formulated the second method of calculating $Channel_{Clear}Rate$ as we reached the real experimental phase of our research when we aimed to implement the fuzzy-enabled MAC on real SHIMMER sensor platforms' TinyOS operating system, as described in detail in Section 4.2.2.

In the first method for obtaining the $Channel_{Clear}Rate$ parameter, every transmission attempt inside the CAP is examined for the number of its second successful CCAs, which is indicated when the CW reaches zero. Once a second CCA (denoted as CCA_2) is assessed as successful, it will increment a counter that keeps account of the number of CCA_2s. $Channel_{Clear}Rate$ is calculated by dividing the number of all experienced CCA_2s to the interval of time (*n* super

frames) they were monitored over. Therefore in each simulation run, the $Channel_{Clear}Rate$ in the second CCA (i.e. a CW equal to zero), will be calculated based on the formula below:

$$Channel_{Clear}Rate = \frac{\sum_{i=0}^{i=n} CCA_2}{n}\ i \tag{3}$$

Notably, the second CCA (denoted as CCA_2) is selected in (3). The reason is because we only care about the clear channel assessment that has led to a successful transmission, which in the standard (with a CW=2) will be the second consecutive CCA. There might be a busy reported channel right after the first channel is clear, thus making it a not-reliable evidence of a clear channel. We have calculated the $Channel_{Clear}Rate$ after every *n* super frames to produce the next backoff time for each node. The number of CCA_2 experienced for every super frame will be denoted by an index (i) showing the specific super frame; for example CCA_{21} shows the number of second CCAs of the first super frame. The total number of all the second CCAs up to the n^{th} super frame will be divided by the number of super frames considered as the update interval, which is denoted by *n*. For the first 20 super frames to pass, in case of *n=20*, the value of $Channel_{Clear}Rate$ is zero. After *n* super frames, the node gets its assigned value of $Channel_{Clear}Rate$ which is absolutely calculated based on its past trials during the last *n* super frames. Using this parameter does not incur any communication overhead as it is an already available parameter that keeps happening in the course of the communication and could simply be assessed as a success or a failure. The defined $Channel_{Clear}Rate$ parameter only keeps account of CCA_2's history of success outcomes during *n* super frames. Together with node's sample rate, the fuzzy system is able to locate a *FuzzyMaxDelay* within a range of (8, 127) symbols as the upper limit (bound) for the generated backoff time. This backoff time in the original CSMA/CA MAC algorithm is a randomly assigned value chosen from the interval of $(0, 2^{BE}-1)$ in which the BE is always initialized to *macMinBE* as its default minimum according to Figure 4-2(a). If the range of variations for BE is assumed to be 3–7 to denote its minimum and maximum values respectively, then the value of the real time generated delay (backoff time) will be always a value between eight and 128 symbols. Therefore, *FuzzyMaxDelay* can never be more than 128 symbols. The main contribution of our fuzzy system in the repeating loop of the CSMA/CA is the change in the produced backoff time every time two consecutive idle CCAs cannot be assessed and a backoff time has to be generated.We use fuzzy logic in Figure 4-2(b) to produce a maximum bond or limit for the variation of the randomly produced backoff time. We call this maximum limit *FuzzyMaxDelay*, and it decides the upper limit for

the window from which the backoff time is randomly chosen. In (4) the IEEE 802.15.4 behavior is depicted for random backoff time generation and (5) shows the difference our algorithm has made.

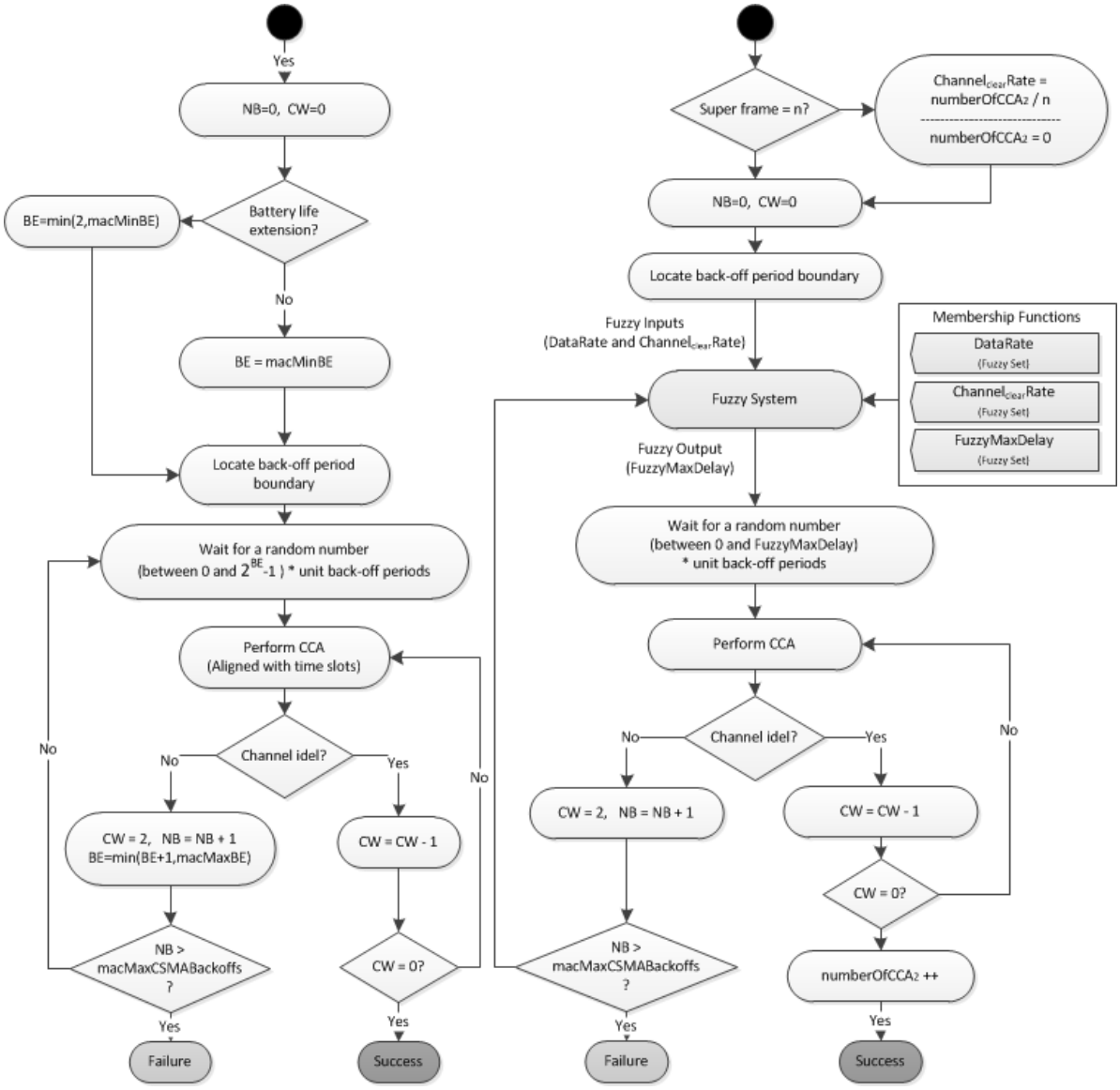


Figure 4-2(a): Original IEEE 802.15.4 CSMA/CA **Figure 4-2(b): Fuzzy-enabled CSMA/CA**

$$backoff\ time = rnd(0, 2^{BE} - 1) * unitBackoffPeriod * symbolLen \quad (4)$$

$$backoff\ time = rnd(0, FuzzyMaxDelay) * unitBackoffPeriod * symbolLen \quad (5)$$

The pseudo-code of the proposed MAC is noted here to highlight the changes we made to the IEEE 802.15.4 MAC mechanism. The randomly assigned backoff time from a window of (0,

FuzzyMaxDelay) is denoted as "BackoffDelayTime" in the code. The fuzzy sets for each of the inputs and the output are explained in next section.

Fuzzy-enabled Mac pseudo-code (with first method of $Channel_{ClearRate}$ calculations)

```
void Mac802154b::startup() {
numberOfReceivedBeacon = 0;
  numberOfChannelIsCLEAR = 0;
  channelClearRate = 0;
  ...
  //Initializing all parameters
  //Set timer for starting the frame
setTimer(FRAME_START, 0);
}

void Mac802154b::initiateCSMACA() {
  // initiate CSMA-CA algorithm
setMacState(MAC_STATE_CSMA_CA);
  NB = 0;
  CW = enableSlottedCSMA ? 2 : 1;
continueCSMACA();
}

void Mac802154b::timerFiredCallback(int index) {
  switch  index {
    // Starting the frame
case FRAME_START
if  isPANCoordinator {
        // as a PAN coordinator, create and broadcast beacon packet
      } else {
        // if not a PAN coordinator, then wait for beacon
toRadioLayer(createRadioCommand(SET_STATE, RX));
setMacState(MAC_STATE_WAIT_FOR_BEACON);
setTimer(BEACON_TIMEOUT, guardTime * 3);
        // beacon timeout fired
        //indicates that beacon was missed by this node
      }
case BEACON_TIMEOUT
      lostBeacons++;
      //Perform all functionality needed for handling a lost beacon
      ...
case PERFORM_CCA
      // preform carrier sense
      CCA_result CCAcode = radioModule->isChannelClear();
if CCAcode == CLEAR {
        CW--;
if CW != 0 {
          // since carrier is clear,
          // no need to generate another random delay
      setTimer(PERFORM_CCA, unitBackoffPeriod * symbolLen);         }
else {
          numberOfChannelIsCLEAR++;
transmitNextPacket();
        }

      } elseif CCAcode == BUSY {
```

```
        CW = enableSlottedCSMA ? 2 : 1;
        NB++;
      }
    }
}

void Mac802154b::fromRadioLayer(cPacket * pkt, rssi, lqi) {
  //Recive and casting the packet
  Mac802154bPacket *rcvPacket=dynamic_cast<Mac802154bPacket*>(pkt);
switch rcvPacket->getMac802154bPacketType()
case MAC_802154_BEACON_PACKET
numberOfBeaconReceived++;
if numberOfReceivedBeacon == 20 {
         channelClearRate = numberOfChannelIsCLEAR / 20;
         numberOfReceivedBeacon = 0;
         numberOfChannelIsCLEAR = 0;
      }
case MAC_802154_ASSOCIATE_PACKET
      // Performe request to associate functionality
case MAC_802154_GTS_REQUEST_PACKET
      // Performe GTS request packet
case MAC_802154_ACK_PACKET
      // Performe ack frames
case MAC_802154_DATA_PACKET
      // Performe Data packet recive functionality
}

void FuzzyMac802154::continueCSMACA() {
  previousChannelClearRate = channelClearRate
  Fuzzy Inputs, Output and Rule Definition;
  Fuzzy Inputs Initialization;
  fuzzyBackoffDelay = Fuzzy output.defuzzify();
  rand = random(0, fuzzyBackoffDelay)
  simtime_t CCAtime = rand * (unitBackoffPeriod * symbolLen);
  // Perform CCA after CCAtime delay
setTimer(PERFORM_CCA, CCAtime);
}
```

4.2.2 *Channel$_{Clear}$Rate* Averaged over *all* Trials in *n* Super Frames

Calculating the *Channel$_{Clear}$Rate* values according to (3) in Section 4.2.1 would not encounter any specific problem in terms of implementation in a simulation environment. However, carrying out the same concept in TinyOS for the real SHIMMER sensor platforms when nodes would transmit at relatively higher data rates (NHIGH and HIGH, compared to the data rates assigned in simulation environment) would result in a big range of variations for *Channel$_{Clear}$Rate*. In the real world, a node would experience over 1,000 CCA_2s during a 20 super-frame time interval (we investigated this by monitoring the number of CCA_2s in real experimental set ups). Therefore, using the method of calculating *Channel$_{Clear}$Rate* in (3) would result in values much greater than 100. The immediate problem caused by this is the relatively

big produced range for *Channel*$_{Clear}$*Rate* as one of the inputs of our fuzzy system, which would degrade the sensor's performance considerably. This will add more complexity to the heart of our proposed method when we explain an additional caching method in Chapter 5, which proved as a necessary simplifying method for preserving energy on real SHIMMER sensor platforms. When the range of an input becomes large it is not possible to cache all the previously calculated matches (described more in Chapter 5) on the limited memory of a sensor node. Therefore our history-based simplifying method described in Section 5.2.2.2 became a feasible option no longer, and the sensors had to perform the fuzzy calculations for every iteration. To address this problem we sought another possible way of calculating *Channel*$_{Clear}$*Rate* that would rely on the same concept of exploiting the number of CCA_2s in *n* super frames yet with another method of averaging it. Such an averaging method would lead to a constant interval for the variations of *Channel*$_{Clear}$*Rate* as indicated in (6).

$$Channel_{clear}\ Rate = \frac{\sum_{i=1}^{i=n} CCA_2 i}{\sum all\ CCA\ trials\ during\ n\ super\ frames} \quad (6)$$

Where *i* indicates the index of each super frame and CCA2*i* indicates the number of CCA_2s in that super frame. The sum of all the CCA_2s experienced during *n*super frames is then divided by the total number of CCAs performed by the node when attempting to have access to the channel. The *Channel*$_{Clear}$*Rate* range in this method will be a float number between 0 and 1 and will be explained in more details in Chapter 5. The pseudo-code of the new method for nesC language of the TinyOS is brought here followed by its flow chart (Figure 4-3) integrated in the CSMA/CA mechanism, which is an updated version of Figure 4-2(b).

Fuzzy-enabled Mac pseudo-code (alternate method of *Channel*$_{Clear}$*Rate* calculations)

```
void Mac802154b::startup() {
numberOfReceivedBeacon = 0;
  numberOfChannelIsCLEAR = 0;
  numberOfChannelIsNotCLEAR = 0;
  channelClearRate = 0;
  ...
  //Initializing all parameters
  //Set timer for starting the frame
setTimer(FRAME_START, 0);
}

void Mac802154b::initiateCSMACA() {
  // initiate CSMA-CA algorithm
setMacState(MAC_STATE_CSMA_CA);
  NB = 0;
```

```
  CW = enableSlottedCSMA ? 2 : 1;
continueCSMACA();
}

void Mac802154b::timerFiredCallback(int index) {
  switch  index {
    // Starting the frame
case FRAME_START
if  isPANCoordinator {
        // as a PAN coordinator, create and broadcast beacon packet
      } else {
        // if not a PAN coordinator, then wait for beacon
toRadioLayer(createRadioCommand(SET_STATE, RX));
setMacState(MAC_STATE_WAIT_FOR_BEACON);
setTimer(BEACON_TIMEOUT, guardTime * 3);
        // beacon timeout fired
        //indicates that beacon was missed by this node
      }
case BEACON_TIMEOUT
      lostBeacons++;
      //Perform all functionality needed for handling a lost beacon
      ...
case PERFORM_CCA
      // preform carrier sense
      CCA_result CCAcode = radioModule->isChannelClear();
if CCAcode == CLEAR {
        CW--;
if CW != 0 {
          // since carrier is clear,
          // no need to generate another random delay
      setTimer(PERFORM_CCA, unitBackoffPeriod * symbolLen);         }
else {
numberOfChannelIsCLEAR++;
transmitNextPacket();
        }


      } elseif CCAcode == BUSY {
        CW = enableSlottedCSMA ? 2 : 1;
        NB++;
numberOfChannelIsNotCLEAR++;

      }
    }
}
void Mac802154b::fromRadioLayer(cPacket * pkt, rssi, lqi) {
  //Recive and casting the packet
  Mac802154bPacket *rcvPacket=dynamic_cast<Mac802154bPacket*>(pkt);
switch rcvPacket->getMac802154bPacketType()
case MAC_802154_BEACON_PACKET
numberOfBeaconReceived++;
if numberOfReceivedBeacon == 20 {
sum = numberOfChannelIsNotCLEAR + numberOfChannelIsCLEAR;
         channelClearRate = numberOfChannelIsCLEAR / sum;
         numberOfReceivedBeacon = 0;
         numberOfChannelIsCLEAR = 0;
      }
case MAC_802154_ASSOCIATE_PACKET
      // Performe request to associate functionality
case MAC_802154_GTS_REQUEST_PACKET
      // Performe GTS request packet
```

```
case MAC_802154_ACK_PACKET
      // Performe ack frames
case MAC_802154_DATA_PACKET
      // Performe Data packet recive functionality
}
void FuzzyMac802154::continueCSMACA() {
  previousChannelClearRate = channelClearRate
  Fuzzy Inputs, Output and Rule Definition;
  Fuzzy Inputs Initialization;
  fuzzyBackoffDelay = Fuzzy output.defuzzify();
  rand = random(0, fuzzyBackoffDelay)
  simtime_t CCAtime = rand * (unitBackoffPeriod * symbolLen);
  // Perform CCA after CCAtime delay
setTimer(PERFORM_CCA, CCAtime);
}
```

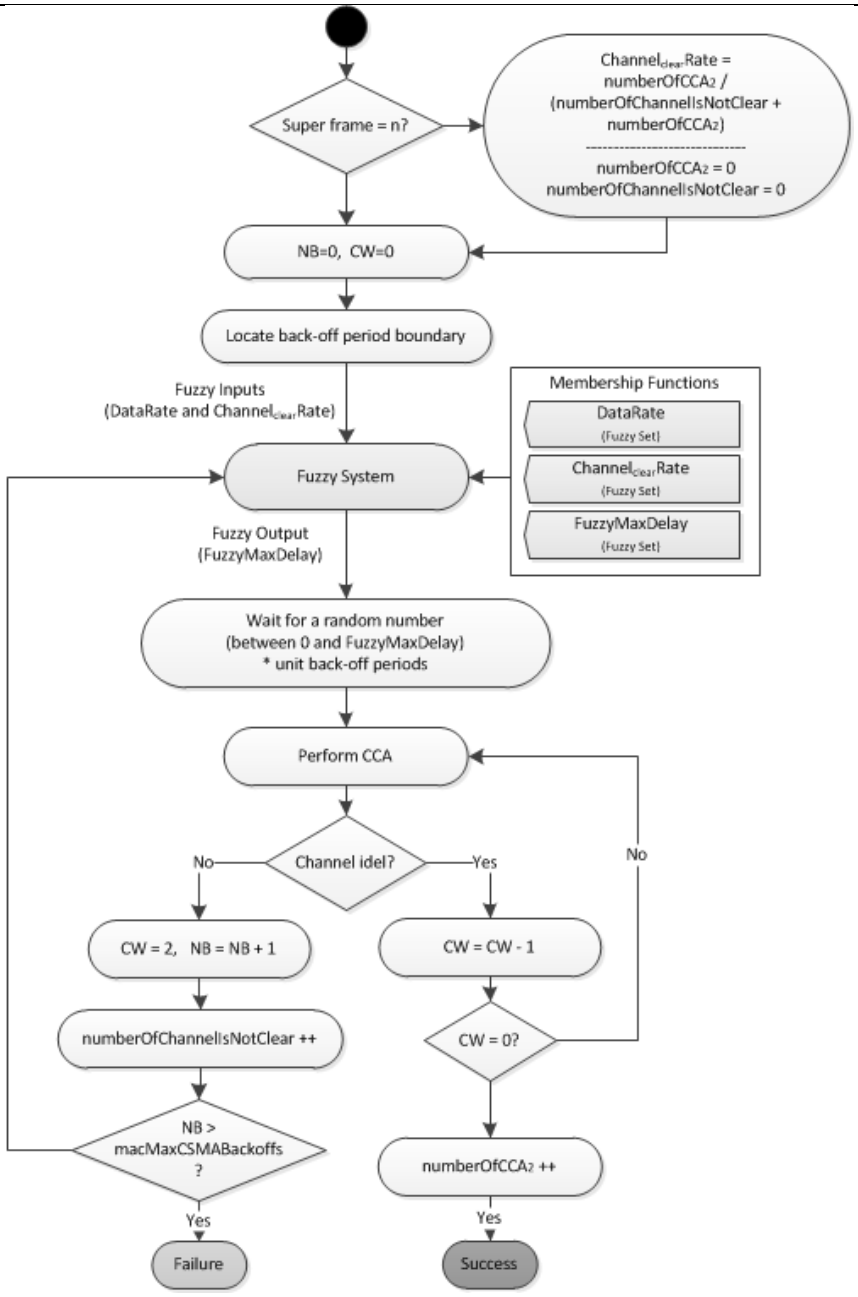


Figure 4-3: Fuzzy-enabled CSMA/CA detailed mechanism

4.3 *min* and *max* Values of $Channel_{clear}$ *Rate*

Determining the *min* and *max* values of the inputs of a fuzzy system is said to be the expert's job, and identifies the range of values that reflects the variable's real ups and downs. In the world of wireless communications, however, we witness a huge range of contributors to any visible behavior or any parameter's value in any designed network. Be it a received signal strength (RSS) value, packet loss, or delivery ratio of the system, experiments should be done to observe real fluctuations of the target parameter even if previous experiences suggest close-to-reality expectations. Figure 4-4 shows a simple overview of our fuzzy system discussed in the previous section.

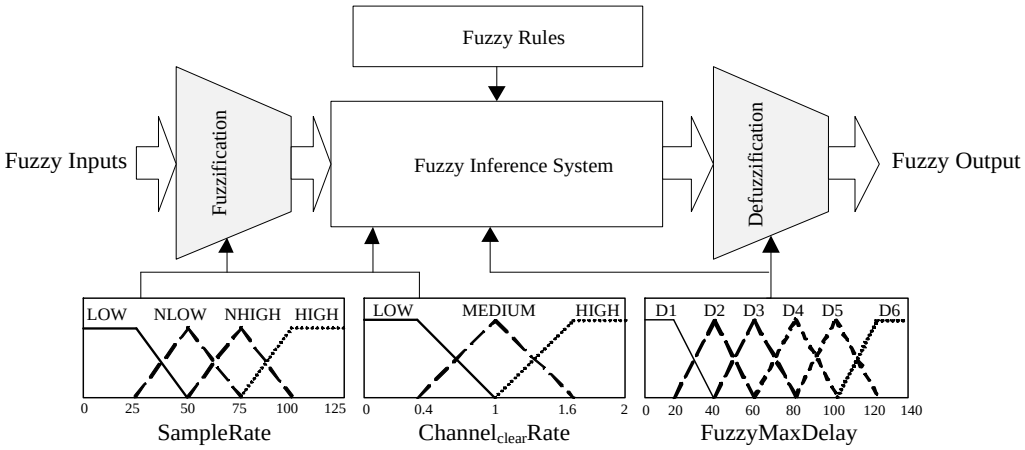


Figure 4-4: Overview of fuzzy system

Data rate and $Channel_{Clear}Rate$ are fuzzified into the linguistic variables we have defined for each of them. The defined ranges for both inputs are shown in Figure 4-4. Data rate range reflects the *min* and *max* of the data rate array assigned to nodes in our simulations later discussed in Chapter 5. One easy but not so accurate way of finding out the channel clear assessment rate in the second CCA was to make an average of all the CCA_2 made up until the very super frame that we were in at any given instant of time. However, this way of $Channel_{Clear}Rate$ calculation led us to a very smooth diagram that was not showing much of the current conditions as it was history-based and kept account of all the CCA_2numbers until this very moment. Hence it failed to show the real time fluctuations in smaller intervals. Therefore we decided to break down the number of passed super frames till the current time and do the calculations of the $Channel_{Clear}Rate$ with steps of "*n*" where *n* could be wisely chosen based on some testing. The explanations discussed here are in regard to the first method of $Channel_{Clear}Rate$ calculations as described in (3). The value of "*n*" has been investigated

with different values of 10, 20, 40, 80, and 100 steps (i.e. every 10, 20,…, 100 super frames). This is only because we wanted to examine the fluctuations in the number of CCA_2 when we considered different numbers of super frames at a time (different steps). A value of 20 for *n*, for instance, means the $Channel_{Clear}Rate$ of every 20 super frames has been taken into account for its average value to be made. It is obvious that the larger the *n* is, the smoother the fluctuations in the average number of CCA_2 will be and this is what we try to avoid in determining the min and max values for the $Channel_{Clear}Rate$ as an average value to cover a larger range of min and max for this variable when other parameters in the network change. Nevertheless we have tried to show the results of $Channel_{Clear}Rate$ values for different values of *n*. Therefore we ran the simulations 10 times for each *n* to show the min and max fluctuations of our first fuzzy logic input, $Channel_{Clear}Rate$. The simulations at this stage to determine the min and max values of the $Channel_{Clear}Rate$ were done for five nodes in a WBAN, which scenario is explained in detail in Chapter 5. Due to the large number of figures for observations of the $Channel_{Clear}Rate$ parameter values, we decided to include all the 50 diagrams in APPENDIX I, but we will show the averaged values over different steps in five diagrams in this section. Figure 4-5 shows in just one diagram the average for the 10 simulation runs only for n=10, which means the value of $Channel_{Clear}Rate$ has been determined according to (3) every 10 super frames. The individual diagrams in APPENDIX I suggest that the range of variations for this parameter is indeed between 0 and 1.6, even if steps of 80 and 100 are chosen. The average diagrams represented here suggest the same, and declares that any of these values (10, 20, 40, 80, or 100) could have been chosen in our calculations of $Channel_{Clear}Rate$. We chose *n* to be 20 for our own implementations of this proposed MAC algorithm but it can be configured to a different step. The data rates assigned to the nodes for these observations were set as 10, 20, 20, 100, and 60 kbps, assigned to Node1–Node5 respectively. The topology was a simple star topology with a coordinator the same as the topology set in the example BANtest scenario as one of the examples of Castalia simulator. Figure 4-6 shows in just one diagram the average for the 10 simulation runs only for n=20, for the five different nodes. This average was made over the 10 individual simulation results of $Channel_{Clear}Rate$ in intervals of 20 super frames throughout the 500 seconds of simulation. Although it shows the most possible values of $Channel_{Clear}Rate$, it does not clearly show the *min* and *max* values that occurred in the individual graphs we tested (as shown in APPENDIX I), as this is an average diagram of all the 10 individual simulation diagrams. The decided *min* and *max* values for $Channel_{Clear}Rate$ are therefore 0 and 1.6, which is concluded based on the fluctuations of this variable in all the diagrams achieved in APPENDIX I.

We have tried to ensure, by running the simulations for at least 10 times, that the *min* and *max* fluctuations of $Channel_{Clear}Rate$ stays no lower than 0 and no higher than 1.6. Since the variations in the value of our parameter reached an steady state after 10 simulation runs for each value of *n* it was absolutely clear that the fluctuations of the values of $Channel_{Clear}Rate$ are always going to be between 0 and 1.6 no matter how many times we test it.

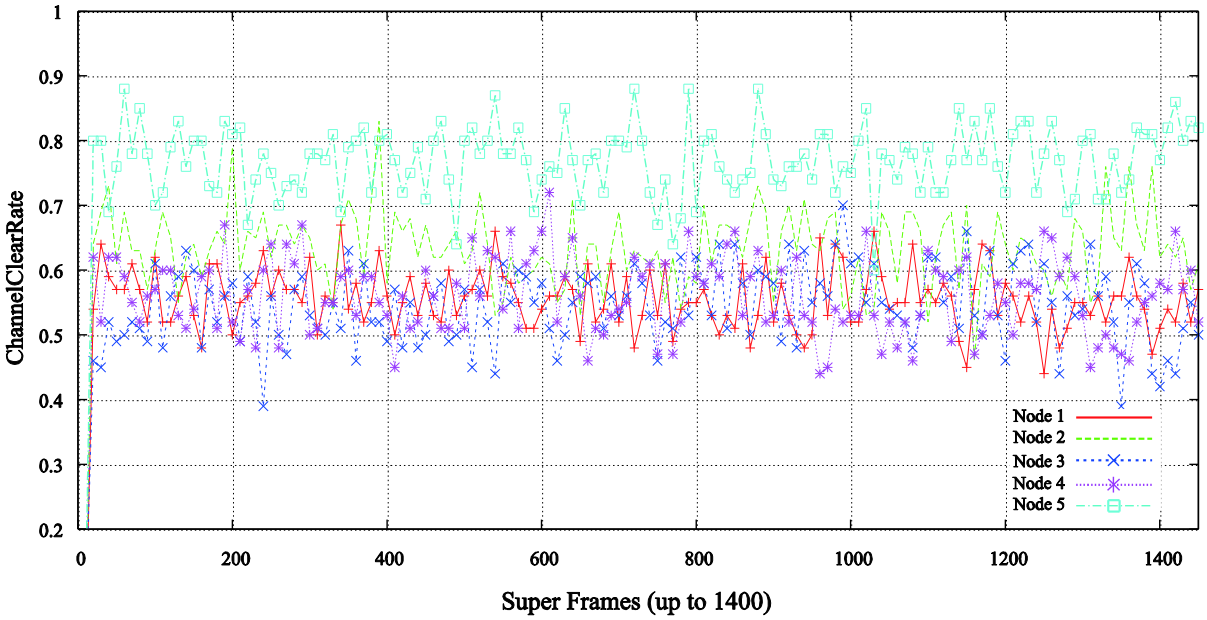


Figure 4-5: CCA_2 averaged variations for five nodes every 10 super frames (*n*= 10) during 500 seconds simulation run

This is also shown in Figure 4-3 for $Channel_{Clear}Rate$. As shown in Figure 4-5, X axis represents the number of super frames ranging from 0 (which is the beginning of the simulation) up to 1400 (as the very last super frame), with steps of *n* (for every *n* super frame), while Y axis is the percentage of $Channel_{Clear}Rate$ of the five existing nodes in our typical body area network.The strokes in this diagram (represented by different colors) are denoted by the node's ID.

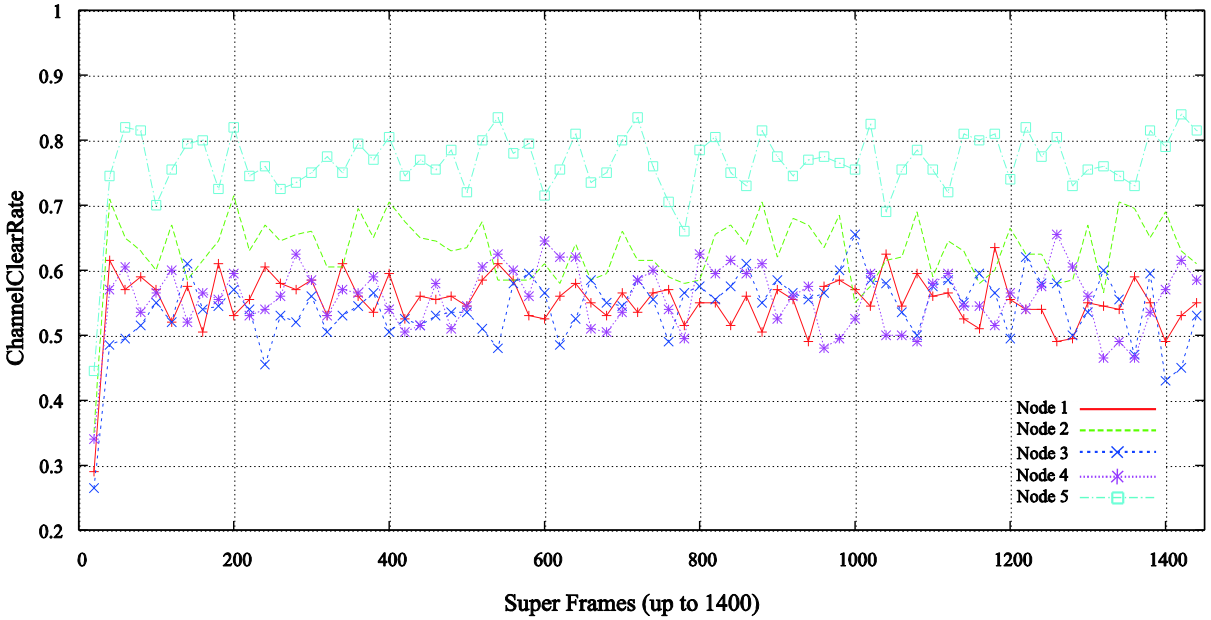


Figure 4-6: CCA_2 averaged variations for five nodes every 20 super frames (n= 20) during 500 seconds simulation run

Figure 4-7 shows in just one diagram the average for the 10 simulation runs only for n=40.

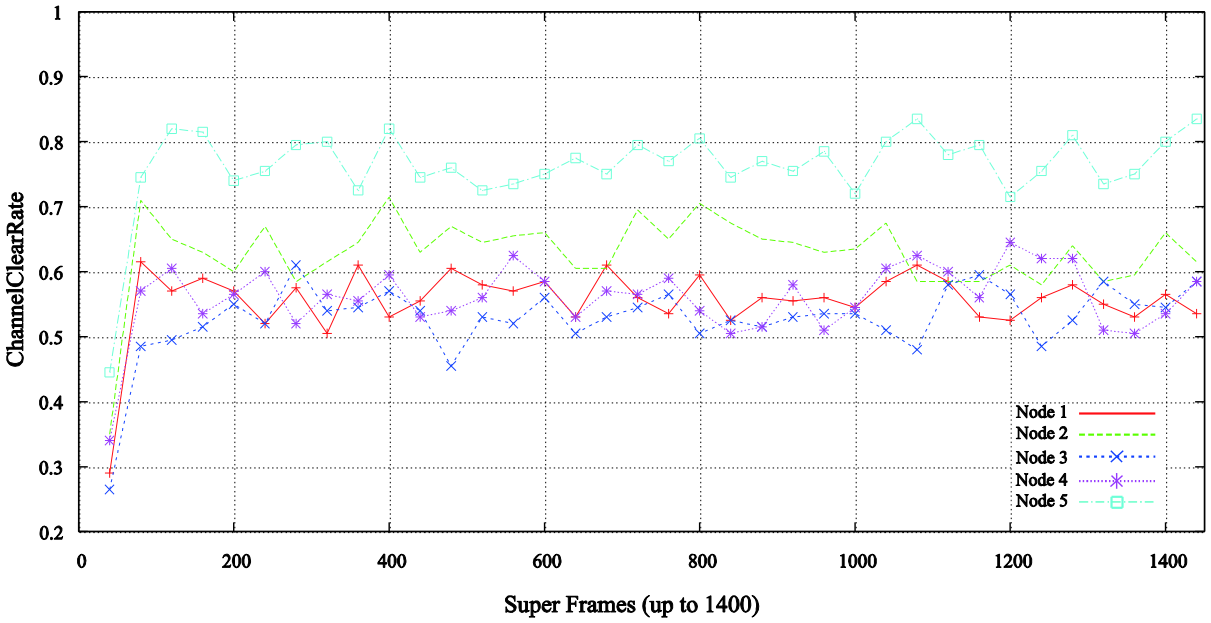


Figure 4-7: CCA_2 averaged variations for five nodes every 40 super frames (n= 40) during 500 seconds simulation run

Figure 4-8 shows in just one diagram the average for the 10 simulation runs only for n=80.

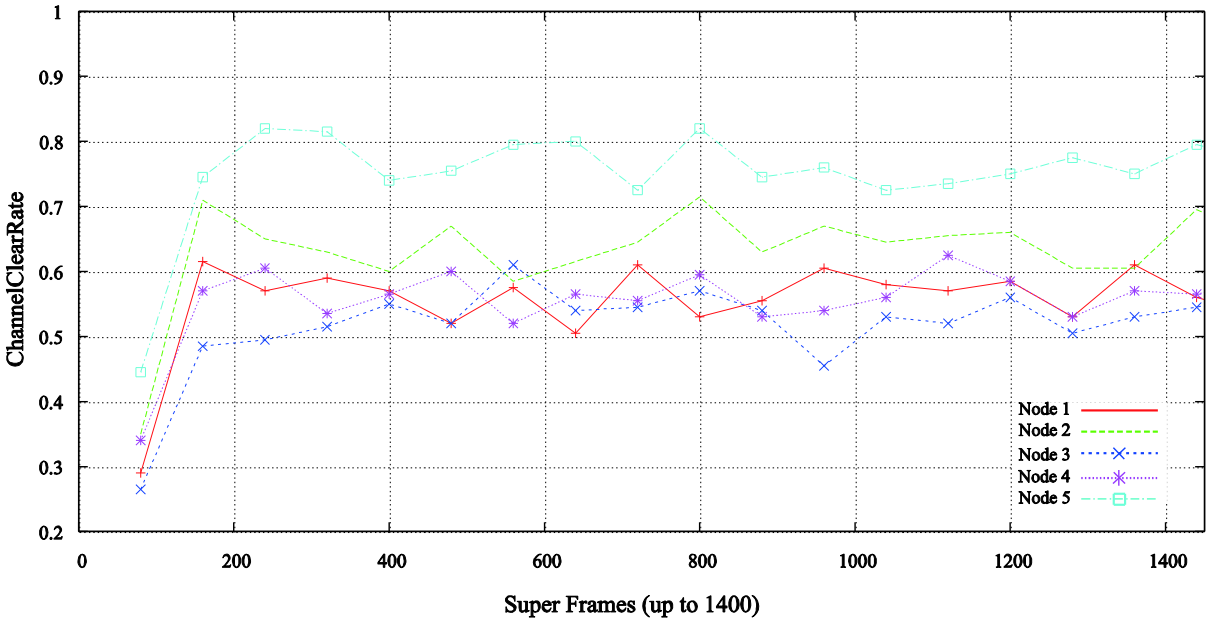


Figure 4-8: CCA_2 averaged variations for five nodes every 80 super frames (n= 80) during 500 seconds simulation run

Figure 4-9 shows in just one diagram the average for the 10 simulation runs only for n=100. Dividing the interval between the *min* and *max* values of an input such as $Channel_{Clear}Rate$ into sub-intervals showing the Low, Medium and High behavior of the parameter is the second step and is normally done by a triangular function as the fuzzy set. Of course, we had other choices of membership functions in the fuzzy library of the FuzzyLite v1.03 (Rada-Vilela, 2013), which we integrated into the Castalia implementation of the IEEE 802.15.4 for our fuzzy logic system, but for the sake of simplicity we built our fuzzy sets with triangular membership functions as shown in Figure 4-4.

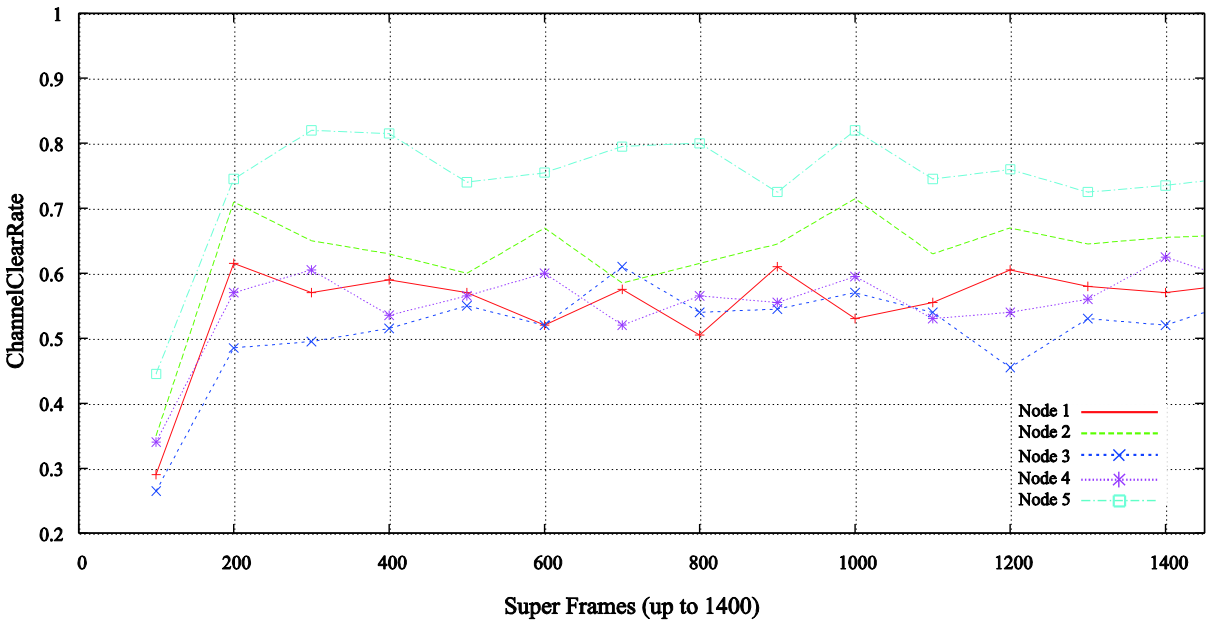


Figure 4-9: CCA$_2$ averaged variations for five nodes every 100 super frames (n= 100) during 500 seconds simulation run

The fuzzy sets defining *Channel$_{Clear}$Rate*, for example, are represented with linguistic variables of Low, Medium, and High with their defined range. For the five different existing sample rates in our scenario we defined a set of linguistic variables ranging from Low to High sets with intermediate sets of Near Low (NLow) and also Near High (NHigh) for each of which the defined range can be found at Figure 4-4. The figure also shows the *FuzzyMaxDelay* divided into six distinct ranges, making the upper bond for the produced random backoff time transiting gradually from small values to larger ones. The reason why we have considered more numbers of fuzzy sets for the *FuzzyMaxDelay* is the flexibility it gives to the fuzzy system to choose from more possible options in diverse situations and thus resulting in a more precise backoff period that fits the real time traffic conditions of the network. D1 is the smallest upper bound for the backoff a node can be assigned to, for which the defined range is 20–40 ms. The D6, on the other hand, is the largest upper bound for the random backoff period a node can be assigned ranging 100–120 ms. The rest of the devoted ranges to *FuzzyMaxDelay* can be studied from the figure itself. Therefore, for the output of our fuzzy system we have considered an interval of 2–128 symbols representing the *min* and *max* for the maximum backoff time variations for any random backoff in milliseconds. The rationale to follow into the rules of the fuzzy system based on which the effects of these two inputs are combined is explained in the next section.

4.4 **Dynamic Delayed Maximum Bound for Backoff Interval**

As discussed in Chapter 3, traffic in a typical wireless body area network, operating upon IEEE 802.15.4, can be interpreted through diverse variables that keep changing during the CSMA/CA procedure of the contention-active period. If the chosen parameters inside a CAP are not precisely characterized then it will not be possible to interpret the traffic they represent.The data rate parameter is an attribute of each sensor node whose initial value is defined for each sensor node separately when it is first configured. The second parameter that we considered is an average parameter we make over a period of time (as described in Section 4.2.1), whose initial value will always be setto zero at the time of configuration. We chose these two variables (data rate and $Channel_{Clear}Rate$) as the inputs of the fuzzy system. Other variables such as average length of collisions could also be used for the inputs of the fuzzy system. Using a fuzzy logic system as the interpreter of such variables gives us the flexibility and ease in modifying and extending the work in the future without making big changes to the main system.In building our fuzzy system, the very first step was determining the range for each of the inputs in our system and selecting the required membership functions to define the fuzzy sets of the linguistic variables we chose corresponding to each fuzzy set. Early in this chapter, we explained how the linguistic variables denoting each fuzzy set must be clearly defined and set based on the expert's experience. We then did extensive simulation runs to verify our knowledge of LOW, MEDIUM, or HIGH fuzzy sets of an input such as $Channel_{Clear}Rate$ by closely studying the range of its fluctuations through simulations. Once the *min* and *max* values of $Channel_{Clear}Rate$ were determined by our observation, we were able to use a simple triangular function to break down its variations' range into three distinct intervals of LOW, MEDIUM, and HIGH. In other words, we were able to formulate a membership function for each of the fuzzy sets in our fuzzy system settings based on our observational knowledge. Based on the descriptions in Section 4.3 we identified the *min* and *max* values for our $Channel_{Clear}Rate$ to be 0 and 1.6, which can be seen in the expression below:

$Channel_{Clear}Rate_{min} < Channel_{Clear}Rate < Channel_{Clear}Rate_{max}$;
$Channel_{Clear}Rate_{min}=0$; and $Channel_{Clear}Rate_{max}=1.6$;
And for the data rate we have a static array whose *min* and *max* values are clear at the time of initialization:

$DataRate_{min} < DataRate < DataRate_{max}$; (the *min* and *max* values can be configurable)

$DataRate_{min}$=5; and $DataRate_{max}$=125;

Membership functions are to characterize the fuzzy sets that we have determined over the variation range of a parameter. We must now be able to find a suitable function that would define the fuzzy sets of LOW, MEDIUM, and HIGH. One simple function to use (used in most of the research works) is a triangular function, such as the one we used for our proposed MAC. There are different types of membership functions to choose from with their own mathematical expressions, and a triangular function is the one that best suits our future implementations of this fuzzy algorithm into the operating system of SHIMMER sensors. This step can be considered as the fuzzifying stage, where the input values are fuzzified into their corresponding fuzzy sets by a membership function. Any fuzzy set in the defined range (or generally in the universe of discourse) can be characterized by a membership function that is denoted by μ which takes values in the [0 1] interval as the degree of membership. This is as opposed to the classical sets where the membership function of a set defined by it would only take values of either 0 or 1. For example, in our case when we say $Channel_{Clear}Rate$ is LOW we are referring to a fuzzy set that could have been described as in Figure 4-10 below:

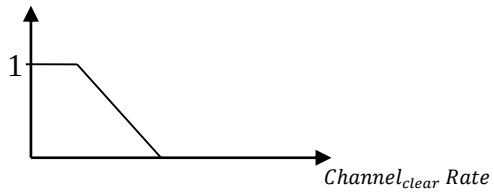


Figure 4-10: LOW fuzzy set example for $Channel_{Clear}Rate$

And $Data\ Rate$ is NHIGH (near high) is a fuzzy set, which could be described as Figure 4-11 below:

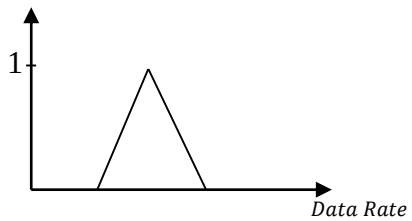


Figure 4-11: NHIGH fuzzy set example for data rate

A situation where $Channel_{Clear}Rate$ is LOW and $Data\ Rate$ is NHIGH can lead to a unique value of the *FuzzyMaxDelay* as the output of the fuzzy system, but how do these two effects

combine? A fuzzy inference system works based on defining specific rules that are also a human expert's knowledge based in the form of "IF THEN" phrases, which are mathematically expressed. The defined rules give a combined effect of the inputs of the fuzzy system in the form an output to be utilized for a planned purpose. There are still some concerns that will add to the quality of the outcome of a fuzzy rule and these concerns are mostly related to the infrastructure elements contributing to form any typical fuzzy rule as a mathematical expression. This mainly happens as "IF THEN" rules of a fuzzy system are interpreted by certain mathematical operators such as union, intersection, and complement, for which in contrast to classic algebra a handful variety exists. In the fuzzy world, any expression for a union operator is called an s-norm and any expression for an intersection operator is referred to as a t-norm. In Wang (1997, p. 48), at least four different types of s-norms and t-norms exist, each giving a different effect to the area covered by the resulting fuzzy set of such union or intersection operation over two different fuzzy sets when interpreting a rule. Therefore any 'IF THEN'clause declaring a fuzzy rule with a specific operator, will be equalized with a fuzzy *relation* in few different ways. A fuzzy relation is a fuzzy set defined in the Cartesian product of two or more crisp sets. In our case it would be the Cartesian product of the sets $Channel_{Clear}Rate$ and *Date Rate*. Such a Cartesian product can be visualized by a matrix whose elements represent the degrees of membership values of the two crisp fuzzy sets that belong to such fuzzy relation.

Considering one rule of the fuzzy system to be like this:

if ($Channel_{Clear}Rate$ is LOW and *DataRate* is NHIGH) then *FuzzyMaxDelay* is D1; (7)

Condition q

Rule1

Here the "$Channel_{Clear}Rate_{\text{LOW}}$ and $DataRate_{\text{NHIGH}}$" is a fuzzy relation defined in $Channel_{Clear}Rate*DataRate$ and can be described with a membership function as below:

$$\mu_{condition}(Channel_{clear}\ Rate, Data\ Rate)=\mu_{LOW}(Channel_{clear}\ Rate)\mu_{NHIGH}(Data\ Rate) \quad (8)$$

We can now formulate the Rule1 as below:

$$M_{\text{Rule1}}(Channel_{clear}\ Rate\ ,\ Data\ Rate\ ,\ FuzzyMaxDelay)=min[1-\mu_{\text{condition}}(Channel_{clear}\ Rate,\ Data\ Rate\ ),\ \mu_{D1}(FuzzyMaxDelay)] \quad (9)$$

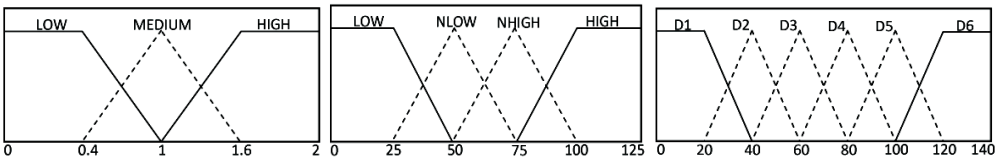
The above equation is just one of the many methods of executing a t-norm in (8) for the intersection of two fuzzy sets in an "AND" clause in expression (7) using a *min* function. A *min* function here is one of the possible t-norms to be used, and the simplest one in terms of implementations on real sensors, that can be integrated with any fuzzy system. Different operators for one expression are, however, intended to fulfill different demands represented by different systems. When two different and non-equal crisp sets are, for example, to be intersected as our example above, we may want to let the larger fuzzy set have more impact on the result, and that cannot be fulfilled by a *min* function (Wang, 1997, p. 34). A *min* function actually defines the membership function for an intersection as one possible way. The *min* is simply defined as the biggest fuzzy set that would contain both crisp fuzzy sets at hand. The same is the union of two fuzzy sets, which is denoted by *max* and would represent the smallest fuzzy set containing both crisp sets at hand. Running the t-norm is an intersection of all the $Channel_{Clear}Rate$ values that are LOW and also all the *Data Rate* values that are NHIGH. What the above formula in (9) does is manipulate the output fuzzy set, which is the *FuzzyMaxDelay* fuzzy set, into a new fuzzy set affected by running the crisp value calculated from the proposition "$Channel_{Clear}Rate_{\mathrm{LOW}}$ and $DataRate_{\mathrm{NHIGH}}$" using a chosen t-norm. The crisp value obtained from that fuzzy proposition will be passed to an *implication* function (of which a variety exists). The implication function's main role is to modify the *FuzzyMaxDelay* fuzzy set in a way that reflects the effect of "$Channel_{Clear}Rate_{\mathrm{LOW}}$ and $DataRate_{\mathrm{NHIGH}}$". How each fuzzy output (the fuzzy set for the output) for each rule is shaped by the result of its antecedent in the related rule is by different implication methods that will do such modification on the fuzzy output. For each rule defined in the system the above process must be done, thus leaving us with a group of fuzzy outputs that must be aggregated into one single fuzzy output to be further defuzzified. Once all the modified fuzzy outputs are there, they should be aggregated by an aggregation method that gives the best result and form one aggregated fuzzy output as a scalar value. Here we will look at the rationale followed in the rules of our fuzzy system based on our knowledge of the system conditions.

The rules of the fuzzy system are described in Table IV. It shows how the *FuzzyMaxDelay* depends on both $Channel_{Clear}Rate$ and also data rate when they take on different values of low, average, or high, which makes a good balance between the waiting time and the channel conditions simultaneously. We have drawn the surface diagram of the *FuzzyMaxDelay* in Matlab ("Mathworks: MATLAB", 2014) to illustrate the inputs and the output effects on each other, depicted in Figure 4-12. We can see that a node would be assigned moderate values of

FuzzyMaxDelay if the node has a good rate for its *Channel$_{Clear}$Rate* in its transmissions during last *n*super frames, and has also a relatively high data rate.

Table IV: Fuzzy rules

	FuzzyInputs				FuzzyOutput
	Channel$_{Clear}$Rate		*Data rate*		*FuzzyMaxDelay*
If	LOW	and	LOW	then	D3
	MEDIUM		LOW		D5
	HIGH		LOW		D6
If	LOW	and	NLOW	then	D2
	MEDIUM		NLOW		D4
	HIGH		NLOW		D6
If	LOW	and	NHIGH	then	D1
	MEDIUM		NHIGH		D5
	HIGH		NHIGH		D5
If	LOW	and	HIGH	then	D1
	MEDIUM		HIGH		D1
	HIGH		HIGH		D4



(Fuzzy rules: Correlation of inputs' values and the output value to locate a maximum bound). For *Channel$_{Clear}$Rate*: {LOW, MEDIUM, HIGH}={(0,1), (0.4, 1.6), (1,2)}. For data rate: {LOW, NLOW, NHIGH, HIGH}={(0,50), (25,75), (50,100), (75,125). For *FuzzyMaxDelay*: {D1…D6}={(0,40), (20,60), (40,80), (60,100), (80,120), (100,140). (NLOW=Near Low, NHIGH= Near High)

Therefore the node has had a higher chance to access the channel while other nodes received lower chances. On the other hand, the smaller *FuzzyMaxDelay*s assigned are to those nodes with a relatively high data rate but a very low *Channel$_{Clear}$Rate*, which would almost always restrict them from having access to a channel. The fact is there is not just one rule. The more complete the collection of rules, the better coverage of all the system's states under study. The defined rules of the fuzzy system are towards making a balance between the successful access experiences of all nodes considering the last *n* super frames access history so that no node undergoes massive backoff delay or conquers the channel for a long time leading to diminishing other nodes' chances to send out their data to the coordinator. This can be fulfilled by tracing each rule:

if ***Channel$_{Clear}$Rate*** is low and **data rate** is low then ***FuzzyMaxDelay*** is low;

if ***Channel$_{Clear}$Rate*** is high and **data rate** is low then ***FuzzyMaxDelay*** is high;

if ***Channel$_{Clear}$Rate*** is high and **data rate** is high then ***FuzzyMaxDelay*** is medium;

.

.

The rule table defined in Table IV has 12 rules in total. When combining the different effects of the different values of inputs, the *FuzzyMaxDelay* will actually take on six different intervals in its fuzzy triangular membership function. This is to give more precision to the length of the backoff period decided for the nodes with respect to the history of their past trials and their data rate.

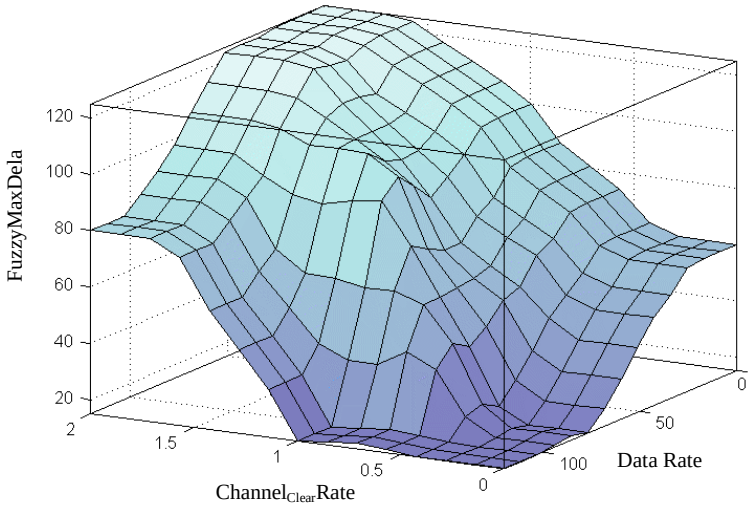


Figure 4-12: Surface diagram of the fuzzy rules (visualizing the correlation between inputs and output)

The fuzzy library that has been created in Castalia to implement this fuzzy system is based on the work in Rada-Vilela (2013) called FuzzyLite, which is a fuzzy library implementation in C++ language. It is a cross-platform, free and open-source fuzzy logic control library that allows easy implementation of a fuzzy logic system by utilizing object-oriented programming. Different controllers are available in FuzzyLite such as Mamdani, Takagi-Sugeno, Larsen, Tsukamoto, and Inverse Tsukamoto. Detailed descriptions of these controllers are not the focus of this research. We however used Mamdani as the controller to implement our fuzzy system. The components of this fuzzy library are extendable based on the implementation requirements, which give extra flexibility in using it. Some of the advantages of using this fuzzy logic developer, according to (Rada-Vilela, 2013) are:

- It is free and open source
- It has a commercially friendly license
- Many available design choices
- More features compared to all current fuzzy logic libraries up to date
- Very easy to use

- Available for major platforms such as Linux, MacOSX, Windows, and others.

Here is the source code of our fuzzy implementation by using the FuzzyLite library in Castalia:

```
#include "FuzzyLite.h"
void Mac802154c::continueCSMACA()
 #initializing the fuzzy system
 flScalar maxFuzzyDelayTime;
 FuzzyOperator& op = FuzzyOperator::DefaultFuzzyOperator();
 FuzzyEngine engine("complex-mamdani", op);

 #initializing the fuzzy engine
 engine.hedgeSet().add(new HedgeNot);
 engine.hedgeSet().add(new HedgeSomewhat);
 engine.hedgeSet().add(new HedgeVery);

 #initializing the fuzzy input set of data rate
 #dR: data Rate
 InputLVar* dR = new InputLVar("dR");
  dR->addTerm(new ShoulderTerm("LOW", 25, 50, true));
  dR->addTerm(new TriangularTerm("NLOW", 25, 75));
  dR->addTerm(new TriangularTerm("NHIGH", 50, 100));
  dR->addTerm(new ShoulderTerm("HIGH", 75, 100, false));
 engine.addInputLVar(dR);

 #initializing the fuzzy input set of channel clear rate
 #cCR: channel clear rate
 InputLVar* cCR = new InputLVar("cCR");
  cCR->addTerm(new ShoulderTerm("LOW", 0.4, 1, true));
  cCR->addTerm(new TriangularTerm("MEDIUM", 0.4, 1.6));
  cCR->addTerm(new ShoulderTerm("HIGH", 1, 1.6, false));
 engine.addInputLVar(cCR);

 #initializing the fuzzy output set of max delay
 #fMD: Fuzzy max delay
 OutputLVar* fMD = new OutputLVar("fMD");
  fMD->addTerm(new ShoulderTerm("D1", 20, 40, true));
  fMD->addTerm(new TriangularTerm("D2", 20, 60));
  fMD->addTerm(new TriangularTerm("D3", 40, 80));
  fMD->addTerm(new TriangularTerm("D4", 60, 100));
  fMD->addTerm(new TriangularTerm("D5", 80, 120));
  fMD->addTerm(new ShoulderTerm("D6", 100, 120, false));
 engine.addOutputLVar(fMD);

 #initializing the fuzzy roles
 RuleBlock* block = new RuleBlock();
 block->
 addRule(MamdaniRule("if cCR is LOW and dR is LOW then fMD is D3"))
 addRule(MamdaniRule("if cCR is MEDIUM and dR is LOW then fMD is D5"))
 addRule(MamdaniRule("if cCR is HIGH and dR is LOW then fMD is D6"))
 addRule(MamdaniRule("if cCR is LOW and dR is NLOW then fMD is D2"))
```

```
addRule(MamdaniRule("if cCR is MEDIUM and dR is NLOW then fMD is D4"))
addRule(MamdaniRule("if cCR is HIGH and dR is NLOW then fMD is D6"))
addRule(MamdaniRule("if cCR is LOW and dR is NHIGH then fMD is D1"));
addRule(MamdaniRule("if cCR is MEDIUM and dR is NHIGH then fMD is D5"))
addRule(MamdaniRule("if cCR is HIGH and dR is NHIGH then fMD is D5"))
addRule(MamdaniRule("if cCR is LOW and dR is HIGH then fMD is D1"))
addRule(MamdaniRule("if cCR is MEDIUM and dR is HIGH then fMD is D1"))
addRule(MamdaniRule("if cCR is HIGH and dR is HIGH then fMD is D4"));
engine.addRuleBlock(block);

#set the fuzzy inputs
dR->setInput(dataRateValue);
cCR->setInput(channelClearRateValue);

#run fuzzy engine
engine.process();

#defuzziy the fuzzy output
fuzzyMaxDelayValue = fMD->output().defuzzify();

int rnd = genk_intrand(1, fuzzyMaxDelayValue) + 1;
simtime_t CCAtime = rnd * (unitBackoffPeriod * symbolLen);
```

4.5 Dynamic Delayed Maximum and Minimum Bounds for Backoff Interval

The dynamic backoff window explained in the previous sections in this chapter that would place a traffic adaptive maximum bound on the backoff window can be expanded to influence the minimum bound of the backoff window as well. However, the range of variations for the output of the fuzzy system will not be great for the minimum end since the default minimum value of BE parameter in the standard is three. Considering the possible values of BE parameter to be as small as one to as large as seven, the biggest possible value to be considered as the minimum bound for BE could be five for very certain applications with probably a high load. Therefore, we have not considered a dual fuzzy system for the implementations of our proposed MAC that would alter the backoff window from its both ends (minimum and maximum limits). A default value of normally two and three for BE_{min} in the standard will not leave much room for enhancement by a dynamic approach as it would only let the variation range to be between one and eight (2^0 to 2^3). Nevertheless we implemented the concept once and did some preliminary testing to see the effects, which are presented in Section 5.1.2. Figure 4-13 shows a simple overview of this fuzzy system. Data rate and $Channel_{Clear}Rate$ are fuzzified into linguistic variables that we have defined for each. The defined ranges for both inputs are shown in the figure.

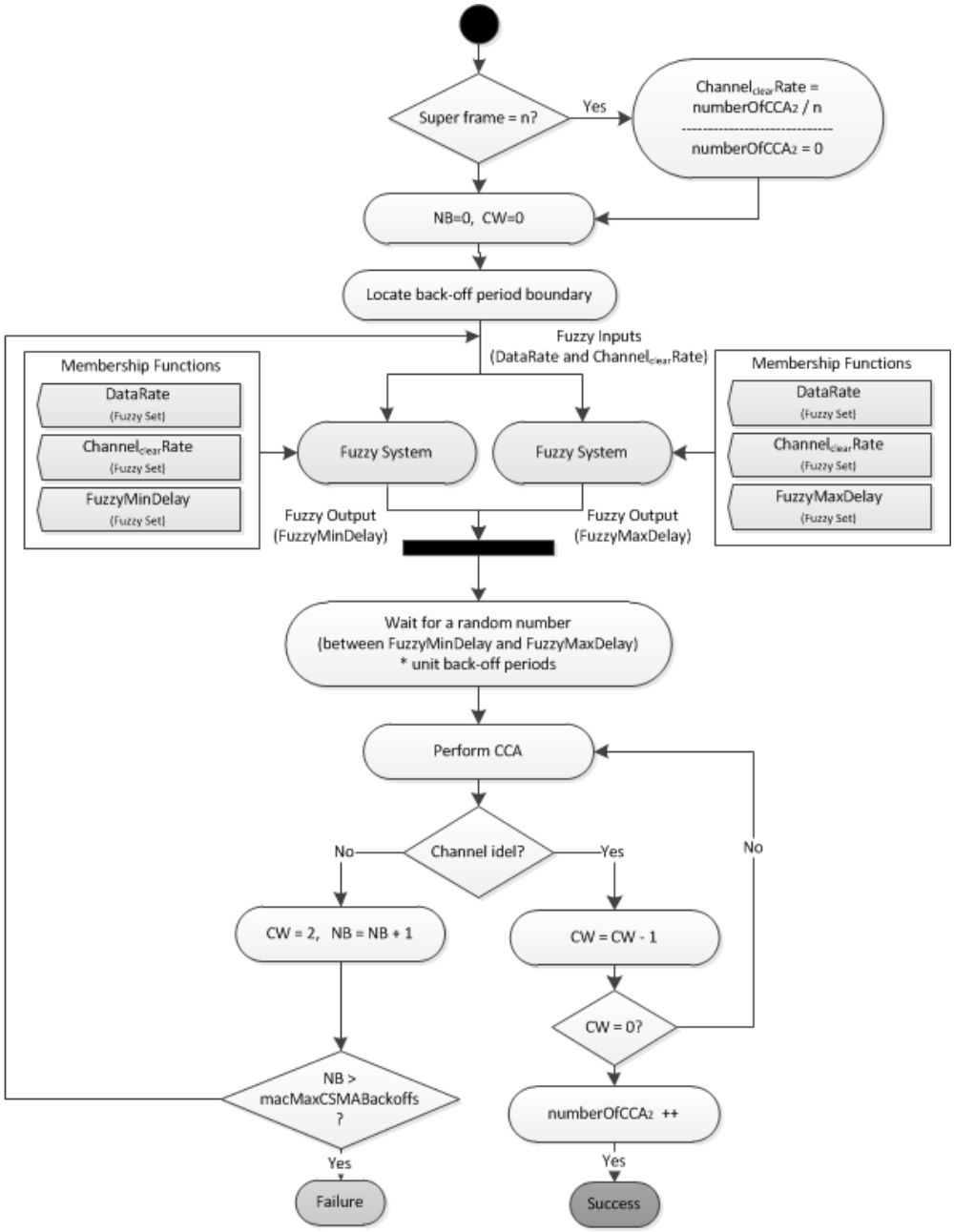


Figure 4-13: Integration of dynamic fuzzy-enabled MAC for both *min* and *max* bounds of BE window into CSMA/CA algorithm

Data rate range reflects the *min* and *max* of the data rate array assigned to nodes in our simulations as discussed before.For this implementation we have a dual fuzzy system running in parallel and based on the same rules that combine the effects of their two inputs. The ranges for the output differ for each fuzzy system. For *FuzzyMinDelay* the assumption is a BE_{min} parameter initiated to five, which as declared earlier best fits a very high load application. For *FuzzyMaxDelay* the output range is exactly the same as described in Section 4.2. The rules to apply are also the same as in Table IV.

A modified version of the fuzzy system overview in Figure 4-4 is shown in Figure 4-14.

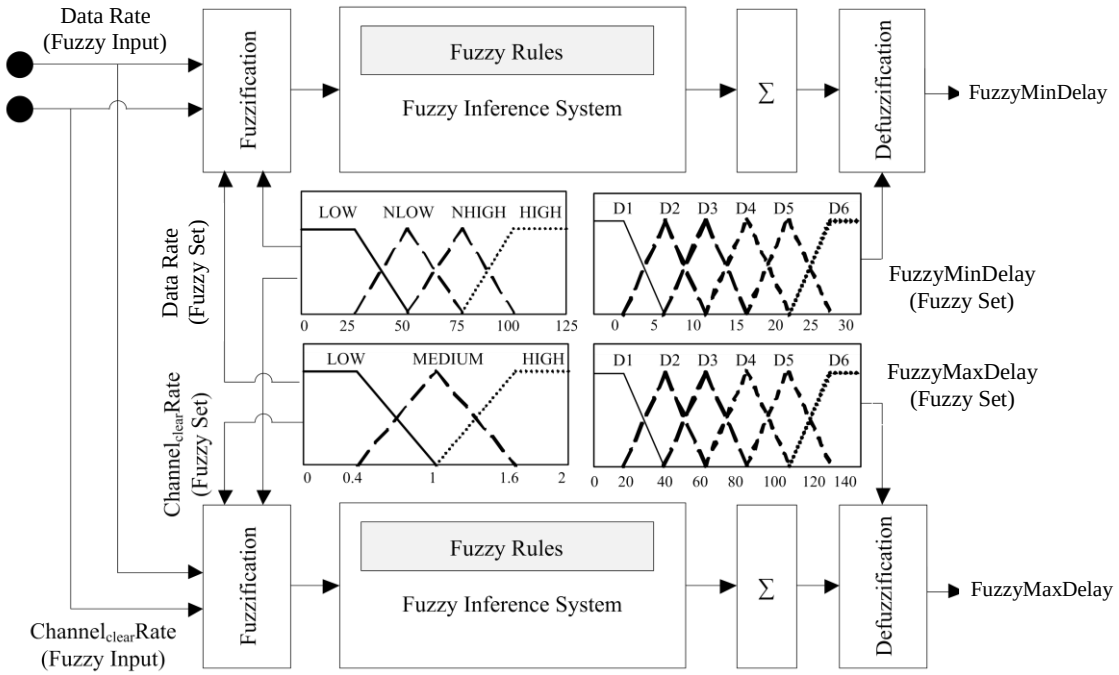


Figure 4-14: Fuzzy system overview for both *min* and *max* limits of the backoff winodow

We have devoted Section 5.1.2 to the reliability and delay measurements of this dual implementation except that the entire focus of this research is on the fuzzy-enabled MAC protocol which would only lead to a dynamic and traffic adaptive adjustment of a maximum bound for the backoff window. The code architecture in the Castalia simulator described in the next section would also discuss the enhanced CAMA/CA mechanism of the IEEE 802.15.4 in regard to dynamic and traffic adaptive values of the maximum bound for the backoff window.

4.6 Code Architecture in the Castalia Simulator

4.6.1 Castalia: The Selected Simulator

Castalia ("Castalia", 2011) is a good choice for doing the simulation parts as an event-driven, modular simulator that is designed specifically for wireless body area network implementations. It has the significance of realistic channel models for body area networks and is based on OMNeT++ ("OMNet++ Network Simulation Framework", 2014). The simulator was designed by NICTA ("National Information and Communications Technology Australia", 2011). The motivation behind building such a simulator as reported in the manual is declared

as *"because the available WSN simulators were falling short of the current state of the art modeling done in sensor networks. Especially in communication where the impact to the result can be significant, models remain simplistic or unsuitable for short range low power communications despite the existence of proper models developed the last couple of years"*.The Castalia project was first motivated by a lack of accurate enough channel models which are mostly needed for short range communications. It is stated in its manual that the wireless channel becomes extremely difficult to model when it comes to WBAN, due to a changing environment. Several reasons have been identified as why there have not been great advances so far in terms of accurate modeling of the wireless channel, which are summarized as:

- Complexity of the problem
- Smaller commercial benefits offering least support for the costly job of modeling
- Lack of interest in PHY layer from researchers worldwide ("Castalia", 2011, p. 51).

The accurate models offered by NICTA are the result of hundreds of thousands of measurements that have been analyzed to achieve average values of path losses around the body. We have used these path-loss models for our simulation scenarios in Castalia, which will be described in more detail, through the assumed topology of the physiological sensor nodes in simulations. For real experiments in Chapter 6, however, the sensors are not really placed on a body. The reliability measurements for real experiments were carried out for five SHIMMER sensor platforms with specific data rates assigned to them and placed on a static surface in normal indoor conditions. The main aim of the real experiments in Chapter 6 is to validate the feasibility of our fuzzy-enabled MAC algorithm to execute on real sensor platforms such as SHIMMER. Reliability and energy measurements prove the efficiency of such a method to push the future of this research towards deploying the SHIMMER sensor platforms with our fuzzy-enabled MAC on the body of a patient with dynamic physiological data rates.

There are a few advantages in using Castalia as our simulator tool. The main benefit would be the availability of IEEE 802.15.4 MAC code in its library. Plus Castalia provides a variety of scenarios to evaluate MACs in body area networks. There are also implementations of primary MAC protocols such as S-MAC (Ye, Heidemann & Estrin, 2002) and T-MAC (Gong, Liu,

Mao, Chen, & Xie, 2005) in wireless sensor networks available in it. The main features of Castalia are classified as below ("Castalia", 2011, p. 5):

- Advanced channel model based on empirically measured data
- Advanced radio model based on real radios for low-power communication
- Extended sensing modeling provisions
- Node clock drift
- MAC and Routing protocols available
- Designed for adaptation and expansion

The advanced channel model in Castalia allows for a map of path loss especially designed for close-to-reality simulations of sensor nodes on a body and the unique environment they are deployed in, plus consideration of temporal variations in channel and mobility. It is indicated in the manual ("Castalia", 2011, p. 51) that "*we are confident to claim that, concerning the wireless channel, Castalia is the most realistic simulator one could find for WSN and BAN*". The last feature declared in the above list highlights the capability of Castalia to let its users implement and import their own algorithms and protocols into the simulator with utmost ease, which is thanks to its modularity and its reliance on OMNeT++, which is an event-driven base. The existing path-loss model in Castalia is based on real on-body measurements that model the temporal variations. A series of message passing and function callings make a node module in Castalia simulator to function. Although the existing modules in Castalia are highly tunable through their parameters, creating a new protocol is also possible by means of creating abstract classes ("Castalia", 2011, p. 8). The language to write the modules with is NED. The default values of the parameters for each particular module will also be defined in its NED file, which is denoted with a .ned extension in the simulator. The simulation scenario can be defined through different configurations set in the main configuration file that will be executed for each simulation run. This file is named "omnetpp.ini" which compared to NS-2 Simulator, acts as tool command language (TCL) file where all the configurations and scenarios are defined. The trace files created by each run of this configuration file can then be observed and examined for different evaluations such as energy, delay, and reliability measures. We will learn more about the omnetpp.ini file used for our simulations along with the specific configurations and also initial values of main parameters in Chapter 5. We will end this section by looking at the path loss model in Castalia.

The close-to-reality path loss model presented in Castalia is a result of explicitly setting their own path loss map in Castalia and not following the lognormal shadowing model that is usually used for simulations of wireless sensor networks. A lognormal shadowing model is suitable for WSN because the distance between two nodes could possibly be in terms of a hundred meters but it would not appear accurate for path loss models of the links in a WBAN. A cleaner and more independent way of controlling the correlation between two directions of a link is introduced by Castalia where it adds a separate Gaussian zero-mean random variable with standard deviation (default=1.0) to return an average path loss ("Castalia", 2011, p. 52). The average path losses have been measured using a test bed in NICTA and are available in the simulator through *SN.wirelessChannel.pathLossMapFile* parameter. The lines of this file follow the below format:

TxNodeID>RxNodeID1:dB_value, RxNodeID2:db_value,…

TxNodeID simply means the transmitting node with its ID indicating which node is transmitting in this star topology, and because it is a star topology it does not have an index for the one and only transmitting node. The links from this transmitting node to the other nodes in the WBAN each represent a specific path loss value, denoted by dB_value. For example, RxNode1:56 would simply mean the path loss between Node0 (as the transmitting node) and Node1 is 56 dB. The table of the path loss used for our simulations will be explained in Chapter 5.

4.6.2 **Code Hierarchy**

In this section we will draw a map of Castalia's main components to help understand the structure of our code better. The reader is encouraged to refer to APPENDIX II for a detailed map of the CSMA/CA procedure in IEEE 802.15.4. Earlier in Chapter 1 we explained the main MAC services by MLME and DATA service points in IEEE 802.15.4. Comprehending and following the steps in the detailed flow chart of CSMA/CA in APPENDIX II will be easier by referring to the described primitives in Figure 1-13 and also the standard's manual itself ("IEEE, Std. 802.15.4", 2003). Most of the concepts explained in this thesis and also where the intended fuzzy-enabled MAC algorithm is going to be implemented are associated with *initiateCSMACA()*, and *continueCSMACA()* modules in the APPENDIX II flowchart. Here in this section we draw a map of the file structure we followed in the Castalia simulator to implement the fuzzy-enabled MAC algorithm described in this chapter. Castalia uses

OMNET++ features to define the architecture of a sensor node. All definitions are described and implemented in the Castalia-3.2/src directory. Here is a snapshot of the Castalia folder hierarchy:

```
Castalia-3-2
----Simulations
--------BANtestNewMAC
------------Omnetpp.ini
--------Parameters
----src
--------node
------------application
---------------- throughputTest
------------communication
----------------mac
--------------------mac802154
--------------------mac802154fuzzyMethod1
--------------------mac802154fuzzyMethod2
----------------radio
----------------routing
------------mobilityManager
------------resourceManager
------------sensorManager
--------physicalProcess
--------wirelessChannel
```

The Node module is the main component: it includes most of the other modules such as the routing or radio module (in the src/node/communication folder), and of course the application modules (in thesrc/node/application folder). Regarding the application module, it must be understood that the behavior of a sensor node depends on which application module is specified in the omnetpp.ini file with the `SN.node[*].ApplicationName="ThroughputTest"` line. If you look at the node, you can easily understand that a sensor node is composed of a mobility module, a resource module, a device manager module, a network module, and an application module, which implements what the sensor is doing. Here are the components of node.communication with its three main sub-modules of Radio, MAC, and Routing.

```
package node.communication;

module CommunicationModule {
 parameters:
    string MACProtocolName = default ("BypassMAC");
    string RoutingProtocolName = default ("BypassRouting");

gates:
    output toApplicationModule;
```

```
    output toNodeContainerModule;
    input fromApplicationModule;
    input fromNodeContainerModule;
    input fromResourceManager2Net;
    input fromResourceManager2Mac;
    input fromResourceManager2Radio;

submodules:

    Radio: node.communication.radio.Radio;
    MAC: <MACProtocolName> like node.communication.mac.iMac;
    Routing: <RoutingProtocolName> like
   node.communication.routing.iRouting;

 connections:

    fromApplicationModule --> Routing.fromCommunicationModule;
    Routing.toCommunicationModule --> toApplicationModule;
    Routing.toMacModule --> MAC.fromNetworkModule;
    MAC.toNetworkModule --> Routing.fromMacModule;
    MAC.toRadioModule --> Radio.fromMacModule;
    Radio.toMacModule --> MAC.fromRadioModule;

    fromNodeContainerModule --> Radio.fromCommunicationModule;
    Radio.toCommunicationModule --> toNodeContainerModule;

    fromResourceManager2Net --> Routing.fromCommModuleResourceMgr;
    fromResourceManager2Mac --> MAC.fromCommModuleResourceMgr;
    fromResourceManager2Radio --> Radio.fromCommModuleResourceMgr;
```

Figure 4-15 shows the component architecture in Castalia and the message passing between layers of the communication stack.

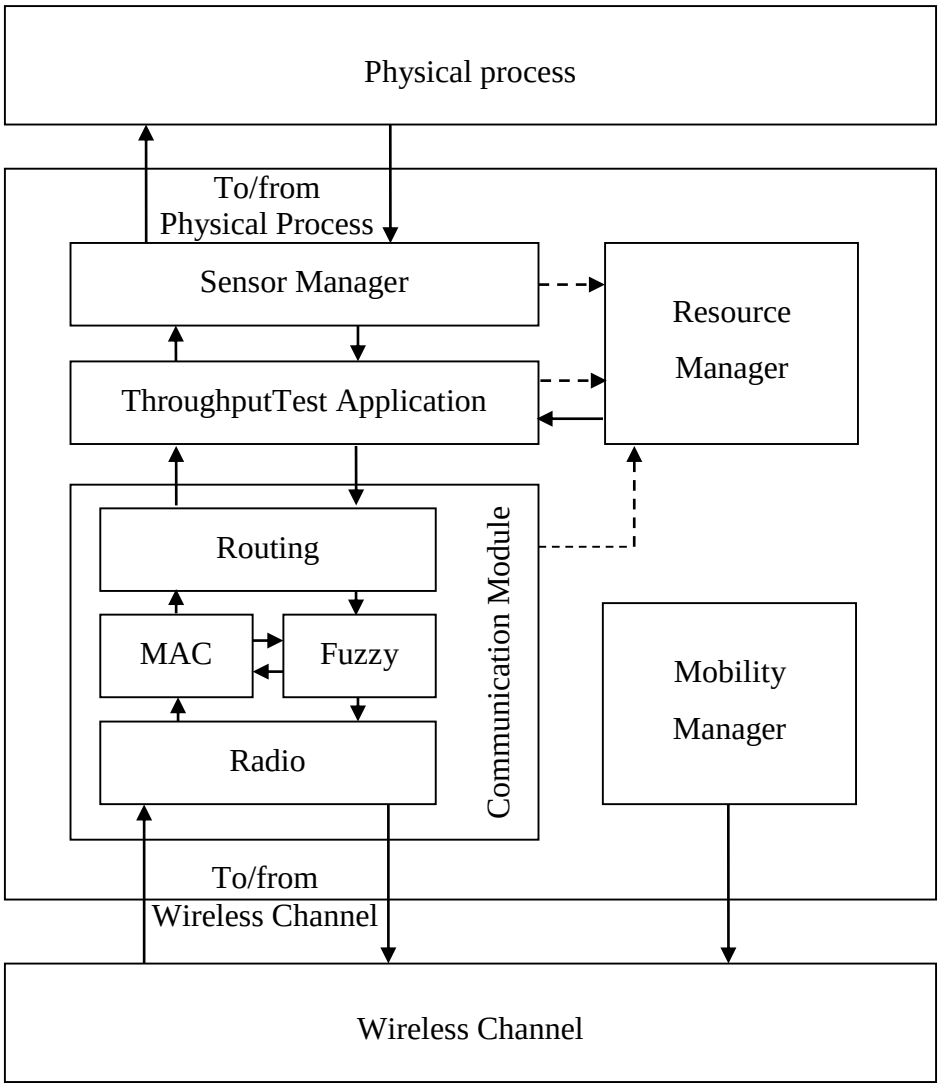


Figure 4-15: Castalia's architecture

4.7 Chapter Summary

In this chapter we discussed the main rationale incorporated into our fuzzy-enabled MAC algorithm. We described why fuzzy logic proved to be an effective tool in implementing the technique we had in mind. The validation of choosing fuzzy logic as a tool in our method can also be assessed through the reviewed works in Section 3.4, which discusses some of the dynamic methods in traffic control exploiting fuzzy logic. The elements of the designed fuzzy logic in our proposed MAC protocol had to be discussed before we move to the performance evaluation in the next chapter. Therefore we conducted extensive simulations to investigate the range of variations for $Channel_{Clear}Rate$ as one of the inputs of our fuzzy system before elaborating on the rules of the fuzzy system. We then discussed in Section 4.2.2 the main motivation behind choosing a second method for calculating $Channel_{Clear}Rate$. Although the fuzzy system has been tested on both the minimum and maximum bounds of the backoff window in the CSMA/CA mechanism, we have only discussed the dynamic minimum bound

for the backoff window in Section 4.5 of this chapter and have conducted limited evaluations on its performance in Section 5.1.2. All other evaluations and discussions and implementations of our fuzzy-enabled MAC are based on only a dynamic maximum bound for the backoff window, as investigated in Section 4.4. Related pseudo-codes are also discussed wherever applicable to help understand the functionality of the proposed method better.

5. Chapter 5: SIMULATION AND EVALUATION

This chapter summarizes some of the evaluations we have carried out in the simulator along with some real energy measurements on SHIMMER (SHIMMER Sensing Technology, 2014) sensor platforms, which will be explored in Section 5.2. The chapter starts by giving a brief introduction to the adjustment of configurations and their definitions in the Castalia simulator. We then evaluate our fuzzy-enabled MAC technique through the two different methods of calculating the $Channel_{Clear}Rate$ (M1 & M2) that we became familiar with in the previous chapter. The initial values of key parameters are discussed in this chapter along with the describing parameters of any considered scenario such as the data rates assigned, the number of nodes, and path loss values. The chapter also includes some real energy testing on SHIMMER sensor platforms, which are discussed extensively in Section 5.2. Section 5.2.2 focuses on developing a micro fuzzy engine to be developed in the operating system of the SHIMMER sensor platforms, TinyOS ("TinyOS, an Operating System Designed for Low Power Wireless Devices", 2013). The micro fuzzy engine developed is to aid us with running a fuzzy algorithm (similar to our fuzzy-enabled MAC algorithm) at the application layer. The same fuzzy engine will then be used in Chapter 6 for implementations of the proposed fuzzy-enabled MAC in the CSMA/CA mechanism of IEEE 802.15.4 in SHIMMER sensor platforms. The developed fuzzy engine in this chapter is mainly used for the real monitoring of SHIMMER sensor platforms' energy behavior over a long period to justify the energy efficiency of our fuzzy-enabled MAC algorithm before we implement it at MAC layer in Chapter 6. We also describe a simplifying technique to reduce the complexity of the fuzzy system for our fuzzy-enabled MAC in this chapter, which is explained in Section 5.2.2.2. The significance of this simplifying technique manifested when we were implementing the fuzzy system in CSMA/CA mechanism of SHIMMER sensor platforms described later in Chapter 6. We, however, did evaluate the reliability factor with such simplifying technique through simulations as well, which is discussed in Section 5.2.2.3 as the last section of this chapter.

5.1 Configuration Parameters and Assumptions

The assumptions described in this section are explained through different configurations set in the omnetpp.ini file (Section 4.6.1) where different configurations can be created in Castalia to create different scenarios. The general parameters at the beginning of an omnetpp.ini file define the basic scenario of the simulation, which includes the number of nodes, the simulation time, field size, deployment type, path loss model to be used, radio parameters etc. The simulation

time is defined as "`sim-time-limit`" which is the only generic OMNet++ parameter in Castalia. A MAC frame for our simulation scenarios is assumed to be 120 ms. If we have a close look at the omnetpp.ini that we created for our simulations, we will notice a variety of possible configurations that are available for different simulation scenarios. Not all the configurations are to be used in every scenario and choosing a configuration to be used in a simulation run can be done in the command line. In the command line, an available and defined configuration in the omnetpp.ini file can be executed using the `-c` switch (detailed in the manual ("Castalia", 2011, p. 11)). A configuration in omnetpp.ini file is generally denoted with the word `Config` followed by the name given to that particular configuration and is written within brackets. Some of these configurations are briefed as below in Table V:

Table V: Simulation configuration parameters as in omnetpp.ini file

[Config VarySimTime] sim-time-limit = ${SimulationTime=500s,1000s,2000s,3000s,4000s}
[Config ZigBeeMAC] SN.node[*].Communication.MACProtocolName = "Mac802154" SN.node[0].Communication.MAC.isFFD = true SN.node[0].Communication.MAC.isPANCoordinator = true SN.node[*].Communication.MAC.phyDataRate = 1024 SN.node[*].Communication.MAC.phyBitsPerSymbol = 2
[Config ZigBeeMACa] SN.node[*].Communication.MACProtocolName = "Mac802154a" SN.node[0].Communication.MAC.isFFD = true SN.node[0].Communication.MAC.isPANCoordinator = true SN.node[*].Communication.MAC.phyDataRate = 1024 SN.node[*].Communication.MAC.phyBitsPerSymbol = 2
[Config GTSon] SN.node[*].Communication.MAC.requestGTS = 5
[Config GTSoff] SN.node[*].Communication.MAC.requestGTS = 0

```
[Config allNodesDifferentRate]
SN.node[0].Application.packet_rate = 100
SN.node[1].Application.packet_rate = 5
SN.node[2].Application.packet_rate = 10
SN.node[3].Application.packet_rate = 10
SN.node[4].Application.packet_rate = 20
SN.node[5].Application.packet_rate = 20
SN.node[6].Application.packet_rate = 30
SN.node[7].Application.packet_rate = 30
SN.node[8].Application.packet_rate = 50
SN.node[9].Application.packet_rate = 50
```

```
[Config oneNodeVaryPower]
SN.node[2].Communication.Radio.TxOutputPower = ${power="-
10dBm","-12dBm","-15dBm","-20dBm"}
```

```
[Config setPower]
SN.node[*].Communication.Radio.TxOutputPower = "-30dBm
```

```
[Config varyReTxNum]
SN.node[*].Communication.MAC.maxPacketTries =
${pktTries=1,2,3,4}
```

The selected configurations once created in the configuration file can be run simultaneously by putting them together with Castalia script in the console command line. Having said that, no two configurations to be combined can have an overlapping parameter. The `SN` (sensor network) denotes the topmost composite module. We used `BANtest` example file in Castalia as the base configuration file and scenario for our simulation runs, since the simulation examples provided in Castalia have almost all their parameters set and configured at reasonable values to reflect a real WBAN situation. We then created different configurations in its omnetpp.ini file, which we will refer to for each simulation run. The four modules that can be dynamically selected in Castalia are MAC, routing, application, and mobility manager, which we saw before in Chapter 4. The Application Module, once set, will identify how the traffic is going to flow in the network. For instance, by choosing the application module as `ThroughputTest` in the omnetpp.ini file, the sensor nodes in a star topology will send their traffic to the coordinator node, which by default is Node 0. The packet rate and size are easily configurable. Once the transmissions start by running the simulation, the information in regard to such data will be saved in a trace file, which is also declared and activated in the omnetpp.ini file. The simulations can get configurations at the command line but the intended configuration

must have been first defined in the configuration file omnetpp.ini. The intended parameters to look for and study can then be derived by running `CastaliaResults` script on them. More detailed descriptions of how to derive results and how to run filters and different repetitions of simulations appear in the manual.

In Table V, the first configuration, `[ConfigVarySimTime]`, is for setting the duration of the simulation run. If this configuration is set to an array of different values, as set in Table V, that means the simulations will run (in this case) for five times and each time with a different duration assigned.

```
[Config VarySimTime]
sim-time-limit = ${SimulationTime=500s,1000s,2000s,3000s,4000s}
```

In our simulation scenarios we use different configurations to set this parameter. Sometimes we use one value for it that the simulation will run for that duration once. For some other scenarios we may have to assign different values to it as shown above to let the simulation run with different durations. In the example above, the `sim-time-limit` gets five different values through a configuration defined as `VarySimTime`. The simulation in this case runs five times and each time with a different duration. We will explain all the assigned values for each scenario on the spot.

Radio is part of the Communication composite module for every node so there is one radio module for each node and they get their default values from `RadioParametersFile`. It can be seen how it is set in omnetpp.ini file here:

```
SN.node[*].Communication.Radio.RadioParametersFile =
"../Parameters/Radio/BANRadio.txt"
SN.node[*].Communication.Radio.symbolsForRSSI = 16
SN.node[*].Communication.Radio.TxOutputPower = "-15dBm"
```

The parameters are set for each and every node. The values assigned could be different (for which each node index has to be specified) or the same, which could simply be set by *. The possibility of assigning a few different `TxOutputPower` values also exists, which can be stated as {TxPower="-5dB", "-10dB", "-15dB"}, for instance, which would make the simulator run the simulations individually for each of these transmit powers.

The path loss model used for the simulations is derived from Castalia's group experimental measurements.It is defined in pathLossMap.txt file and is included in omnetpp.ini as follows:

```
SN.wirelessChannel.pathLossMapFile =
"../Parameters/WirelessChannel/BANmodels/pathLossMap.txt"
SN.wirelessChannel.temporalModelParametersFile=
"../Parameters/WirelessChannel/BANmodels/TemporalModel.txt"
```

Earlier in Chapters 1 and 3 we discussed the difficulty of the wireless medium to be modeled specially for WBANs due to a remarkably changing environment such as the human body with a great level of mobility. The path-loss model in the above mentioned file follows the path-loss pattern achieved through extensive simulation runs at NICTA. For each specific simulation scenario, though, the parameters of the path-loss model must be tuned carefully to reflect the simulation scenarios as closely as possible. The path-loss values in the file `pathLossMap.txt`will be applied to the simulation scenarios being carried out in our research. The content of the `pathLossMap.txt` file is shown in Table VI:

Table VI: Nodes' relative positions on body and path loss values

Nodes' description	Path loss values
#0 is Right-hip	0>1:56,2:40,3:59,4:54,5:58 1>0:56,2:52,3:52,4:58,5:61 2>0:40,1:52,3:58,4:54,5:61 3>0:59,1:52,2:58,4:50,5:63 4>0:54,1:58,2:54,3:50,5:63 5>0:58,1:61,2:61,3:63,4:63
#1 is Left-wrist	
#2 is Right-wrist	
#3 is Left-ankle	
#4 is Right-ankle	
#5 is chest	

We saw earlier in Section 4.6.1 what each line of the above format means. If Node0 is assumed to be the coordinator (as assumed in our simulation scenarios) then the path loss experienced with each single node (given the position where that node is placed on the body) can be derived from the table above. For our simulation scenarios of five nodes we have used the exact path loss model and the exact node positions. These positions can also be seen in Figure 2-1 of Chapter 1 except the EEG sensor. The positioning of the nodes for simulations of 10 nodes and their path loss values will be similar to that of five nodes with relative positions of Left/Right knee, Left/Right elbow, and left hip and with similarly assigned path loss values as those in Table VI.

In Table V the two configurations of `ZigbeeMAC` and `ZigbeeMACa` refer to the MAC specifications of the IEEE 802.15.4 standard and its modified fuzzy-enabled version of our proposed MAC respectively. Therefore the `ZigbeeMACa` configuration, once selected at the command line as a configuration, will run the simulation scenario based on our proposed fuzzy-enabled MAC. For comparisons of the average number of received packets at the coordinator these two configurations are run at the command line together.

```
[Config ZigBeeMAC]
SN.node[*].Communication.MACProtocolName = "Mac802154"
SN.node[0].Communication.MAC.isFFD = true
SN.node[0].Communication.MAC.isPANCoordinator = true
SN.node[*].Communication.MAC.phyDataRate = 1024
SN.node[*].Communication.MAC.phyBitsPerSymbol = 2

[Config ZigBeeMACa]
SN.node[*].Communication.MACProtocolName = "Mac802154a"
SN.node[0].Communication.MAC.isFFD = true
SN.node[0].Communication.MAC.isPANCoordinator = true
SN.node[*].Communication.MAC.phyDataRate = 1024
SN.node[*].Communication.MAC.phyBitsPerSymbol = 2
```

The `GTSon` and `GTSoff` configurations enable us to choose between a MAC protocol that has the capability of guaranteed time slots or not. Obviously for our simulations this configuration has to be set to `GTSon`, which is what we have done as a result of a combined TDMA and CSMA approach.

```
[Config GTSon]
SN.node[*].Communication.MAC.requestGTS = 5
[Config GTSoff]
SN.node[*].Communication.MAC.requestGTS = 0
```

As can be seen by the default configuration in Castalia, the number of available GTS slots is set to five for the CFP of the super frame.

Different configurations can also be created to set the values of data rates. If the data rate of all the sensors sending data to the coordinator is same the data rate can be configured through `SN.node[*].Application.packet_rate`. The configurations to set the data rate of the nodes in our simulations will be described later for each created scenario. One sample configuration of data rate, `allNodesDifferentRate`, is brought in Table V which gives different data rates to different nodes in the simulation as shown below:

```
[Config allNodesDifferentRate]
SN.node[0].Application.packet_rate = 100
SN.node[1].Application.packet_rate = 5
SN.node[2].Application.packet_rate = 10
SN.node[3].Application.packet_rate = 10
SN.node[4].Application.packet_rate = 20
SN.node[5].Application.packet_rate = 20
SN.node[6].Application.packet_rate = 30
SN.node[7].Application.packet_rate = 30
SN.node[8].Application.packet_rate = 50
SN.node[9].Application.packet_rate = 50
```

The performance of the proposed MAC is examined through two different configurations of the method explained in Sections 5.1.1 and 5.1.2 where the fuzzy algorithm calculates "only the maximum bound" of the backoff window (Section 5.1.1) and also the "maximum and minimum bounds" of the backoff window (Section 5.1.2). Different performance parameters such as reliability, delay, and energy have been put into testing and contrasted against IEEE 802.15.4 standard MAC performance in different scenarios. The performance parameters that we studied to elaborate on the performance of our fuzzy-enabled MAC against IEEE 802.15.4 are:

- Average packets received/Reliability;
  This parameter can also be considered as packet delivery ratio which has been obtained as an average value during our experiments.

- Average end-to-end delay;
  There are different factors influencing the value of this parameter both at PHY and MAC layers which has been discussed in Section 5.1.1.2.

- Energy consumption;
  This parameter has been evaluated both for simulations and also in the experimental set up to study the battery dynamics of real SHIMMER sensor platforms.

5.1.1 Case#1: Dynamic Delayed Maximum Bound for Backoff Interval

The simulation evaluations in this section will discuss the performance of theproposed fuzzy-enabled MAC protocol against IEEE 802.15.4 MAC under our main traffic adaptive method described in Section 4.4. In Section 4.4, a dynamic delayed maximum bound is calculated by the aid of our proposed fuzzy system for the backoff window and no dynamic minimum is considered, which we declared as the main approach to follow in this thesis. The dynamic maximum and minimum bounds for the backoff window explained in Section 4.5 are also evaluated in Section 5.1.2 below, but are not implemented elsewhere in this thesis following the discussion in Section 4.5. Our main focus in proposing reliability-guaranteed medium-

access control protocol is the evaluations of reliability factor in this chapter. The simulations were carried out in regard to the two different techniques of calculating $Channel_{Clear}Rate$ that we described in Sections 4.2.1 and 4.2.2. Unless mentioned otherwise, the simulations in regard to $Channel_{Clear}Rate$ calculations in Section 4.2.1 are referred to as M1 (Method1) and the simulations in regard to $Channel_{Clear}Rate$ calculations in Section 4.2.2 are referred to as M2 (Method2) from here on.

5.1.1.1 Average Packets Received/Reliability

Data delivery ratio can be considered as a measure of reliability the same as the necessary bit error rate (BER), which represents the number of lost packets (Latré, et al., 2011). It is stated in Latré, et al. (2011) that with a low data rate a device can endure a high BER in contrast to higher data rates that can only cope with a low BER. In data communications, *transmission reliability* is treated as a different concept to *having less collision*. While reliability shifts to a more effective way of spreading the retransmission attempts over the next available time slots after a failed transmission, having less collisions is more of an always-present issue in case of a non-scheduled access. In other words, CSMA is considered as a solution to having more reliability in handling the failed attempts while suffering from possible collisions itself (Timmons & Scanlon, 2004). This mainly happens as scheduled-based methods send the data in consecutive time slots. In Chipara et al. (2010) which is an empirical study of a real multi-hop WBAN in a hospital unit, reliability is defined as the average packets received from all the medical sensors at the base station. The base station, though, is not always the next immediate hop to the medical sensor that has sent its data over the channel in their approach. Nevertheless, in a multi-hop implementation of a WBAN such as theirs, reliability is defined closely to how we have measured it in our simulations and experimental works. In the case of a body area network, failures happen mostly when a misconnection occurs due to a deep fade in a channel. Consecutive retrials of transmitting the failed packet, thus most probably, would still result in a failure again. This is because, when the wireless channel undergoes a deep fade, it stays in that condition for at least a few milliseconds, the time period it takes for a transmission and its possible retransmissions to be carried out (Boulis & Tselishchev, 2010). This suggests that the CSMA techniques act more moderately in such deep-fading situations, which is a direct effect of their backoff procedure as also described in Section 1.9.1.The average packets received from all nodes are simulated versus different lengths of simulation runs in Figure 5-1. This is reliability measurements of the 10 sensor nodes with different assigned data rates over a prolonged period of time starting from a duration of 500 seconds long to 5000 seconds long for

each simulation. This is almost two hours of testing the proposed protocol (an hour and 40 minutes). We believe this duration is enough to prove the efficiency of the proposed technique over the IEEE 802.15.4 as any protocol-related problem would have occurred at least once in that time period and we believe a malfunction of the system (whether our protocol or the IEEE 802.15.4) beyond two hours would definitely have a different cause relating to *sensing reliability* of the sensors (problems associated with the sensor's hardware) rather than the *networking reliability* (reliability associated with the protocol's functionality). What is important is the results achieved when comparing the performance of these two MAC protocols during any time period whether short or prolonged. We then evaluated the reliability of our fuzzy-enabled MAC when using both methods of $Channel_{Clear}Rate$ calculations (M1 & M2) against that of IEEE 802.15.4 MAC performance. The two different methods of $Channel_{Clear}Rate$ calculations do not show much difference in the simulation results but, as described in Section 4.2.2, the main inspiration for the second method of $Channel_{Clear}Rate$ calculations (M2) was driven by real implementation of our fuzzy-enabled MAC on real SHIMMER sensor platforms. For this simulation run the following configurations were set in the omnetpp.ini file:

```
[Config allNodesDifferentRate]
SN.node[0].Application.packet_rate = 100

SN.node[1].Application.packet_rate = 5
SN.node[2].Application.packet_rate = 10
SN.node[3].Application.packet_rate = 10
SN.node[4].Application.packet_rate = 20
SN.node[5].Application.packet_rate = 20
SN.node[6].Application.packet_rate = 30
SN.node[7].Application.packet_rate = 30
SN.node[8].Application.packet_rate = 50
SN.node[9].Application.packet_rate = 50
```

As can be seen, we defined a configuration named `allNodesDifferentRate` which, when called in the command line at the time of running a simulation, would assign the written data rate to that particular node. Data rate, as described earlier, is an application-specific parameter that is used in our fuzzy system and is denoted as `Application.packet_rate` in Castalia. We have considered 10 nodes for this simulation scenario. The data rate of the coordinator is set to 100 kbps by default in Castalia. The simulation time varies according to another configuration defined as `VarySimTime` shown as below:

```
[Config VarySimTime]
sim-time-limit =
${SimulationTime=500s,1000s,1500s,2000s,2500s,3000s,3500s,4000s,4500
s, 5000s}
```

It can be seen from Figure 5-1 how reliability has improved using a dynamically adjusted backoff window that would take into account both the node's success history in having access to the channel as well as its individual application-related data rate. Although the data rates assigned to our 10 nodes in the simulation follow a static array as also shown in the `[Config allNodesDifferentRate]` above, we can change the average data of all the nodes in the network to see the effect of higher data rates on the measured reliability. Fuzzy-enabled MAC (M1) denotes the reliability measurements of our fuzzy-enabled MAC when the first method of calculating $Channel_{Clear}Rate$ is used (Section 4.2.1) and Fuzzy-enabled MAC (M2) is referring to the reliability measurements in regard to M2 (Section 4.2.2).

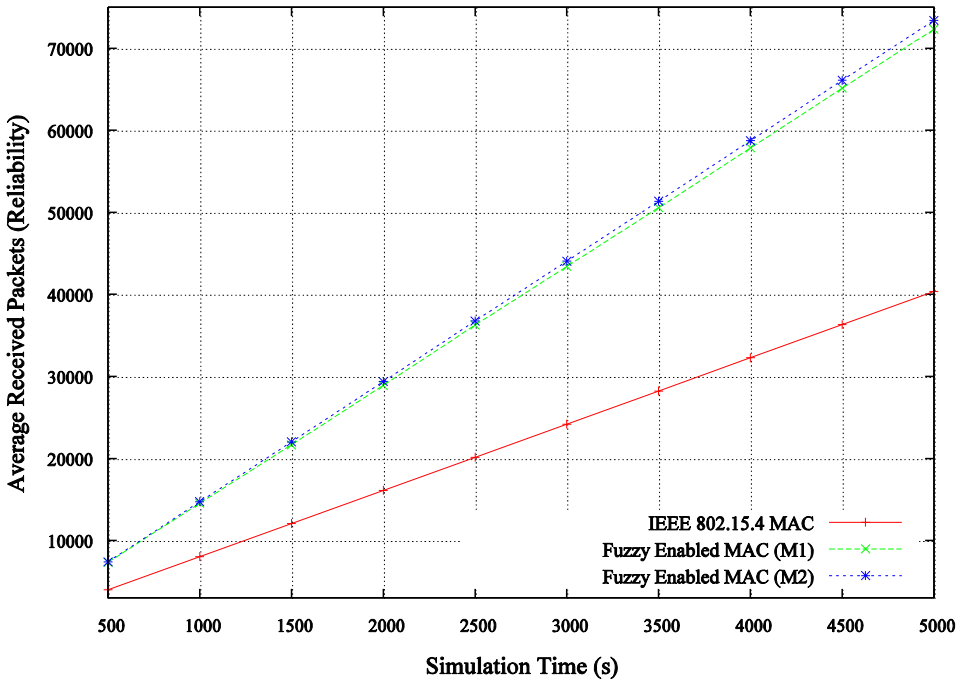


Figure 5-1: Reliability measurements versus time. Fuzzy-enabled MAC (with two methods (M1 & M2) described in Sections 4.2.1&4.2.2) versus IEEE 802.15.4 MAC.

In the next simulation scenario we investigate the effect of increasing the average data rate from low to high. The highest data rate will not exceed the highest data rate assigned in our previous simulation scenario as we seek best achievable reliability for low data-rate applications. The average data rate for the nodes is increased from an average of one kbps to

an average of 52 kbps, which is almost equal to the highest of 50 kbps in the previous scenario. Here is the configuration for when the average data rate for all nodes is one:

```
[Config allNodesDifferentRate]
SN.node[0].Application.packet_rate = 100
SN.node[1].Application.packet_rate = 1
SN.node[2].Application.packet_rate = 1
SN.node[3].Application.packet_rate = 1
SN.node[4].Application.packet_rate = 1
SN.node[5].Application.packet_rate = 1
SN.node[6].Application.packet_rate = 1
SN.node[7].Application.packet_rate = 1
SN.node[8].Application.packet_rate = 1
SN.node[9].Application.packet_rate = 1
```

The above configuration was repeated with different values of data rates to increase the average data rate of all nodes according to Table VII where the S_i array shows the array of data rates assigned to nodes 1 to 9 starting from `SN.node[1].Application.packet_rate` to `SN.node[9].Application.packet_rate[9]`. This has also been shown in Figure 5-2.

Table VII: Increase in the averaged data rate per simulation run (simulation time=1000 seconds)

Nodes' data rate array in each simulation run	The averaged data rate (bits per second)
S_1=[1,1,1,1,1,1,1,1,1]	1
S_2=[1,1,1,1,1,1,1,1,5]	1.4
S_3=[1,1,1,1,1,1,1,5,5]	1.8
S_4=[1,1,1,1,1,1,5,5,10]	2.8
S_5=[1,1,1,1,1,5,5,10,10]	3.8
S_6=[1,1,1,1,5,5,10,10,20]	6
S_7=[1,1,1,5,5,10,10,20,20]	8
S_8=[1,1,5,5,10,10,20,20,30]	11
S_9=[1,5,5,10,10,20,20,30,30]	14
S_{10}=[5,5,10,10,20,20,30,30,40]	17
S_{11}=[5,10,10,20,20,30,30,40,40]	22
S_{12}=[10,10,20,20,30,30,40,40,50]	27
S_{13}=[10,20,20,30,30,40,40,50,50]	32
S_{14}=[20,20,30,30,40,40,50,50,60]	37
S_{15}=[20,30,30,40,40,50,50,60,60]	42
S_{16}=[30,30,40,40,50,50,60,60,70]	52

Figure 5-2: Increase in the average data rate of 10 nodes in the simulation (data rates increase from 1Kbps to 52 Kbps (aggregated)); Simulation time = 1000 seconds

Figure 5-3 shows the effect of higher data rates and it can be seen that, when the network gets too saturated with an average data rate of higher than 50 such as 100 packets per second or over, both protocols reach a steady state in terms of reliability, which is an indication of the efficiency of our proposed MAC algorithm only for an average data rate of 100 and less.However this does not mean that in a particular scenario one single node cannot have a higher data rate than 100 packets per second, since this is the average value made over the array of data rates for all the sensor nodes in the network.It is so likely in medical applications that only one or two nodes tend to have data rates as high as over 100 packets per second, for instance an ECG sensor node, but with kinematic or temperature or blood saturation and the like, the data rates are most likely to be much less. For this evaluation we considered relatively close data-rate values of very low to high data rates for our averagedata rate. The average data rate increase can be read from the averaged data rate in Table VII. The simulation duration for the results in Figure 5-3 is 1000 seconds long. The fact that for aggregated data rates of over 50 kbps the average packets received at the coordinator device does not change any more is because of saturation.

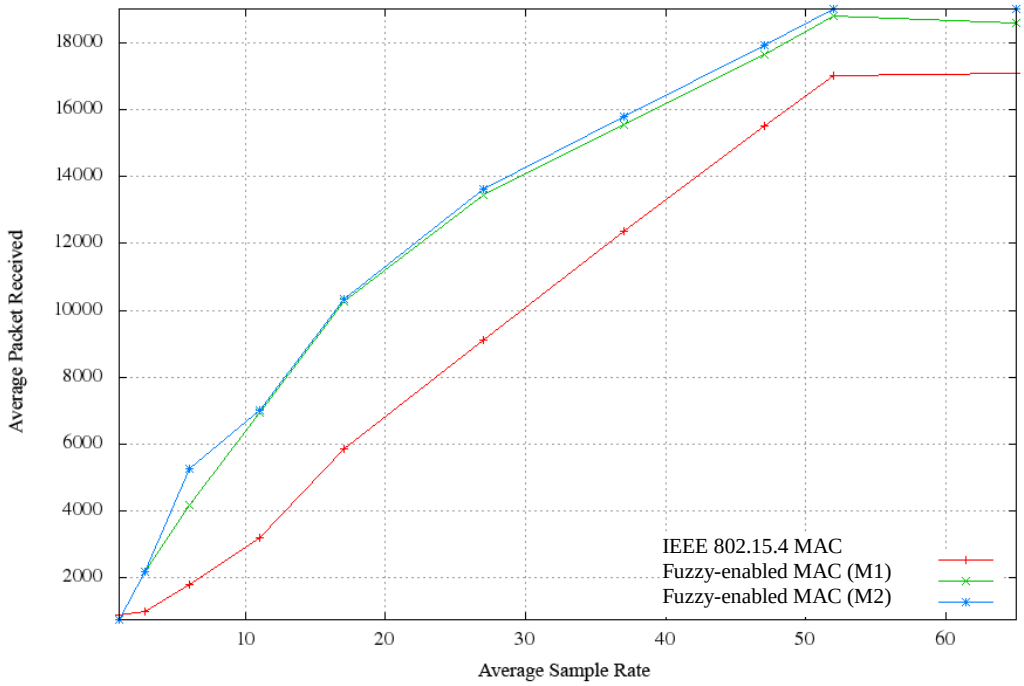


Figure 5-3: Reliability measurements versus average data rate of all nodes. Fuzzy-enabled MAC (with two methods (M1 & M2) described in Sections 4.2.1&4.2.2) versus IEEE 802.15.4 MAC. (Simulation time of 1000 seconds)

In the next scenario we investigate the performance of the proposed fuzzy-enabled MAC protocol when the number of nodes increases. We increase the number of nodes from five to 30. Thirty, of course, is not a practical number for a real implementation of a WBAN in current up-to-date deployments, but we have examined its effect in the simulation environment only. As can be seen from the obtained results in Figure 5-4, the average number of received packets decreases by increasing the number of nodes, which shows a lesser delivery ratio due to having more losses during data transmissions in the network.This could be induced by a higher level of interference and also a higher level of congestion at MAC as a result of contention for the shared medium among sensors to transfer their data. Our fuzzy-enabled MAC exhibits a more successful scheme in balancing the load among the sensors by considering their past successful trials in having access to a channel and also their data rate levels. By manipulating the range of variations for delay time according to $Channel_{Clear}Rate$ for a specific node and its data rate, the protocol gives the chance of data transmission to the other nodes with a less experienced clear channel in their attempts by assigning them a smaller delayvalues during backoffs. The fluctuations for the number of nodes over 20 is quite understandable, as when the number of nodes in a WBAN increases to that extent, the randomness in the nodes' relative positions can

occasionaly put the IEEE 802.15.4 MAC protocol in a better condition in forwarding packets with its original CSMA/CA mechanism. But the trend happens completely by chance and a sudden decrease can be as much anticipated as a sudden increase in its reliability. The trend is therefore not consistent as there is no balancing technique incorporated and a high backoff delay can be as much catastrophic as it can sometimes be moderating. Every simulation scenario reaches its best performance with certain configurations. Here in this example we always reach the best performance with 10 nodes, which most likely happens as a result of the default path loss values defined by Castalia for the assumed topology explained in Table VI. The results would definitely vary with respect to the path loss values for the relative positions of the sensor nodes. But what is most important in this entire discussion is that the general trend of this diagram for when the number of nodes increase is descending.

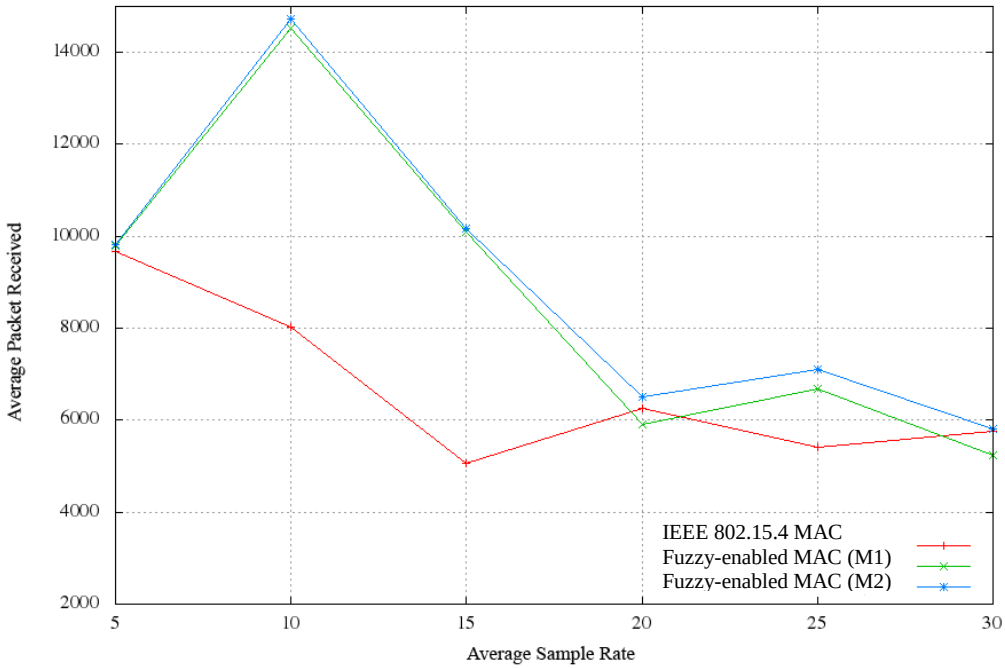


Figure 5-4: Reliability measurements versus number of nodes. Fuzzy-enabled MAC (with two methods (M1 & M2) described in Sections 4.2.1&4.2.2) versus IEEE 802.15.4 MAC (Simulation time of 1000 seconds)

5.1.1.2 Average End-to-end Delay

Latency or delay is usually defined as the time interval between when a packet is ready to be sent from the MAC queue and the time when it is successfully received by the destination node. If a packet gets dropped, the time that has been taken due to its incomplete transmission is not considered in delay calculations (Raptis, Vitsas, & Paparrizos, 2008).The experienced packet latency reflects several factors that have affectedthat packet transmission. The time needed for

a transmission to take place is normally composed of four different types of delay that define the overall delay of a successful packet transmission. Therefore the end-to-end delay for each packet is introduced as follows:

$$d_{end\text{-}to\text{-}end}=d_{trans}+d_{prop}+d_{proc}+d_{cs}; \quad (10)$$

where the $d_{end\text{-}to\text{-}end}$ stands for end-to-end delay, d_{trans} stands for transmission delay, d_{prop} is the propagation delay, d_{proc} is the processing delay and d_{cs} is the carrier sense delay (Ye, Heidemann & Estrin, 2004). The transmission delay depends on the data rate of the wireless link i.e, how many bits per second it is able to forward between the nodes, and also the packet length i.e, how many bits a packet is made up of. The propagation delay is however proportional to the distance between source and destination and it also depends on the propagation speed. The processing delay is the fraction of time needed to process the packet header and its next destination and such (Kurose, Ross, & Ross, 2003). Therefore the packet latency or delay in short can be written as:

$$T_{receive} - T_{ready\text{-}to\text{-}send} \quad (11)$$

Remember that in case of a collision the corrupted packets must be resent, which imposes additional delay. A good scheduling done by a MAC protocol will lead to less probability of collisions, letting nodes contend for the medium when it is more likely available. A good tuning of the backoff window can result in a more realistic backoff time, which imposes less delay in each packet transmission. Four main sources of delay or latency are described in Zhuo et al. (2012) that categorize the produced delay based on the different stages a packet goes through when it travels from the source to the destination: these could be hardware-related or non-hardware related. The buffering delay where the packets are buffered at MAC queue level, for instance, is associated with the protocol performance and no environmental factors play a role in it. The other three identified sources of delay known as forwarding, signal propagation, and receiving and processing delays are directly affected by the physical hardware conditions. The buffering delay in Zhuo et al. (2012) is said to dominate the total packet delay, which reflects the efficiency of resource (bandwidth) allocation to that particular node during the communication especially when the wireless medium is a shared one. Ghildiyal et al. (2011) described the latency requirement for WBAN as “low and predictable”. In Isikman et al. (2011), it is stated that the latency for medical applications should not be more than 125 ms

whereas it can be up to 250 ms for non-medical applications. Zhisheng and Liu (2011) define latency or end-to-end delay as the time it takes for a data packet generated at a slave node to reach its coordinator.

Castalia offers the possibility of depicting the latency as a histogram so that several protocol variations can be compared to each other in terms of the latency window associated with different average values of the number of received packets at the coordinator device. In Figure 5-5(b) for example different latency windows are displayed on the X axis which start with a lowest latency window of [0, 20] mili seconds. Larger average number of received packets are received with this lowest level of latency which can be seen for both versions of our proposed protocol and also the IEEE 802.15.4. As the latency window reaches the higher values such as [120, 140] ms less average number of packets are shown on the Y axis which indicates that not many packets are actually sent with such high latency window during the course of the simulation run (in this case 1000 seconds). Figure 5-5(a) which is achieved for a shorter simulation period (100 seconds) shows less values of average number of packets as expected since less number of packets arrive at the coordinator device from the sensor nodes in the network during 100 seconds rather than 1000 seconds.

As evident from the results in Figures 5-5(a) and 5-5(b), we can see that most of the packets are received with latency under 100 milliseconds, which means that they are transmitted in the first MAC frame after their creation. The X axis shows the amount of latency in milliseconds, for instance [0 20) means the latency window in receiving the number of packets that has been mounted on Y axis. It can be seen from Figure 5-5(a) that the average number of packets received at the coordinator device with a latency of no more than 20 ms with our fuzzy-enabled MAC is around 2500 packets, whilst with the same latency window (between 0 and 20 milliseconds) the IEEE 802.15.4 is only able to transmit 1000 packets compared to 2500 packets with our method. This indicates that the latency regarding individual packets sent to the coordinator is less in our method. Figure 5-5(a) shows how the fuzzy-enabled MAC outperforms the IEEE 802.15.4 as the simulation time goes on. The maximum simulation time for this particular graph is 100 seconds and only with the $Channel_{Clear}Rate$ calculations of M1 (Method1). The smaller portion of packets at the [600,inf) for IEEE 802.15.4 is a sign of potential early saturation compared to our protocol, which signifies a better latency-handling strategy by our traffic-adaptive approach. The number of packets received at the coordinator device with a higher latency of 600 ms and higher than 600 ms is more than the number of

packets in our fuzzy-enabled MAC received within the same latency window. When both data rates of a transmitting node and its $Channel_{Clear}Rate$ are taken into account to discover the next maximum allowed for backoff window, the produced delay reflects the update it gets from the frequently reported real-time conditions of the traffic (which is every 20 super frames in this scenario).

For the latency calculations of a WBAN it is recommended that the temporal variations will be considered at the time of simulations for the wireless channel. This will make the obtained results reflect the real latency performance of a WBAN as much as possible. Temporal variation of a wireless channel can be considered by following the model offered by Castalia, which is a general model.

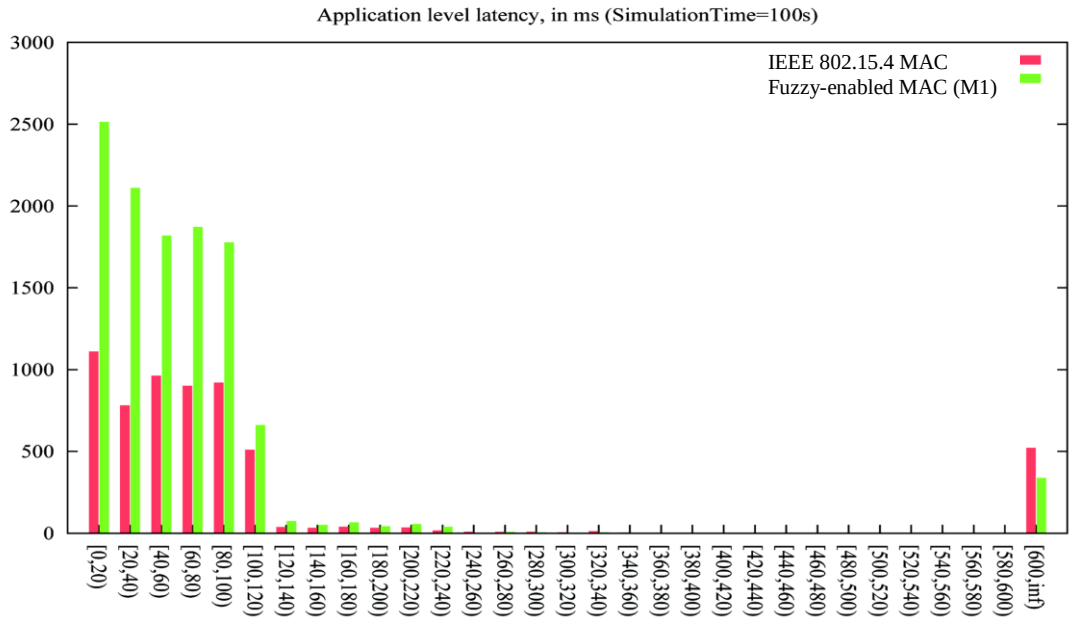


Figure 5-5(a): Average latency of the received packets at coordinator device. Fuzzy-enabled MAC versus IEEE 802.15.4 MAC. (Simulation time of 100 seconds)

This general model is derived based on considering two parameters. The first one is the last known path loss value and the second one is the time that has passed since this value has been observed. By means of these two values a probability density function would be able to draw the new (next) value. In the omnetpp.ini file the directory of the temporal file is given as below:

```
SN.wirelessChannel.pathLossMapFile =
"../Parameters/WirelessChannel/BANmodels/pathLossMap.txt"
```

```
SN.wirelessChannel.temporalModelParametersFile =
"../Parameters/WirelessChannel/BANmodels/TemporalModel.txt"
```

The delay measurements of a longer simulation run are given in Figure 5-5(b), which shows the latency of the fuzzy-enabled MAC in milliseconds for both methods of M1 and M2 in contrast with IEEE 802.15.4. The difference between the average number of packets received at the coordinator device with a latency window of [0, 20) milliseconds for two MAC approaches (fuzzy-enabled MAC & IEEE 802.15.4 MAC) is around 2000 packets. Our fuzzy-enabled MAC manages to transmit an average number of 19000 packets (with both methods of M1 and M2) with a latency of [0, 20) milliseconds compared to an average of 17000 packets of IEEE 802.15.4 MAC. The average number of higher latency packets is adversely less in our method compared to IEEE 802.15.4, which is evident from the bottom right of the figure. IEEE 802.15.4 MAC transmits more number of packets within a latency window of [600, inf) milliseconds, which is relatively high.

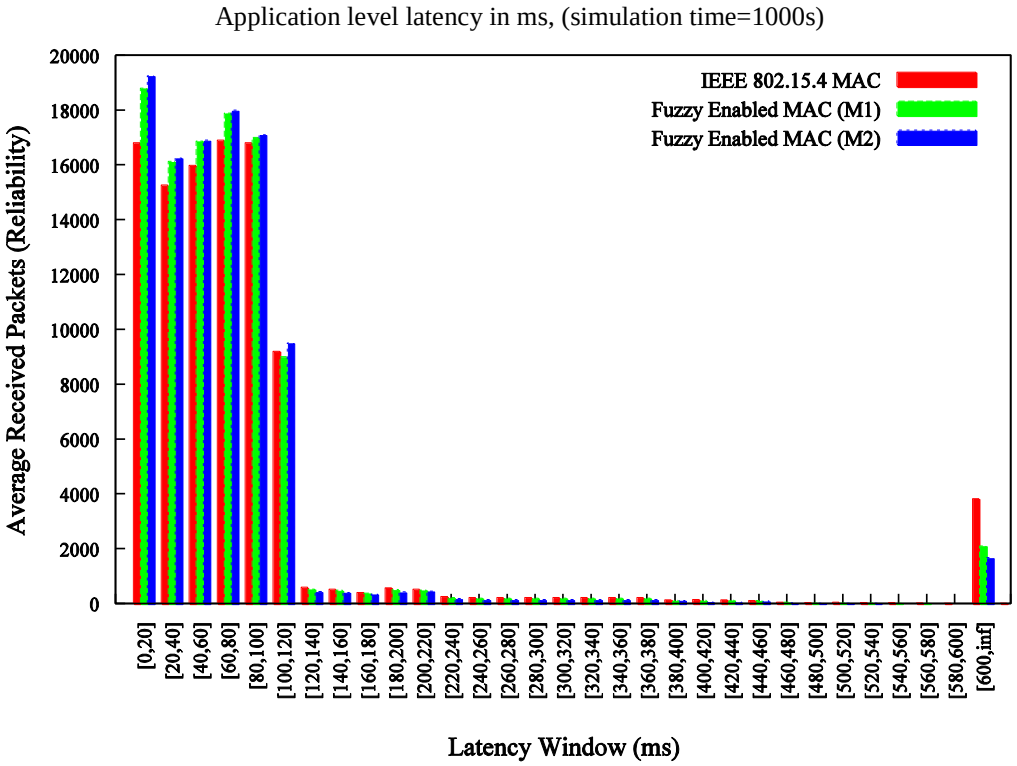


Figure 5-5(b): Average latency of the received packets at coordinator device. Fuzzy-enabled MAC (with two methods (M1 & M2) described in Sections 4.2.1&4.2.2) versus IEEE 802.15.4 MAC. (Simulation time of 1000 seconds)

5.1.1.3 Energy Consumption

Energy is one the main concerns both in WSN and WBAN except that it becomes even more critical when dealing with human lives. If a new algorithm or protocol cannot improve energy consumption, due to its focus on other QoS parameters, it should at least be able to leave it untouched. The normal range for energy consumption in the IEEE 802.15.6, which is specifically devoted to WBANs, is from 0.01 mW for when the device is in its standby mode to 40 mW for when it is operating in its active mode (Isikman et al, 2011). The same energy consumption range in the IEEE 802.15.4 standard is reported to be 25–35 mW. In Ghildiyal et al. (2011), the amount of power consumption is said to be proportional to the data rates that the sensors' radio support, which is hardware oriented. The higher the data rate the less time the node spends in its active or ON mode, which reduces the energy expenditure if the duty cycle is high as well. With lower duty cycles the effect is different i.e. the higher the data rate the more the energy consumption will be as nodes will not be able to keep up with the amount of traffic they receive in their active time through a shared medium and more collisions are likely to happen. Some research papers such as Isikman et al. (2011) have made strong assumptions such as a heart battery as to maximize the lifetime, which is not considered a realistic framework. Not all patients would have heart batteries implanted. The energy analysis in Table VIII demonstrates almost an equal energy consumption level between the two MAC approaches. As evident from the comparison made in Table VIII, the two MAC techniques act similarly in terms of the energy consumed on each sensor node during different simulation lengths. The energy comparison for our prolonged simulation runs, such as the simulations carried out in Figure 5-5(b), followed the exact same pattern, which leaves no concern in terms of energy efficiency of our method in the simulation environment. We have also conducted a feasibility study concerning energy efficiency in Section 5.2, which evaluates the energy spending behavior of a real SHIMMER sensor node when a fuzzy algorithm is implemented in its operating system.

Table VIII: Energy consumption comparison (IEEE 802.15.4 MAC vs. Fuzzy-enabled MAC)

Energy (mJ)		
Time (s)	IEEE 802.15.4 MAC	Fuzzy-enabled MAC
50	0.028	0.027
100	0.054	0.053
150	0.077	0.076
200	0.102	0.101
250	0.128	0.126
300	0.152	0.151
350	0.177	0.176
400	0.202	0.202

5.1.2 Case#2: Dynamic Delayed Maximum and Minimum Bound for Backoff Interval

For this case study a body area network of six nodes (including the coordinator) is considered with node ID numbers indexed from zero to five. Node0 is considered an FFD that acts as the coordinator in the system and is assumed to be carried by the patient. We run the simulations for an array of sample rates that we assign to nodes when setting the main configurations of the network. The assigned sample rates are: 10, 20, 20, 100, and 60 kbps, which are devoted to nodes 1–5 respectively (excluding Node0 as the coordinator). The guaranteed time slot (GTS) capability of the protocol is ON, which means we take advantage of both contention-access and contention-free periods in the super frame structure. The maximum of the packet retries in the event of a packet loss is three times, which is predefined in the standard but is a configurable metric. The maximum length for the MAC queue buffer is assumed to be 32 packets. The five transmitting nodes are assumed on Left/Right wrist, Left/Right ankle, and the chest, whilst the coordinator is placed on the right hip of the patient's body. For the simulations of this section we have only considered Method1 (M1) of the fuzzy system, which produces only a maximum bound for the backoff window. The simulation duration is about 900 seconds. The path loss model introduced in Table V has been used.

The simulations carried out in Section 5.1.2 are totally different from those of the previous section and that is because in Section 5.1.2 we are running our proposed fuzzy system on both ends of the backoff interval. In this scenario both the upper and lower bounds of the backoff window are getting calculated whilst in Section 5.1.1 only the upper bound has been investigated. Therefore we did not compare these two cases with same simulation configurations as they are two different fuzzy-enabled MAC implementations. Testing of the

fuzzy-enabled MAC when the fuzzy algorithm runs to calculate both ends of the backoff window has been brought in this section to examine its suitability for more specific applications with a possibility of very low data rates where extremely slight changes to the backoff window would also impact the latency and reliability performance. However in this thesis we only study the design of a fuzzy-enabled MAC for low to medium data rate applications and therefore except in Section 5.1.2, this version of the fuzzy-enabled MAC protocol has not been investigated."

5.1.2.1 Average Packets Received/Reliability

Figure 5-6 shows the average number of received packets by the five nodes for different simulation runs. We increased the duration of each simulation by 50 seconds each time to examine the impact of a prolonged monitoring application on the amount of successfully received data at the coordinator, which is what we perceive as reliability. It can be seen from the figure that the fuzzy-enabled MAC approach increases the ratio at which the packets are received successfully.

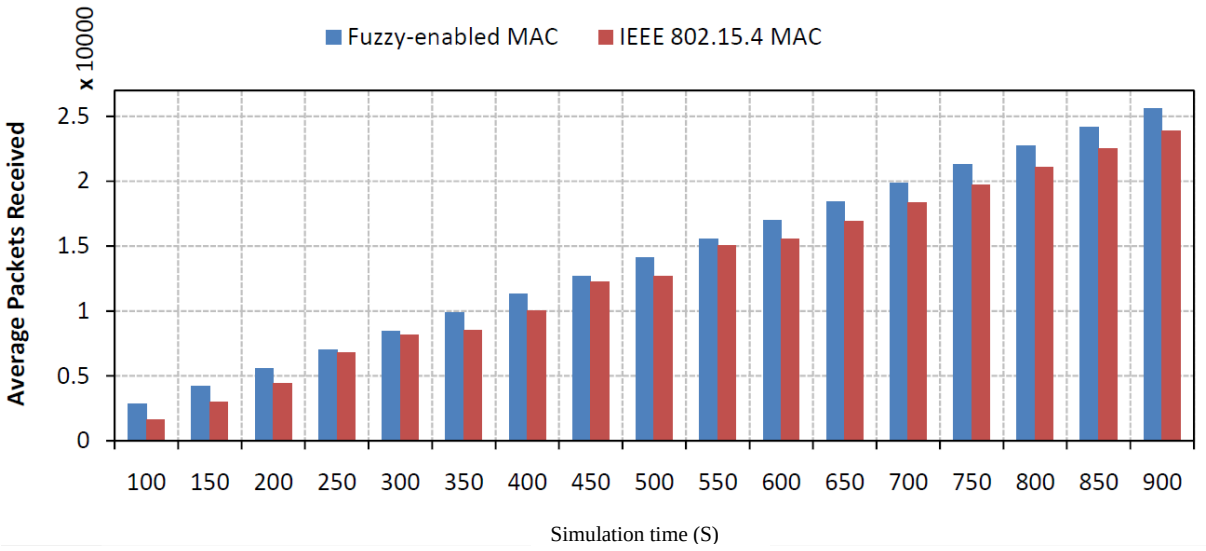


Figure 5-6: Average packets received at coordinator versus simulation time in seconds. Fuzzy-enabled MAC versus IEEE 802.15.4 MAC. (Method1 is used for *$Channel_{Clear}Rate$* calculations).

This however is expected to improve more if all the previous calculations of $Channel_{Clear}Rate$ are taken into account. However, this will increase the computation overhead. Figure 5-7 shows the increase rate in successfully received packets to show the difference in the number of packets received in a smaller scale. It is obvious that as the simulation time increases more

packets are received by both protocols. Transmitting at low data rates does not harm data fidelity or the reliability of data in a wireless body area network. The effect of higher data rates was tested for both MAC protocols to examine the efficiency of each in sending more successful packets to their coordinator (Figure 5-8). Since the proposed fuzzy-enabled MAC protocol was configured with a static set of data rates (earlier described in this section), we changed the set of data rates randomly but every time with a 5% increase in the average of all the data rates assigned to nodes. Data rates are in kilobits per second.

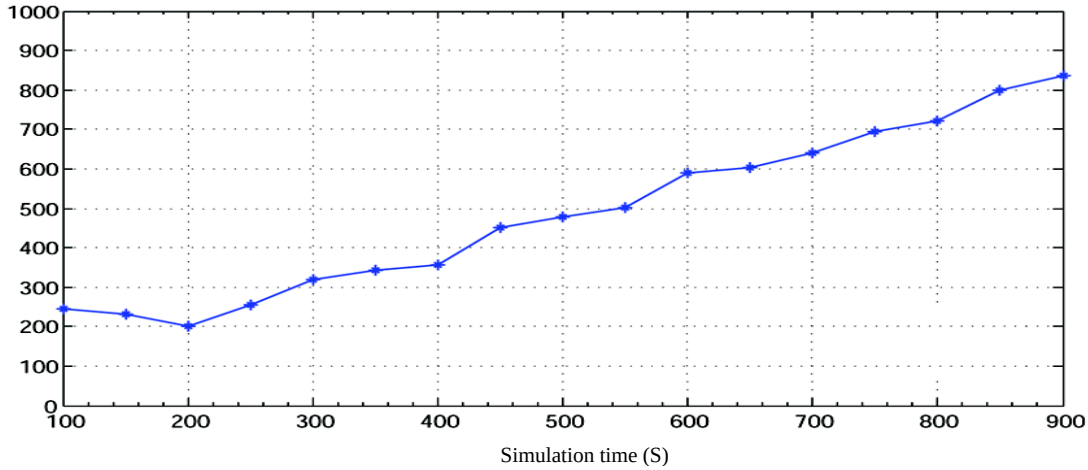


Figure 5-7: Increase in the number of received packets versus simulation time in seconds. (Fuzzy-enabled MAC using Method1 for *Channel*$_{Clear}$*Rate* calculations)

It can be seen that a dynamically adjusted backoff interval reflecting the number of each node' past successful channel access and application layer data rate can yield an improved delivery ratio compared to the IEEE 802.15.4 MAC.

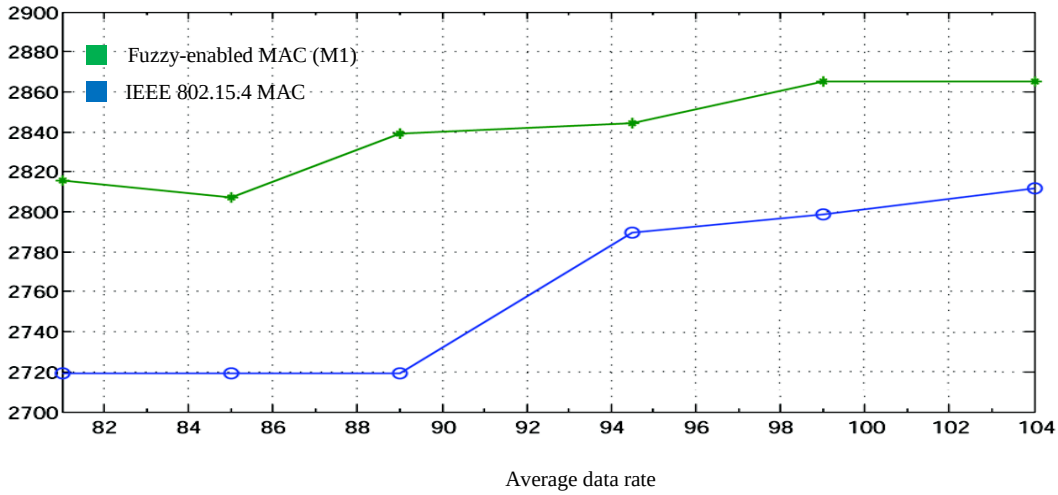


Figure 5-8 Average packets received at coordinator versus average data rate. 5% increase on the averaged data rate in each simulation run. Fuzzy-enabled MAC performance against IEEE 802.15.4 MAC. (Method1 is used for $Channel_{Clear}Rate$ calculations, simulation time=100 seconds)

A brief introduction to setting required configurations for the simulation experiments was given in this section (Section 5.1). Different simulation scenarios were then created using those configurations and the fuzzy-enabled MAC was tested against IEEE 802.15.4 under the defined scenarios in Sections 5.1.1 and 5.1.2. Section 5.1.1 evaluates the reliability of the fuzzy-enabled MAC when only the maximum bound of the backoff window is dynamically adjusted by the fuzzy algorithm. This was earlier discussed in detail in Section 4.4. Section 5.1.2, though, evaluates the reliability metric of the fuzzy-enabled MAC when both the minimum and maximum bounds of the backoff window are dynamically adjusted by the fuzzy algorithm. This was earlier discussed in Section 4.5. With reference to the discussions in Section 4.5, which declare the dynamically adjusted maximum bound of the backoff window to be the main approach of our proposed MAC, we only justified the reliability measure in Section 5.1.2.

5.2 Realistic Energy Measurement Challenges

Besides reliability, energy is also one of the main concerns in any WBAN implementation, especially when it comes to healthcare applications. In many of the practical research works discussed in Section 1.3, the number of days a typical medical sensor would perform before its battery goes flat is reported to be 3–4 days. The battery lifetime, however, is not a constant value and it largely depends on the applications the sensor runs and the data rates at which it transmits its data. In Ko et al. (2008), for instance, the ECG sensor draws 20.4 mA at 3.3 mV on average whilst the pulse oximeter and the LCD of the sensor draw 110 mA at 3.7 mV. The pulse oximeter itself draws 64.16 mA at 3.7 mV. Hardware configurations such as antenna gain

could play a role as well. Even environmental factors such as temperature and humidity would have an impact on how efficiently the power supply of a battery unit can be consumed. Maintaining a good level of energy throughout the monitoring application in a medical set up plays an important role in the achieved reliability. In Bonnici et al. (2012), the empirical study over four different types of sensor platforms from different manufacturers proves a better reliability result for the sensor platform with the largest battery capacity over the others. The overall improvement in the system's reliability is a direct result of better sensing reliability earlier discussed in Chipara et al. (2009) in Section 1.3, which can be achieved by a better battery performance. Some research works exploit different techniques to improve the power supply of their medical sensor nodes. In Ko et al. (2008), for instance, radio duty cycling was used to increase the battery lifetime of the ECG sensors we mentioned before up to six fold (down to 3.42 mA) compared to the original battery lifetime of the sensor with all its components at active mode. However, duty cycling the radio always depends on the data rate of the specific application the sensor runs and does not always have a good effect on latency or reliability measurements of the network. Duty cycling may enhance the lifetime of the battery unit in a sensor platform by putting it into periodic sleep intervals but it can produce more latency and sometimes less throughput, which could both affect the reliability which is not mostly desired in WBAN implementations.

Hurni and Braun (2010) show how monitoring applications such as those specifically for healthcare cannot tolerate MAC implementations with tradeoffs on their quality of service parameters such as latency. Such applications differ from traditional WSN monitoring applications in their requirements for a reasonable quality of service performance when there is an urge to transmit the data as fast as possible. Therefore energy saving is not always the first priority in every monitoring application but should be considered to the extent that is possible.In our research, for example, the first priority parameter to effectively achieve is reliability whilst we also keep in mind the requirement for the energy performance of the system not to drop behind its original performance. We therefore had to find a way to observe the battery-spending behavior of the sensor nodes (in our case, SHIMMER sensor platforms) to ensure the efficiency of our fuzzy algorithm when it runs on resource-restricted sensor nodes. Measuring the real-time energy consumption of nodes as the real monitoring application goes on is a huge step forward compared to energy calculations in the simulation environment. Actual considerations such as battery collapse lead to uncertainty in determining how much energy has really been spent for the application running on the sensor's CPU. This normally

happens when the battery capacity drops to 80%, which is only 20% down of the maximum capacity (Battery University, 2011, The secrets of battery runtime). Some researchers have tried to control this effect by limiting the peak power, as in Wong et al. (2009) where the peak power is set to only 3mA.In order to be able to compare the energy-spending behavior of a SHIMMER sensor platform running our fuzzy algorithm we must first be able to implement the fuzzy algorithm on the sensor platforms. Section 5.2.2 describes such an implementation of a light fuzzy library into the TinyOS, which is the SHIMMER sensor's operating system. We then move to Section 5.2.2.1 where we have done battery behavior monitoring of these sensor platforms at the application layer to elucidate the actual amount of energy that has been spent for just running the added fuzzy algorithm. The next section touches on some general information and a brief introduction to the specifications of the SHIMMER sensor platforms that we used for our experiments.

5.2.1 **SHIMMER Sensor Platform**

The term SHIMMER^{TM} is the trade mark and name registered for the sensors by its group which would represent Sensing Health with Intelligence, Modularity, Mobility, and Experimental Reusability. In this section we will examine at the sensor platforms we used for the experimental stages of our research. SHIMMER sensor platforms are basically designed for wearable monitoring activities, which could range from sport and entertainment to athletic improvement, ambient sensing solutions, and healthcare applications. We purchased a SHIMMER platinum development kit from SHIMMER research group (SHIMMER Sensing Technology, 2014) which is currently (January 2014) available in the market for UK €2,749.00 and offers support for the development of Biophysical, Kinematic (IMU), or Ambient sensing applications. Figure 5-9 shows the kit with all its included equipment and peripherals. The kit includes the following items:

- 6 x Shimmer units
- 1 x ECG Daughter Board with biophysical top enclosure
- 1 x EMG Daughter Board with biophysical top enclosure
- 1 x GSR Daughter Board with biophysical top enclosure
- 1 x Strain Gauge Daughter Board with Biophysical top enclosure and 3.5mm Jack
- 1 x 9DoF Daughter Board with kinematics top enclosure
- 1 x GPS Daughter Board with top enclosure
- 1 x AnEx Board
- 1 x Span module

- 2 x USB Readers / Programming Docks with USB leads
- 1 x 6 Gang Multi Charger with power supply and region specific plug
- 6 x 2GB microSD cards with adapter
- 9 x ECG/EMG leads
- 1 x Live Distribution USB key
- 1 x Shimmer Manual
- 2 x Standard Straps (1 Arm/Ankle, 1 Waist/Chest)
- 1 x Kinematic Straps (1 Arm/Ankle)
- 3 x Biophysical Straps (3 Arm/Ankle, 1 Waist/Chest)

Figure 5-9: SHIMMER Platinum Development Kit

The sensors' radio supports both IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 communication technologies. The sensing modules of the platforms include Accelerometer, Gyro, Magnetometer, ECG, EMG, GSR, Strain Gauge, GPS, Temperature, and Barometric Pressure. It has local memory storage of 2GB microSD card for data logging applications. In terms of compatibility, they easily integrate and interact with the existing systems and technologies. Examination of the fact sheets of the ECG, EMG, and GSR daughter boards that have been used for the experimental stages of this research, it can be seen that they all share the same set of specifications, which are gathered in Table IX (SHIMMER Sensing Technology, 2014).

Table IX: SHIMMER Unit Specifications

Processing	MSP 430 Microcontroller (8mHz, 16 bit)
Communication	Bluetooth- RN42, 802.15.4 radio- CC2420
Storage	Integrated 2GB MicroSD Card Slot
Battery	450 mAh rechargeable Li-ion
Integrated 3 Axis accel	Freescale MMA7361
Accel Range	+/- 1.5g – 6g

These daughter boards provide different sensing capabilities for different application types for a real WBAN implementation. The ECG (Electrocardiogram) daughter board with three leads, for instance, is to record pathways of electrical impulses of the heart muscle activity. It converts the ECG signals into digital format and streams it live to a gateway device that we call coordinator in this thesis. Its gathered ECG data can also be combined with the accelerometer readings for more precise data. EMG (Electromyogram) has similar capabilities as those described for ECG except that it is for muscle activity. Such activities are listed as muscle contractions, assessing nerve conduction, muscle response, etc. as discussed in the Shimmer Research Group (2014) EMG Sensor Module Specification Fact Sheet. GSR (Galvanic Skin Response) is used to assess the level of sweat sensed from the skin of a person. The two electrodes that it has will be attached to two of the person's fingers and will determine a higher level sweat when the conductivity between the two electrodes increases. These are the explanations of such sensors when they are used in a real WBAN implementation. However, we did some energy measurements with these SHIMMER platforms just to assess the energy efficiency of the fuzzy algorithm to be performed on these platforms to validate its feasibility for future implementation. The user interface that we used as the application development tool was the Shimmer Connect (Shimmer Research Group, 2014, Shimmer Connect Rev 0.10a User Manual). In addition to that, the Shimmer 2r LabVIEW (Shimmer Research Group, 2014, Shimmer Sensing LabVIEW Instrument Driver Library User Manual Revision 2.1a) provides a library of some predesigned and helpful small applications that are specific to SHIMMER platforms, along with their source codes. In the following sections, the method we used to monitor the energy behavior of SHIMMER sensor platforms is explained in detail.

5.2.2 **Developing an Energy-efficient Fuzzy Engine on TinyOS for SHIMMER**

The goal of this section is to develop an energy and computational-efficient fuzzy-logic engine (called MFE: Micro Fuzzy Engine) in TinyOS operating system ("TinyOS, an Operating System Designed for Low Power Wireless Devices", 2013) with nesC language (Gay et al., 2003) in order to do the energy measurements of the proposed MAC protocol as close to reality as possible. After first implementing a fuzzy system into the operating system of the SHIMMER sensor platforms, we conduct a feasibility study of running a sample fuzzy system such as described in Section 5.2.2.1 to measure up the energy spending by the SHIMMER sensor platforms in different scenarios. The implemented fuzzy system in nesC is specifically designed for embedded systems. For the inference engine, we used the fuzzy inference engine implemented in C by Viot (1993), and we modified and added some extra features to be able to run it on TinyOS.

Figure 5-10 illustrates the MFE data structure. The '*fuzzyIoType'* structure is used to define the fuzzy inputs and output. The '*fuzzyIoType'* structure contains a membership function pointer, and the next input or output pointer. The membership functions are defined in "*fuzzyMfType*" structure, which contains two X axis points and two slope values that describe a triangular or trapezoidal shape membership function. This data can be used to calculate the degrees of membership for each input. The resulting antecedent value is stored in the "*value*" field of the membership function structure. Rules can be represented by two sets of pointers in "*fuzzyRuleType*" structure. The first set indicates an "IF" statement, and the second set points to a "THEN" statement. In "*fuzzyRuleType*"the "*ruleOperation*" parameter defines the operation type between the two "IF" statements: "AND" or "OR". Finally a pointer similar to the *fuzzyIoType* structure handles the output.

This implementation of this fuzzy logic engine is very light and designed for embedded systems as declared by Viot (1993), who also declares that fuzzy approaches generally require much less memory and computing power than conventional methods. The following pseudo-code briefs the set of functions that utilizes the data structure that we defined above. This code was implemented in C language, based on the fuzzy engine implementation developed by Viot (1993).

The pseudo-code also includes the simplifying technique explained later in Section 5.2.2.2 where an array is used to keep account of the data regarding a previously calculated *FuzzyMaxDelay* value. This part of the code has later been used in the CSMA/CA mechanism of the SHIMMER sensor node when implementing the fuzzy system described in Chapter 6 and the pseudo-code is noted in Section 5.2.2.2 with real inputs of our fuzzy-enabled MAC.

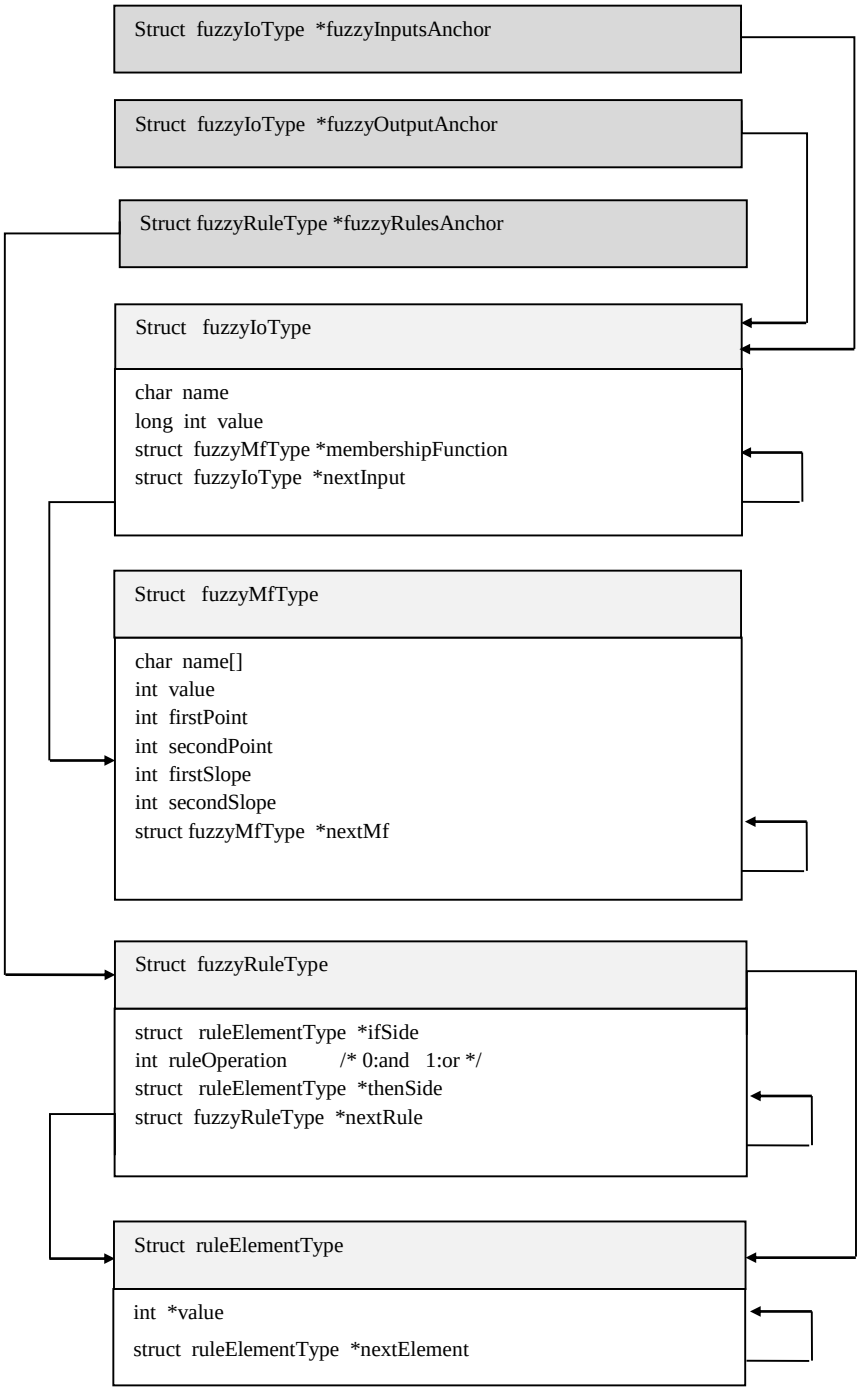


Figure 5-10: Fuzzy Engine Structure in C Language

Pseudo-code for micro fuzzy-engine implementation in C (written for nesC)

```
void resetFuzzy()
  for all system_Inputs
    for all membership_functions in system_input
      membership_functions_input->value = 0;
  system_Output->value=0;
  for all membership_functions in system_output
    membership_functions_output->value = 0;
```

```
void initializeFuzzy(int initializeFuzzyOutputCachingArrayFlag)
  initializeFuzzyOutputCachingArray();

  # Initialize inputs and outputs
  for all fuzzy inputs and outputs as item
    ioptr = Alloc memory with the size of input or output type struct
    # Define Anchor to top of inputs or outputs
    if item is input
      System_Inputs[] = ioptr;
    else
      System_Outputs[] = ioptr;
    for all membership functions in inputs or outputs
      if mfptr == NULL
        mfptr = Alloc memory with the size of membership function type
struct
        ioptr->membership_functions = mfptr;
      else
        mfptr->next = Alloc memory with the size of membership function
type struct
        mfptr = mfptr->next;
      mfptr->value = 0;
      mfptr->leftPoint = X1;
      mfptr->rightPoint = X4;
      mfptr->leftSlope = UPPER_LIMIT / (X2 - X1);
      mfptr->rightSlope = UPPER_LIMIT / (X4 - X3);

  # Initialise rules
  ruleptr =  Alloc memory with the size of rule type struct
  # Anchor to rule base
  Rule_Base = ruleptr;
  for all rules
    A = Get first rule element
    X = Get fuzzy proposition relationship
    B = Get second rule element
    Y = Then element

    ioptr = System_Inputs;
    # Initialize A, B elements
    for all System_Inputs[1 and 2] as mfptr
      if mfptr->name == A or B
        ifptr = Alloc memory with the size of rule element type struct
        ifptr->value = mfptr->value;
        ifptr->next = Alloc memory with the size of rule element type
struct
        ifptr = ifptr->next;

    # Initialize X element
    ruleptr->fuzzyPropositionRelationship = X;

    # Initialize Y element
```

```
    for all System_Output[1] as mfptr
      if mfptr->name == Y
        thenptr = Alloc memory with the size of rule element then type
struct
        ruleptr->then_side = thenptr;
        thenptr->value = mfptr->value;

    ruleptr->next= Alloc memory with the size of rule element type struct
    ruleptr = ruleptr->next;
    ruleptr->next = NULL;
```

```
void setFuzzyInputs(firstInput, secondInput)
  ioptr = System_Inputs;
  ioptr->value = firstInput;
  ioptr = ioptr->next;
  ioptr->value = secondInput;
```

```
void fuzzification()
  for all System_Inputs as input
    for all input membership functions as membershipFunction
      computeDegreeOfMembership(membershipFunction,input->value);
```

```
void computeDegreeOfMembership(membershipFunction, input)
   delta_1 = input - membershipFunction->leftPoint;
   delta_2 = membershipFunction->rightPoint - input;
   if delta_1 <= 0 || delta_2 <= 0
      mf->value = 0;
   else
      mf->value = min ((membershipFunction->slope1 * delta_1),
(membershipFunction->slope2 * delta_2));
      mf->value = min (membershipFunction->value, UPPER_LIMIT);
```

```
void ruleEvaluation()
  for all rules in Rule_Base
    strength=UPPER_LIMIT;
    if rule->fuzzyPropositionRelationship == AND
      for all rule if side as ifSide
        strength = min(strength, ifSide->value);
    else if rule->fuzzyPropositionRelationship == OR
      for all rule if side as ifSide
        strength = max(strength, ifSide->value);
  for all rule then side as thenSide
    thenSide->value = max(strength, thenSide->value);
```

```
void defuzzification()
  for all System_Output as systemOutput
    sum_of_products = 0;
    sum_of_areas = 0;
    for all systemOutput membership functions as membershipFunction
      area = computeAreaOfTrapezoid(membershipFunction);
      centroid = membershipFunction->leftPoint + (membershipFunction-
>rightPoint - membershipFunction->leftPoint) / 2;
      sumCentroid = sumCentroid + centroid;
      sum_of_products += area * centroid;
      sum_of_areas += area;
```

```
    if sum_of_areas == 0
      systemOutput->value = 0;
    else
      systemOutput->value = sum_of_products / sum_of_areas;
```

```
float computeAreaOfTrapezoid(membershipFunction)
  base = membershipFunction->rightPoint - membershipFunction->leftPoint;
  run_1 = membershipFunction->value / membershipFunction->leftSlope;
  run_2 = membershipFunction->value / membershipFunction->rightSlope;
  top = base - run_1 - run_2;
  area = membershipFunction->value * (base+top) / 2;
  return area;
```

```
int tinyFuzzy(input1, input2)
  setFuzzyInputs(input1,input2);
  fuzzification();
  ruleEvaluation();
  defuzzification();
  fuzzyOutput = getFuzzyOutputs();
  resetFuzzy();
  return fuzzyOutput;
```

```
int optimizeTinyFuzzy(input1, input2)
  if fuzzyOutputCachingArray[input1][input2] == NULL
    fuzzyOutput = tinyFuzzy(input1,input2);
    fuzzyOutputCachingArray[input1][input2] = fuzzyOutput;
    resetFuzzy();
  else
    fuzzyOutput = fuzzyOutputCachingArray[input1][input2];
  return fuzzyOutput;
```

5.2.2.1 Fuzzy Engine Computational Energy Measurements

Using a fuzzy control system on resource-limited sensor devices will lead to a huge waste of the sensors' battery energy and microcontroller capacity if no method of optimizing the fuzzy system performance is incorporated. Therefore the fuzzy system that we propose in this thesis, if ever implemented on real SHIMMER sensor platforms, needs to take advantage of some simplifying techniques that are described in Section 5.2.2.2. But first we must investigate the energy-efficiency level that we obtain from running a fuzzy system on a SHIMMER sensor platform. In order to demonstrate the effect of a fuzzy system, we carried out an experiment on a real SHIMMER sensor platform to understand how much extra energy would be imposed on the system just from the fuzzy engine itself. For real-time energy monitoring (battery discharge behavior) we decided to run an application written for SHIMMER sensor nodes. The `SimpleAccelAppC` application uses Bluetooth to stream three accelerometer channels and the battery usage in a parameter called "`batt`". We made slight changes to the code to measure

only the battery usage on the go and not the accelerometer readings. It does not actually matter what communication standard (IEEE 802.15.1 or IEEE 802.15.4) is being used for this experiment as the goal is only to measure the computational complexity of the fuzzy system in terms of the extra energy imposed when using different implementations and input parameters. One point to remember when doing any real energy measurements is to make sure the battery of the SHIMMER sensors tested are fully charged, as partially charged batteries will not show the exact flow of the battery's discharging behavior. This application (the modified `SimpleAccelAppC`) will help us to plot the energy-spending behavior of a SHIMMER sensor node as it runs a specific application such as a simple fuzzy algorithm that we will explain next. As mentioned earlier, this simple implementation of a fuzzy system is not at MAC layer yet as we only attempt to calculate the difference in the energy spent and therefore we are running this fuzzy algorithm at the application layer of the SHIMMER sensor node. The simple fuzzy system we created in the TinyOS and at node's application layer comprises two inputs and one output to imitate the characteristics of the fuzzy algorithm we designed for MAC layer just so we can measure the energy spent merely for the fuzzy system to run. The pseudo-code of the two-input-one-output fuzzy system implementation is discussed here, which is implemented based on the fuzzy implementation described in Section 5.2.2.

The designed fuzzy system is defined with randomized inputs and outputs that are within the same ranges (as in MAC) every time a packet is sent. The inputs of the fuzzy system for this testing are to be randomly generated in their predefined value ranges (discussed in Chapter 4). By running different iterations of the fuzzy algorithm before every packet transmission in a long-term real experimentation run (over 18 hours for each specified number of iterations), we looked into the possibility of running our fuzzy algorithm on real sensor nodes at MAC layer by studying the energy consumption behavior. We measured the energy during a specified period of time both with and without the fuzzy algorithm described above and drew the usage graph for both cases to see the exact difference in terms of energy. The difference between these two graphs is, of course, the energy that has been merely spent on running the fuzzy algorithm. The battery level changes with respect to the received number of packets and is shown in mV for iterations of up to 100 times of the implemented fuzzy algorithm. The battery voltage level is sketched in Figure 5-11. Each stroke is defined in the figure's label by the number of iterations the fuzzy system had run before sending each packet. `Fuzzy iteration:0`, for instance, means that the designed fuzzy system explained earlier in this

section does not run at all, which represents the packets being sent to their coordinator under a normal Bluetooth connection. The other strokes specify a minimum of 20 iterations to a maximum of 100 iterations per transmitted packet.

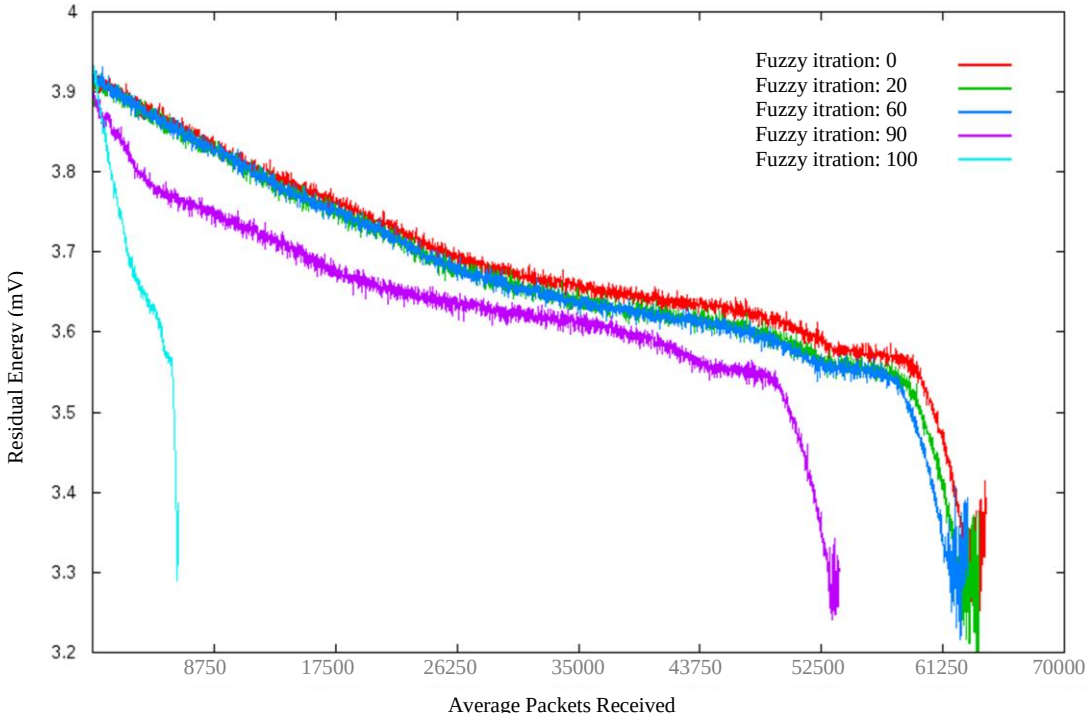


Figure 5-11: Residual battery voltage versus average packets received (over 18 hours); with different iterations of the fuzzy system

When nodes are fully charged, they have a supply of up to 3.9 mV, which is shown on the Y axis. The X axis shows the packets received for the specific node during the entire battery lifetime of the sensor. The red stroke is the battery-discharging behavior when there is no fuzzy algorithm running on the SHIMMER sensor node, which is actually the normal energy consumption for a SHIMMER sensor node when transmitting to the coordinator over the experimentation time. It can be seen from Figure 5-11 that when the fuzzy engine runs at 90 iterations before each packet is sent at the application layer it will deplete the battery faster and with less numbers of packets received. It will stop its transmission at approximately a maximum of 5k packets received, which is a difference of over 55k packets to the case of no iterations of fuzzy algorithm, highlighted in red. Iterations of 20 and 60, though, show almost the same behavior as no fuzzy system running on a sensor platform. In this experiment we have really challenged the energy efficiency of our fuzzy algorithm that we aim to incorporate into the MAC sub layer of IEEE 802.15.4 for a better backoff performance. The imposed

computational complexity of running over 20 iterations of a fuzzy algorithm before each packet is sent over the channel is relatively aggressive and barely happens at the MAC layer. This therefore gives hope to implementing our fuzzy system into the CSMA/CA, ensuring decent energy behavior. In the next sub-section we see how we have tried to address this slight energy efficiency difference by a history-based technique to guarantee an energy efficient behavior for our fuzzy-enabled MAC in real experimental implementations. Figure 5-12 shows a clearer vision of the decrease in the number of received packets for a SHIMMER sensor node when more iterations of a fuzzy algorithm are experienced. It is shown how the number of received packets drops suddenly if an iteration of 100 runs of the fuzzy system is performed before each packet is sent.

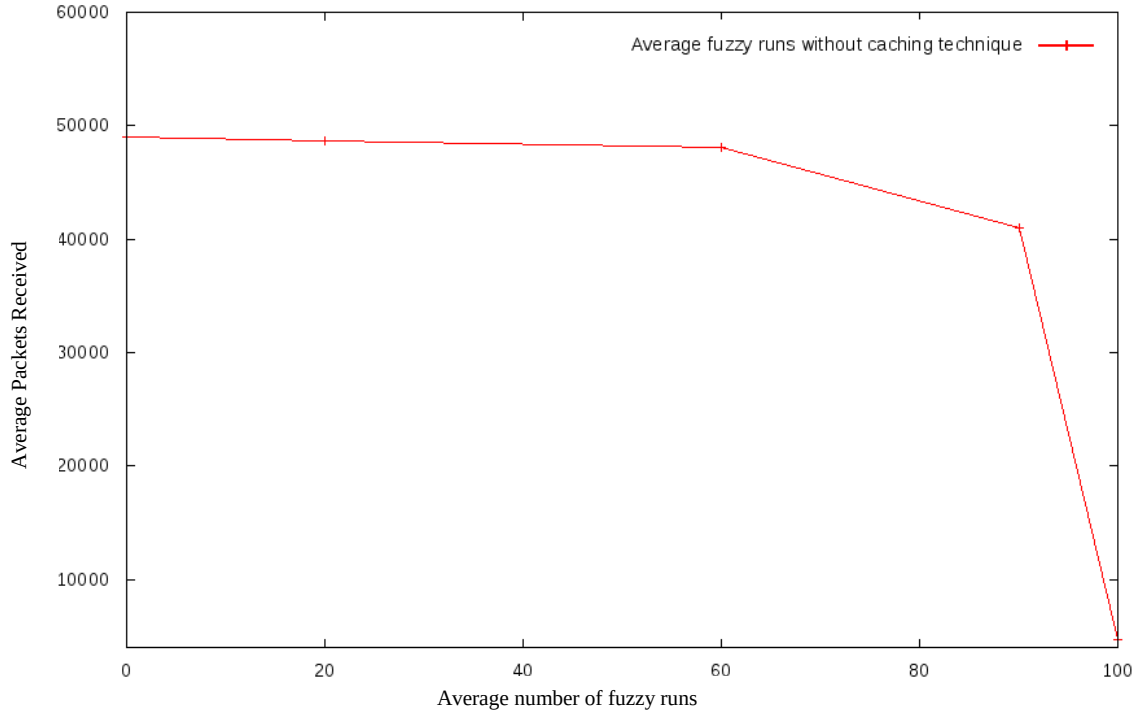


Figure 5-12: Average packets received versus fuzzy iterations per transmitted packet (4000 seconds)

The interesting point about the battery lifetime of this sensor node is more visible in Figure 5-13. Figure 5-13 demonstrates the time taken for the fully charged battery (at 3.9 mV) of each SHIMMER sensor node to deplete completely when different iterations of the fuzzy engine run on them. There is a slight difference between the battery lifetime of the sensor running 90 iterations of fuzzy and the one running 100 iterations. This perhaps shows a saturated state in which the sensor node is not capable of packet transmission any more when it has to run 100 iterations of fuzzy engine before each packet transmission. Therefore the transmission may stop completely and the lifetime of the battery improves a little bit—only a matter of few

minutes. In this case, the depletion time with 90 iterations compared to 100 iterations is only 186 seconds of difference, which is only 3 minutes.

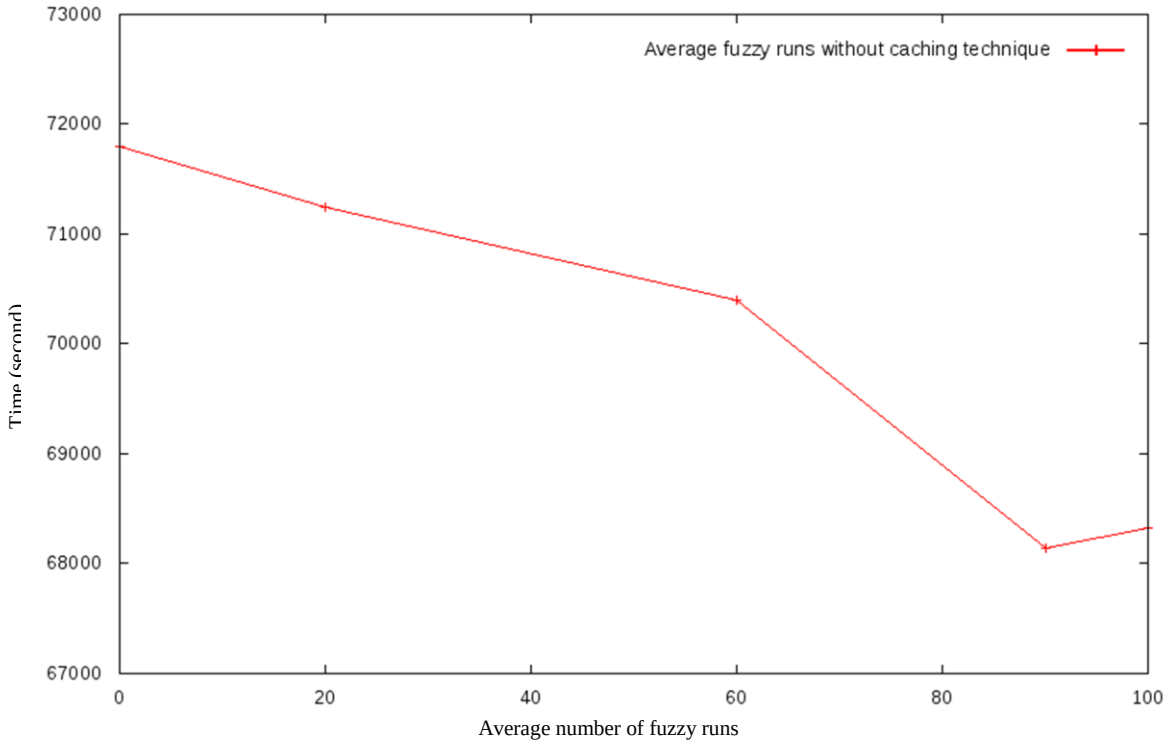


Figure 5-13: Time (seconds) versus fuzzy engine iterations

5.2.2.2 Caching-enabled MFE: Optimizing Fuzzy Algorithm for Real Sensors

We saw in the previous section how running a sample fuzzy system (closely implemented to our fuzzy algorithm at a MAC layer) would have an impact on battery discharge behavior. The effect as shown in the discussed experimental results for iterations of 20 and 60 was very slight but must be mitigated as much as possible. In order to reduce the complexity of the fuzzy algorithm running on each sensor node, we added a caching capability to our fuzzy approach, which was mentioned in Section 5.2.2. This caching algorithm will help reduce the number of times our fuzzy system has to run to generate a new maximum bound for the backoff window. This history-based technique is presented here in the simulation environment of Castalia, which keeps records of the previously calculated *FuzzyMaxDelay* values and their corresponding data rate and $Channel_{Clear}Rate$ values in an array. In the case of a same "data rate and $Channel_{Clear}Rate$" match in the array in any instance of the simulation run, the fuzzy engine will not be carried out and will drive out the corresponding *FuzzyMaxDelay* value for that match to be used for the random backoff generation range in the next *n* super frames. This

makes the fuzzy-enabled MAC algorithm even more suitable to run on the resource-limited sensor nodes, which will be evaluated in Chapter 6 where we will implement the fuzzy-enabled MAC into the MAC layer of the SHIMMER sensor platforms. This array could be considered as a caching array which we name"*FuzzyMaxDelayArray*"and is a two-dimensional array. The dimensions of the array suggest the number of the inputs of the fuzzy system, which in this case, for energy evaluations of the fuzzy algorithm, is two (data rate and $Channel_{Clear}Rate$). Since no element of the defined array can be a decimal value and $Channel_{Clear}Rate$ values are potentially decimals we must convert them into integers before inserting them as an element of the array. For $Channel_{Clear}Rate$, based on the observations of its values in the trace file, we multiply the calculated decimal value by d_2=100, where d_2 signifies the level of accuracy for the recorded values of $Channel_{Clear}Rate$. If a fraction value of over 1000 decimal place happens it will simply be approximated to the nearest three-digit integer value after being multiplied by 100. The data rate values do not need to be converted since they are all integer values between zero and 125. Here is the caching array:

$$FuzzyMaxDelayArray=[(int)(MaxDataRate*d_1)][(int)(MaxChannel_{Clear}Rate*d_2)] \quad (12)$$

Where *MaxDataRate* and $MaxChannel_{Clear}Rate$ in (12) equal to the maximum data rate assigned to the nodes and the maximum value of $Channel_{Clear}Rate$ respectively. d_1 for data rate is always one since it is assumed that data rates never take on decimal values. The array keeps record of any repeated value during the entire simulation time and uses these values in the future if any combination of the node's data rate and its $Channel_{Clear}Rate$repeats over time. This way the fuzzy engine does not have to run over these similar values again to generate a *FuzzyMaxDelay*: it will simply dispatch the corresponding value to save both time and energy. The green lines in the fuzzy-enabled MAC algorithm pseudo-code illuminate the implementation of the fuzzy-enabled MAC and the difference the caching algorithm has made to the previous fuzzy algorithm at the MAC layer. Here is how the array is updated accordingly with a data rate of 20, for instance:

FuzzyMaxDelayArray[20,25]=60 //fuzzy engine running

FuzzyMaxDelayArray[20,150]=100 //fuzzy engine running

FuzzyMaxDelayArray[20,25]=60 //fuzzy engine does not run here to get the value of 60, it simply checks the value of the *FuzzyMaxDelayArray*[20,25] to dispatch the value.

The pseudo-code of the proposed MAC algorithm discussed in Section 4.2.1 must be updated accordingly as follows (note that Method1 as explained in Section 4.2.1 has been used for the calculations of the $Channel_{Clear}Rate$):

History-based fuzzy-enabled MACpseudo-code

```
void Mac802154b::startup() {
numberOfReceivedBeacon = 0;
  numberOfChannelIsCLEAR = 0;
  channelClearRate = 0;
  fuzzyBackoffDelayCacheArray = array(NULL);
  ...
  //Initializing all parameters
  //Set timer for starting the frame
setTimer(FRAME_START, 0);
}

void Mac802154b::initiateCSMACA() {
  // initiate CSMA-CA algorithm
setMacState(MAC_STATE_CSMA_CA);
  NB = 0;
  CW = enableSlottedCSMA ? 2 : 1;
continueCSMACA();
}

void Mac802154b::timerFiredCallback(int index) {
  switch  index {
    // Starting the frame
case FRAME_START
if  isPANCoordinator {
        // as a PAN coordinator, create and broadcast beacon packet
      } else {
        // if not a PAN coordinator, then wait for beacon
toRadioLayer(createRadioCommand(SET_STATE, RX));
setMacState(MAC_STATE_WAIT_FOR_BEACON);
setTimer(BEACON_TIMEOUT, guardTime * 3);
        // beacon timeout fired
        //indicates that beacon was missed by this node
      }
case BEACON_TIMEOUT
      lostBeacons++;
      //Perform all functionality needed for handling a lost beacon
      ...
case PERFORM_CCA
      // preform carrier sense
      CCA_result CCAcode = radioModule->isChannelClear();
if CCAcode == CLEAR {
        CW--;
if CW != 0 {
          // since carrier is clear,
          // no need to generate another random delay
      setTimer(PERFORM_CCA, unitBackoffPeriod * symbolLen);         }
else {
          numberOfChannelIsCLEAR++;
transmitNextPacket();
        }
```

```
      } elseif CCAcode == BUSY {
        CW = enableSlottedCSMA ? 2 : 1;
        NB++;
      }
    }
}

void Mac802154b::fromRadioLayer(cPacket * pkt, rssi, lqi) {
  //Recive and casting the packet
  Mac802154bPacket *rcvPacket=dynamic_cast<Mac802154bPacket*>(pkt);
switch rcvPacket->getMac802154bPacketType()
case MAC_802154_BEACON_PACKET
numberOfBeaconReceived++;
if numberOfReceivedBeacon == 20 {
         channelClearRate = numberOfChannelIsCLEAR / 20;
         numberOfReceivedBeacon = 0;
         numberOfChannelIsCLEAR = 0;
      }
case MAC_802154_ASSOCIATE_PACKET
      // Performe request to associate functionality
case MAC_802154_GTS_REQUEST_PACKET
      // Performe GTS request packet
case MAC_802154_ACK_PACKET
      // Performe ack frames
case MAC_802154_DATA_PACKET
      // Performe Data packet recive functionality
}

void FuzzyMac802154::continueCSMACA() {
index = int(channelClearRate*100);
if fuzzyBackoffDelayCacheArray[index] != NULL {
    fuzzyBackoffDelay = fuzzyBackoffDelayCacheArray[index]
  } else {
    previousChannelClearRate = channelClearRate
    Fuzzy Inputs, Output and Rule Definition;
    Fuzzy Inputs Initialization;
    fuzzyBackoffDelay = Fuzzy output.defuzzify();
    fuzzyBackoffDelayCacheArray[index] = fuzzyBackoffDelay;
  }
  rand = random(0, fuzzyBackoffDelay)
  simtime_t CCAtime = rand * (unitBackoffPeriod * symbolLen);
  // Perform CCA after CCAtime delay
setTimer(PERFORM_CCA, CCAtime);
}
```

Figures 5-14 and Figure 5-15 illustrate the difference in complexity between both MAC algorithms in terms of the “average number of fuzzy runs” that each MAC approach (with and without a caching array) goes through during a specified simulation run of 4000 seconds in Castalia. This simply signifies the importance of the proposed caching technique in providing a simpler fuzzy algorithm that makes our protocol as light as possible when running on real sensor platforms. The data rate configuration for this simulation scenario is as follows:

```
[Config allNodesDifferentRate]
SN.node[0].Application.packet_rate = 100
SN.node[1].Application.packet_rate = 5
SN.node[2].Application.packet_rate = 10
SN.node[3].Application.packet_rate = 10
SN.node[4].Application.packet_rate = 20
SN.node[5].Application.packet_rate = 20
SN.node[6].Application.packet_rate = 30
SN.node[7].Application.packet_rate = 30
SN.node[8].Application.packet_rate = 50
SN.node[9].Application.packet_rate = 50
```

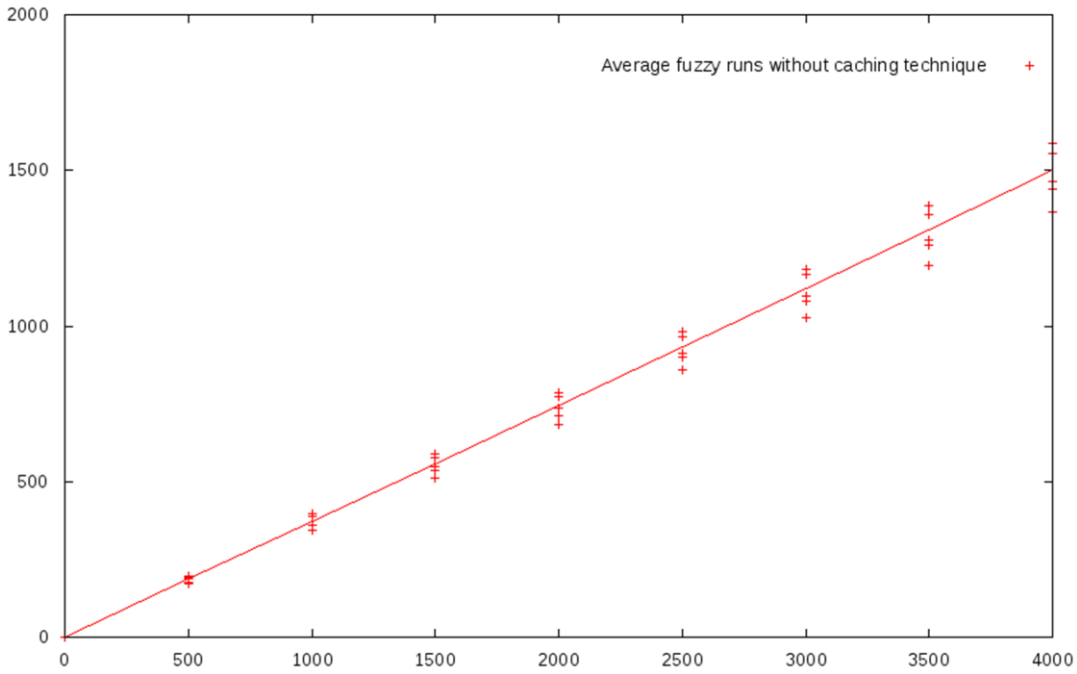


Figure 5-14: Average number of fuzzy iterations for all 10 nodes without the caching technique

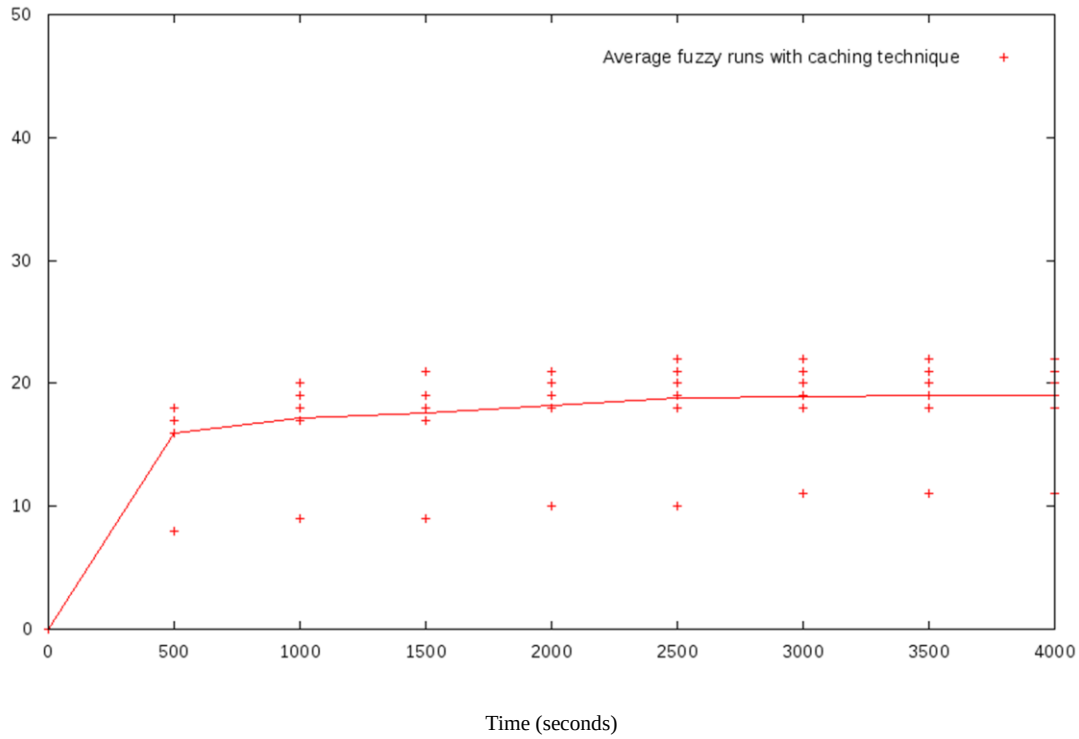


Figure 5-15: Average number of fuzzy iterations for all 10 nodes with the caching technique

Figure 5-16 shows the effect of the history-based techniquewith different runs on SHIMMER sensor's battery consumption at the application layer. It can clearly be seen from the figure that the number of received packets is greatly affected for a caching-enabled fuzzy algorithm with 70 runs per second compared to fuzzy algorithm with 70 runs with no caching technique. Such an effect can be explained by both the additional amount of battery spent and the longer delays endured when having no caching array with the fuzzy algorithm. Although the energy evaluations in Section 5.1.1.3 show no significant change in the level of energy between the two MAC approaches (original IEEE 802.15.4 and proposed fuzzy-enabled MAC), the traffic adaptive backoff period adjustment in the proposed MAC technique would compensate for the extra energy consumed by decreasing the delay times, thus leaving the energy levels untouched, as demonstrated in Table VII.

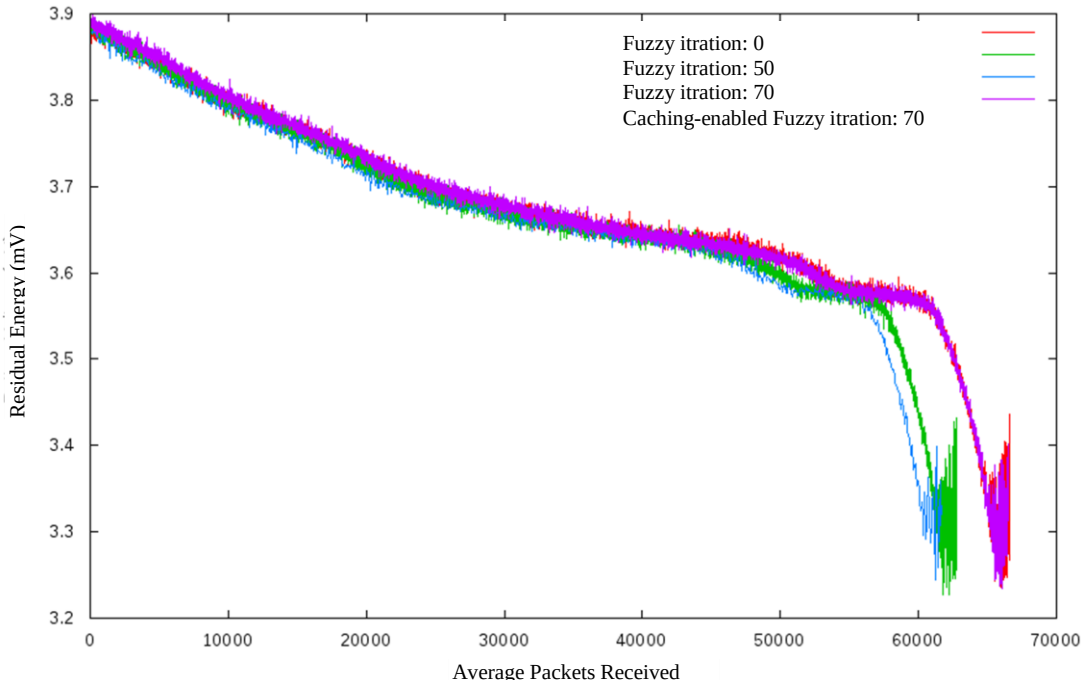


Figure 5-16: The effect of fuzzy runs on battery consumption of SHIMMER sensor platforms (fuzzy algorithm with and without caching array)

The fuzzy algorithm with caching capability with 70 runs per second actually has no effect on the energy consumed.

5.2.2.3 Reliability Evaluations

A backoff procedure would always set packet transmissions apart long enough so that, in the case of a deep faded channel, at least one retransmission finds the channel in its recovered condition. With our history-based fuzzy technique deployed in the CAP with a CSMA-based access scheme, we tried to improve the reliability factor to a reasonable extent. Traffic diversity in WBAN may degrade the reliability as it makes the received traffic unpredictable. We discussed earlier that if the network keeps the pace with the instantaneous traffic conditions imposed by different applications it can improve the reliability of data transmissions. This could simply be done by tuning the backoff-related parameters of the CSMA procedure as they take effect from the current traffic in the network. We discussed reliability earlier in Section 1.4.8. We also did some reliability measurements in Section 5.1.1 when different metrics such as average data rate and number of nodes would vary in the network. None of the previous evaluations of reliability, though, benefited from a history-based technique such as we described in Section 5.2.2.2. Here we show through simulations in Castalia how reliability can be improved further by implementing a history-based technique described in Section 5.2.2.2.

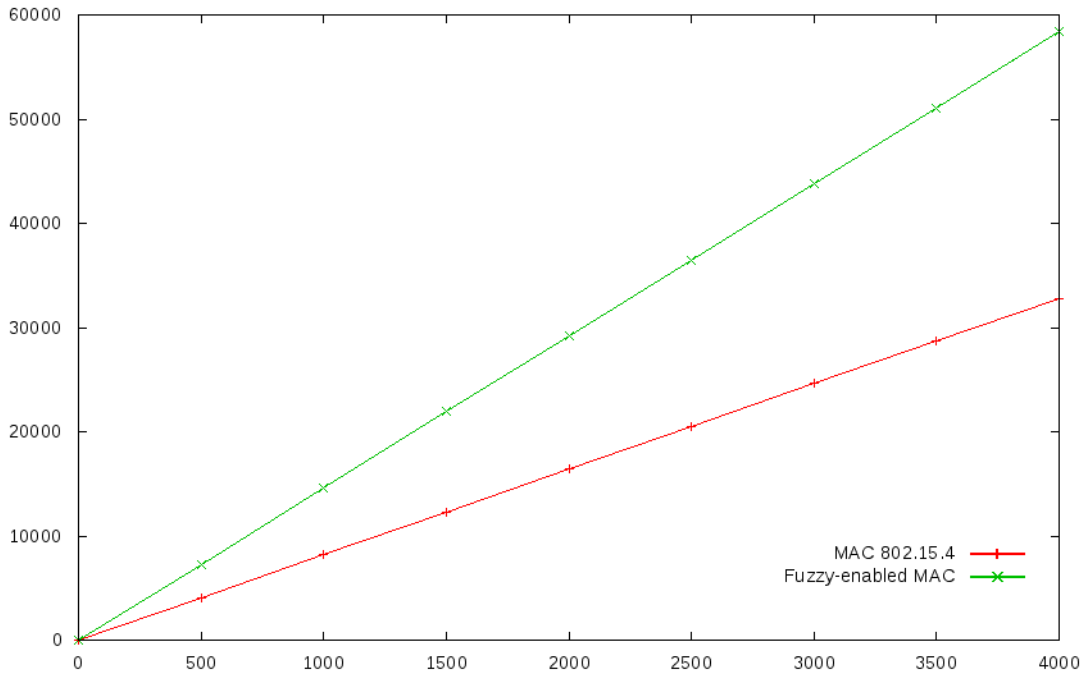


Figure 5-17: Average number of received packets versus simulation time; fuzzy-enabled MAC with the caching array versus IEEE 802.15.4. (Simulation time = 4000 seconds)

In Figure 5-17 we averaged the number of received packets over a period of simulation time to discover the reliability level of all the single communication links between the sensors and their coordinator during the entire simulation time. An assumed WBAN of 10 nodes was considered for testing reliability in this simulation. The simulations were done for 4000 seconds, which is over an hour. The green stroke shows the average packets received from 10 sensor nodes in the simulation transmitted to their coordinator when fuzzy-enabled MAC runs its caching algorithm, which has been contrasted against original MAC in IEEE 802.15.4. The data rate configuration is the same as described for simulations in the previous section.

5.3 Chapter Summary

After describing the main technique incorporated in our fuzzy-enabled MAC through two different methods (M1 & M2) in Chapter 4, we investigated the reliability factor of such fuzzy-enabled MAC in simulation environments in this chapter. We also conducted a feasibility study where the battery of a SHIMMER sensor node was tested to understand the amount of energy spent when only a fuzzy algorithm is running on it. The energy evaluations were conducted in the application layer but proved the fuzzy-enabled MAC to be promising in terms of energy efficiency. We also backed up this energy efficiency by adding a history-based technique that exploits a caching array to reduce the number of times the fuzzy engine has to run to output a

maximum bound for the backoff window. Although a second method was described in Section 4.5, we only addressed the reliability evaluations for it as described in Section 5.1.2, as this method is not implemented in any other case throughout the thesis. The main reasons are declared in Section 4.5, which mostly relates to the small possible backoff window range for the minimum default values of BE in the IEEE 802.15.4 standard. The reliability evaluation over long-term simulations (4000 seconds) was also discussed in Section 5.2.2.3, which shows the effect of our simplified history-based technique. The next chapter focuses on the real implementations on the SHIMMER sensor platforms, the detailed function bodies and also the evaluated reliability on a real WBAN implementation of five SHIMMER sensor platforms in a laboratory environment.

6. Chapter 6: IMPLEMENTATION ON SHIMMER SENSOR PLATFORMS

In this chapter we explain the development of our Fuzzy-enabled MAC (discussed earlier in Chapters 4 and 5) in TinyOS followed by the implementaion of a sample application using our Fuzzy-enabled MAC in a real sensor environment. We will discuss an experimental set up of a small WBAN composed of five SHIMMER sensor nodes acting as RFDs and one SHIMMER sensor node acting as a coordinator device. The basic structure of our testbed is shown in Section 6.1. Although some fundamental facts about the SHIMMER sensor platforms' specifications were given previously in Section 5.2.1, additional information are described in regard to the tools used during the experiment. We explained how we did necessary installations on the SHIMMER sensor platforms to form our WBAN where the presumed RFD devices would send their data packets to a coordinator device assumed to be an FFD. Discussions on how to have access to parameters of interest that are essential for our $Channel_{Clear}Rate$ calculations are given, which involve creating a possibility of saving the sensor nodes' data as they transmit to the coordinator device. We also included a brief introduction to the platform-independent implementation of the IEEE 802.15.4-2006 MAC in TinyOS environment called TKN15.4 (TKN15.4, 2014) in Section 6.3.1. A functionality test is discussed in Section 6.3.2, which compares the output values of our implemented fuzzy algorithm with random input parameters (within the valid ranges of our real input parameters) to justify the flawless and error-free performance of the fuzzy algorithm. Once the functionality of the implemented fuzzy algorithm with random inputs is proved in Section 6.3.2, we then move to Section 6.3.3 where real inputs for our fuzzy algorithm are generated and prepared. Applying the fuzzy algorithm to the CSMA/CA function in TKN15.4 is then described mostly in the form of pseudo-codes accompanied with necessary comments in Section 6.3.4. As the last section of this chapter, we test our target QoS parameter, reliability, in our implemented WBAN of five SHIMMER sensor nodes and a coordinator. Although we did energy testing through a feasibility study in Section 5.2.2.1, which proved an energy efficient trend of the fuzzy algorithm in terms of its computational complexity, we do not run any energy testing in this chapter. This is because the research aims to improve reliability as the target parameter of interest by addressing the fairness factor when nodes transmit their data on one shared medium. The concept of a dynamic backoff window for random generations of the backoff period is to provide a moderate and balanced access among all the sensor nodes so that reliability of data transmission is enhanced with a decent delay performance, which is of importance for medical

WBAN deployments. Here is an outline of the steps taken to implement our WBAN in the laboratory environment illustrated in Figure 6-1.

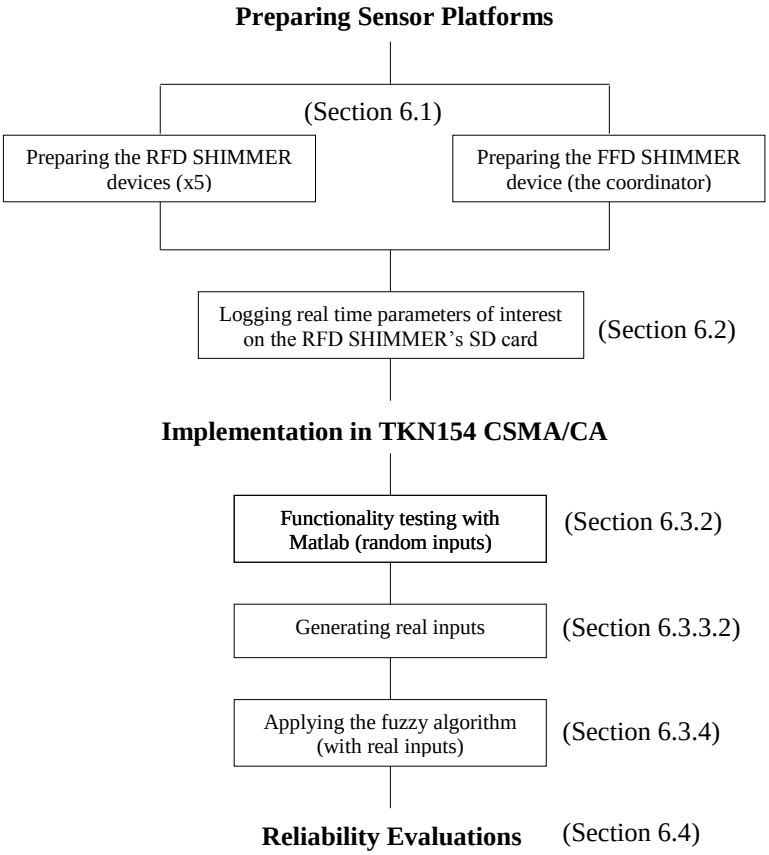


Figure 6-1: Real implementation process outline

6.1 Experimental Set up of a SHIMMER WBAN

For the experimental testing we used five SHIMMER sensors as the data forwarding nodes to a coordinator device, which is also a SHIMMER sensor node, in our WBAN. These sensor nodes send the packets with a rate of 100 packets in a second. The node's parameters such as channelIsClear, channelIsNotClear, and number of received beacon which are described in Section 6.3.3.2 will be calculated locally on each node and will be transferred periodically to the coordinator. These data will be saved on the coordinator's SD card for each node separately. We carried out the experiment several times and each time for an hour. The saved data in each experiment for each node showed a similar pattern in terms of the achieved values which is as expected due to same conditions each time each experiment was carried out. Some of the configuration parameters are shown in Table X below:

Table X: SHIMMER configuration parameter

RADIO_CHANNEL = 26
PAN_ID = 0x8172
COORDINATOR_ADDRESS = 0x4331
BEACON_ORDER = 5
SUPERFRAME_ORDER = 5
TX_POWER = -20 dB
Data_Rate = 100 P/S

For the transmission between the RFD devices in our designed WBAN and their coordinator device to start, we need to first perform the necessary preparation tasks at the application layer of these sensor devices. We used two programs to be installed on our SHIMMER nodes and also on the coordinator SHIMMER device. The utilized application is called TestData[1] the description of which is givenhere and extracted from (Hauer, 2011). This application is divided into two parts, one to be installed on the so-called RFD device and the other to be installed on the coordinator device, which is assumed to be an FFD. Here is the description of TestData application:

README for TestData

```
In this application one node takes the role of a PAN coordinator in
abeacon-enabled 802.15.4 PAN, it transmits periodic beacons and waits
forincoming DATA frames. A second node acts as a device, it first
scans thepre-defined channel for beacons from the coordinator and once
it finds a beaconit tries to synchronize to and track all future
beacons. It then starts totransmit DATA frames to the coordinator as
fast as possible (directtransmission in the contention access period,
CAP).

Criteria for a successful test:
Coordinator and device should both toggle LED2 about twice per second
inunison. They should also each toggle LED1 about five times per
second (butnot necessarily in unison). Note: the nodes should be close
to each other, because the transmission power is reduced to -20 dBm.

Tools: NONE
Usage:

1. Install the coordinator:

    $ cd coordinator; make <platform> install
```

[1] Author/Contact: Jan Hauer hauer@tkn.tu-berlin.de

2. **Install one or more devices**

```
$ cd device; make <platform> install,X
```

```
Where X is a pre-assigned short address and should be different for
every device.You can change some of the configuration parameters in
app_profile.h

Known bugs/limitations:
Many TinyOS 2 platforms do not have a clock that satisfies
theprecision/accuracy requirements of the IEEE 802.15.4 standard (e.g.
62.500 Hz, +-40 ppm in the 2.4 GHz band); in this case the MAC timing
is not standard compliant.
$Id: README.txt,v 1.3 2010-01-05 17:12:56 janhauer Exp $o
```

A brief description of installing the above application on the coordinator device and the five RFD (physiological SHIMMER sensor platforms) in our star topology appears below.

To install the TestData application on the to-be-coordinator sensor device in our WBAN, we first connect the sensor through the SHIMMER dock to the laptop and then we run the command "`make shimmer2r install bsl,/dev/ttyUSB0`" in the application's directory, which in our case is:

```
/tinyos-main/apps/tests/tkn154/beacon-nabled/TestData/coordinator
```

The application to be installed on the RFD devices in our case is located in the following directory:

```
/tinyos-main/apps/tests/tkn154/beacon-enabled/TestData/device
```

We therefore run the command below to install the TestData Device program on each of the sensors that are going to be our five RFD devices:

```
make shimmer2r 1 install bsl,/dev/ttyUSB0
```

Note: in the command above, 1 is a unique address which shall be considered for each device.

At this point the LEDs on two nodes (the coordinator and the device) start to blink, which shows a data transmission between them. The coordinator node will send periodic beacon messages to RFD devices and the devices will send their data packets to the coordinator.

The number of nodes for this real implementation is five, which communicate with their coordinator in a star topology as shown in Figure 6-2. The coordinator is connected to the laptop via a SHIMMER dock, which can be seen in the figure. We can access the data that have been saved on coordinator device via this connection. This is a Shimmer Dock 2 platform which has three main buttons and two LED lights on its board. This board has changed to only one button and one LED in SHIMMER 3 generation. The SHIMMER kits that we have used are SHIMMER 2 generation.This dock is used to both recharge the batteries of the SHIMMER sensors and to program them and also to access the MicroSD card of the SHIMMER sensor platform. It has three buttons on its board; the one that does not have any LED is the user button that indicates the application specific signal to SHIMMER. This user button is next to the GPS enable button which toggles between GPS and Host PC UART connection to Shimmer. We did not use this button's functionality as it did not relate to our application scenario. In the SHIMMER dock's manual (Shimmer Research Group, 2014, Shimmer Dock 2 Quick Start Rev 1b User Manual) it is stated that the GPS enable button is used to switch between GPS and host computer UART functionality. In GPSmode, the indicator on the button will blink green while the GPS location is being acquired. The indicatorwill turn solid after a lock is reached. The standing alone button isto power-on, reset, or power-off SHIMMER. A SHIMMER sensor node gets connected to this dock with the connector in the centre of the board. The board is connected to the laptop via a mini USB port. The different colors of the LED aid understanding of the different situations caused. A green LED light on the reset button, for instance, means the SHIMMER is powered on. A detailed description appears in the manual (Shimmer Research Group, 2014, Shimmer Dock 2 Quick Start Rev 1b User Manual, p. 25). The drivers for the MicroSD Card Access should have already been installed on the PC or laptop. These USB Serial Converter drivers should be manually installed to use the Dock for programming the SHIMMER. We used the sensor platforms of a platinum kit and a kinematic kit. The platinum kit includes sensors as described earlier in Section 5.2.1.

Figure 6-2: Real implementation testbed in lab environment; five SHIMMER platforms as RFD devices and one (connected to dock) as the coordinator device

Now that the five RFD sensor nodes have started their transmission to the coordinator device at their specific rates, we need to carry out the experiment for an hour and then collect the data which have been saved on the coordinator's SD card. We will then have to recharge the batteries of the sensor nodes to do the experiments again in the same conditions and without changing the relative positions of the nodes to their coordinator. In some occasions it might happen with the SHIMMER sensor nodes that they would stop working, in such case we had to carry out the experiments again.

6.2 Logging Sensor Activity on the SD Card

The advantage with the Castalia simulation environment when implementing our fuzzy method in a CSMA/CA algorithm was that we could clearly monitor and save the real time values of the important parameters of CSMA/CA such as the number of times the channel was clear and the number of times it was not clear in a file and investigate their behavior. For example, we performed extensive simulation runs in Section 4.3 to test the minimum and maximum values of $Channel_{Clear}Rate$, which was calculated based on the number of CCA_2 in intervals of time during the simulation run. For doing the same thing in a real environment, i.e. the sensor platforms, we must provide the possibility of saving the real time values of desired parameters such as $Channel_{Clear}Rate$ as the sensors start transmitting their data to the coordinator device.

In this section we explain how we created a file on the SHIMMER's SD card that can save the CSMA/CA parameters'values in it. The important point to note here is that in the real environment there are many limitations for creating a file and working with the contents of a file as compared to the simulation environment. To start, we used one of the already existing firmware available in SHIMMER, the `JustFATLogging`. We modified this firmware to create an application that would log some of the MAC level parameters such as the number of times the channel was and was not clear plus the number of beacons and the sensor'stime in its file. We would then be able to use such data to design and evaluate our fuzzy algorithm into the CSMA/CA process of MAC sub layer on SHIMMER devices.

For logging the CCA and other desired MAC parameters on the SD card of the SHIMMER platforms we used the FatFs[2] interface that exists in the following directory:

```
/tinyos-main/tos/platforms/shimmer/chips/sd/fatfs
```

The FatFs interface is provided by the FatFsp module, which can be reached via the above directory. The pseudo-code of applying FatFsp to record sensor data on the SD cart is provided as below:

Recording the data on the SD cart consists of two main steps. The first step is to add the necessary wiring configurations in the `CC2420TKN154C configuration` file as below:

```
configuration CC2420TKN154C
components FatFsP, diskIOC, LedsC;
PHY.FatFs      -> FatFsP;
FatFsP.diskIO              -> diskIOC;
FatFsP.diskIOStdControl    -> diskIOC;
```

The second step is to declare the using of FatFs interface, initializing some parameters, and finally opening and writing the data in a file on the SD cardas below:

[2] The FatFs implementation is a direct port of the ChaN FatFs project (http://elm-chan.org/fsw/ff/00index_e.html) to TinyOS. During testing of the initial port, Victor Cionca at University ofLimerick discovered a great deal of overhead in the file system'scluster window operations, and devised an improved method to handlethese without compromising the integrity of the fs.

```
module CC2420TKN154P
provides
     …
  Uses
     …
     interface FatFs;
  implementation
  FATFS fs;
    FIL fp;
         boolean isClose = true;
    …
    void nextIterationSlottedCsma()
   uint16_t buf[];
if (isClose)
      isClose = 0;
            FatFs->mount(fs);
            FatFs->fopen(fp, "data.dat", (WRITE READ));
      FatFs->fwrite(buf);
```

Now that we can save the data on the coordinator's SD card, we have to provide and calculate the required parameters on each node and transmit the data to the coordinator. In the next section we explain how to prepare and calculate these data.

6.3 Implementing the Fuzzy Algorithm in CSMA/CA of SHIMMER

6.3.1 Introduction to TKN15.4

Shifting from the simulation environment to real sensor platforms and implementing our method at MAC level would not be feasible if the standard's MAC implementation is not available in its operating system, TinyOS. In 2009, the Telecommunication Networks Group from Technical University Berlin designed a platform-independent implementation of the IEEE 802.15.4-2006 MAC in TinyOS environment called TKN15.4 that provided the possibility of some empirical works besides mere simulation-based research in academia. Flexibility was one of the main factors to be considered in their design criteria as it would break the entire complex of the IEEE 802.15.4 MAC into many self contained components. This was largely inspired by the numerous configuration options such as the standard's support for different modes of access (explained in Section 1.7) and different topologies (star and peer-to-peer) that emphasized an adaptable and pliable design. In TKN15.4, the IEEE 802.15.4 was modeled as closely as possible to its defined specifications in the standard through event-based programming. There is no limitation on what platform TKN15.4 can be used on as long as the operating system of the device is TinyOS2, which is why we can easily use its MAC implementation on our SHIMMER sensor devices.

TKN15.4 MAC implementation is in the tinyos-2.x/tos/lib/mac/tkn154 directory of the TinyOS 2. For the TKN15.4 to be available on a TinyOS 2 sensor platform there are three main requirements to be met:

- Six main interfaces must be available which push many time critical applications to be carried on by the PHY layer rather than the MAC (TKN15.4, 2014). The parameters of these six interfaces are set by the MAC layer but the algorithms are implemented and executed within the PHY layer. These six interfaces are:
    - RadioRx
    - RadioTx
    - RadioOff
    - SlottedCsmaCa
    - UnslottedCsmaCa
    - EnergyDetection

- For the MAC implementation, two main interfaces of *timer* and *alarm* are used, which are standard TinyOS 2 interfaces. The required precision/accuracy for the chosen platform may however be different from the one provided by TinyOS, in which case it must be extended accordingly.

- The configuration component of the platform's TinyOS 2 must connect the platform-independent implementation of the TKN15.4 MAC (`TKN154BeaconEnabledP` or `TKN154NonBeaconEnabledP`) with the platform-dependent timers and PHY drivers.

6.3.2 Testing the Fuzzy System in CSMA/CA

In this section we test our fuzzy system in CSMA/CA algorithm of the MAC sub layer (in TKN15.4) based on the input parameters' range of our fuzzy system in the Castalia simulator. We first create the same fuzzy system in Matlab and and in the TKN15.4 environment of the SHIMMER sensor platform. The rule-set used for both of the fuzzy systems is the same as the rule-set in our Castalia simulations. The goal of this testing is to test the functionality of the fuzzy algorithm by some valid (within the defined range) parameters. For this aim, we decided to make a comparison of the results with the same fuzzy system in Matlab. The results of such validation show a great resemblance between the two fuzzy systems (in TKN15.4 and in Castalia). To start the testing, we implemented the fuzzy system by random input parameters

but within the valid range i.e. Input1 is randomized in the interval of [0–100] and Input2 is randomized in the interval of [0–12]. The generated output of the fuzzy system working with these two inputs will be stored on the sensor's SD card. We then created the same fuzzy system in Matlab with exactly the same generated inputs as saved in the SD card to examine the output of the fuzzy system. The generated output values in Matlab represented almost the same values as the output values of the fuzzy algorithm implemented in TKN15.4 of the SHIMMER sensor nodes, with only a negligible difference. The very small difference that sometimes happened in the output values is because on SHIMMER sensor platforms the decimal calculations will automatically be approximated to the next lesser value. A snapshot of the obtained values, both in real sensor platforms and Matlab appearin Table XI:

Table XI: Input and output values' comparison (in Matlab and Castalia)

Data rate (Input1)	$Channel_{Clear}$Rate (Input2)	Matlab Fuzzy Output	Fuzzy System in TinyOS Output
32	12	84	84
38	5	30	30
76	12	62	63 *
75	2	15	12
59	8	40	40
30	2	15	14 *
36	2	15	12 *
19	9	85	88 *
44	4	30	30
29	10	70	70

* The slight difference in values of some of the output parameters in our fuzzy system and that of Matlab is because of the limitations that exist in TinyOS for the calculations and storing operations of floating numbers.

6.3.3 **Generating Real Inputs of the Fuzzy System in CSMA/CA**

6.3.3.1 TKN Structure in TinyOS

Before we describe how we prepared our inputs for the fuzzy-enabled MAC at the MAC layer, we need to briefly explain the TKN154 structure. An overall view of the related existing directories in TKN in TinyOS appears below in Figure 6-3. In this section we briefly explain the main files we used for our implementations on SHIMMER sensor platforms in these directories.

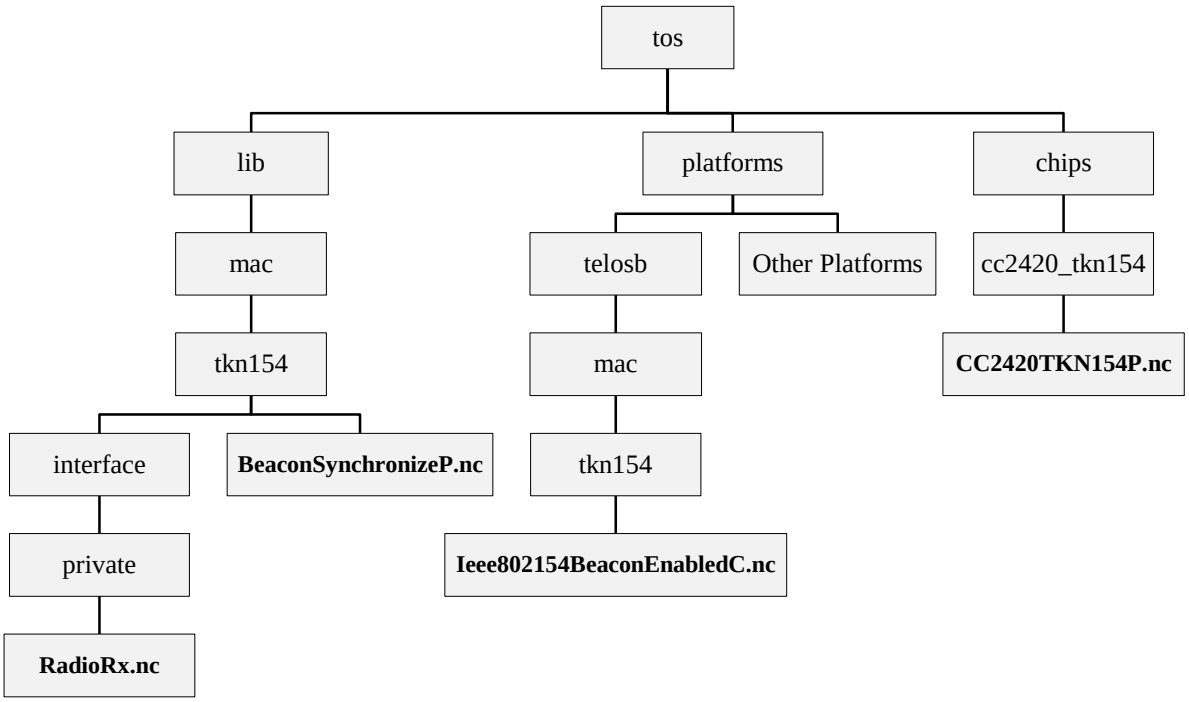


Figure 6-3: Related directories in TKN154 in TinyOS

In the topmost level `tos/lib/mac/tkn154` exists which contains a platform-independent IEEE 802.15.4-2006 MAC implementation. It is essential for the platform to provide three things so that the code can be used on a TinyOS 2 platform. These three are: 1) a suitable radio driver; 2) alarms/timers with symbol precision; and 3) some "platform glue" code (defining guard times, etc.). Currently the supported platforms are telosb, shimmer2(r), and micaz. As of July 13, 2011, the MAC also includes the IEEE 802.15.4-2006 GTS services (Hauer, 2012); this part of the implementation was contributed by CISTER/ISEP, Polytechnic Institute of Porto. This directory also contains the `BeaconSynchronizeP` module, which is responsible for periodic beacon tracking in a beacon-enabled PAN. This module utilizes the `RadioRX` interface. The `tos/lib/mac/tkn154/interfaces/private` directory is a collection of private interfaces that TKN uses to make communicating among its components and modules possible.

The next important directory is `tos/platforms;` as its name implies it has the hardware platforms that TinyOS runs on. A hardware platform as described in (Levis, 2005) as "a collection of "chips", such as microcontrollers, radios, and non-volatile storage, whose implementations can be found in tos/chips. Chip functionality depends on a platform as it provides the necessary code for the hardware resources to operate. For example, the CC1000 chip has a software networking stack (in chips/cc1000), which depends on being able to send

bytes to the CC1000 chip over an SPI bus (https://github.com/tinyos/tinyos-main/tree/master/tos/platforms). How the SPI bus works is platform dependent (e.g., it could be shared between several chips and require software arbitration). Therefore, the mica2 platform, which has a CC1000 radio, connects the CC1000 code to an SPI bus that its microcontroller, an ATmega128, provides".

The final directory that we work with is `tos/chips`, which is designed to house code to drive a particular chip in a way that can be shared across disparate platforms that share that particular chip (Turon, 2005). The `TKN154 CC2420` driver is located in `tos/chips/cc2420_tkn154`. This directory contains the file `CC2420TKN154P.nc` which is the main implementation file of the IEEE 802.15.4 MAC specifications. Therefore it contains all the necessary functionalities that are required for the CSMA/CA procedure. This file is also one of the providers of the `RadioRx` interface.

6.3.3.2 Providing the Inputs in TKN154

Our algorithm requires two main parameters, data rate and $Channel_{Clear}Rate$. The data rate parameter is read from the application layer, which is not complicated at all. For calculating the $Channel_{Clear}Rate$ according to M2 we need three parameters. Firstly the number of times the channel has been assessed as clear which is denoted as channelIsClear in pseudocode provided, secondly the number of times the channel has not been assessed as clear denoted as channelIsNotClear, and thirdly the number of received beacon packets. The first two parameters of the above mentioned three parameters were implemented and calculated as integer parameters in `CC2420TKN154P.nc` module. The third parameter, which is the number of received beacons, is tracked by the `BeaconSynchronizeP.nc` module. Since we need this parameter in the `CC2420TKN154P.nc` module, we used the `RadioRx` interface, which is a communication interface between `BeaconSynchronizeP.nc` and `CC2420TKN154P.nc`. We defined a function named `beaconReceived`, which is called by the `BeaconSynchronizeP.nc` module and is provided by the `CC2420TKN154P.nc` module. The body of this function simply contains a counter to keep account of the number of received beacons and also calculates the $Channel_{Clear}Rate$ in every 20 super frames.

6.3.4 **Applying the Fuzzy System in CSMA/CA with Real Inputs**

In the previous section we gave a brief introduction to the related modules in TKN154 and how they interact with each other. The following pseudo-codes describe such interaction in detail and show how the required MAC parameters for our implementation were provided.

The pseudo-code for `BeaconSynchronizeP.nc` appears below; as described in the previous section this module is responsible for tracking the beacon frames received in the network. The main event in this module is `BeaconRx.received()` which is called when a beacon frame is received. It will be checked inside this function whether the received beacon is from the same network coordinator or an outsider coordinator. If the received beacon is from the same network coordinator, the `BeaconReportRx.beaconReceived()` willbe called which signals the `beaconReceived()` implementation in `CC2420ReceiveP.nc`.

BeaconSynchronizeP.nc

```
module BeaconSynchronizeP
  uses
    interface RadioRx as BeaconReportRx;
    ...
  provides
    ...
  event message_t* BeaconRx.received(message_t *frame)
    if Got a beacon from my coordinator
      call BeaconReportRx.beaconReceived();
    ...
  ...
```

The pseudo-code below is related to the `RadioRx` interface whose implementation only includes the methods that its provider has to provide. The method that we have defined for our algorithm is `beaconReceived()` which, as described in Section 6.3.3.2, is responsible for wiring the `CC2410ReceiveP.nc` with `BeaconSynchronizeP.nc` and also informing the `CC2410ReceiveP.nc` module from any received beacon frame.

RadioRx.nc

```
interface RadioRx
  async command error_t beaconReceived();
  ...
```

The pseudo-code below is for the main part of our algorithm implementation which is in the CSMA/CA. As can be seen, this module provides the `RadioRx` interface and therefore we must

implement the `beaconReceived()` command. This function is responsible for tracking the received beacon frames and calculating the $Channel_{Clear}Rate$ based on Method2 (M2). As evident from the code, the $Channel_{Clear}Rate$ is calculated every 20 super frames and the `numberOfReceivedBeacon` counter resets to zero. The `waitBackoffDoneSlottedCsma` function of the second CCA assessments (CCA_2) is checked and the `channelIsClear` and `channelIsNotClear` parameters will be updated accordingly. The `nextIterationSlottedCsma()` is responsible for performing the next iteration on the slotted CSMA. This function calculates the *FuzzyMaxDelay* parameter as the output of the fuzzy system, which gets called from inside `nextIterationSlottedCsma()`.

CC2420ReceiveP.nc

```
#include "tinyFuzzy.h"
module CC2420TKN154P
  uses
      ...
  provides
      interface RadioRx;
      ...
  implementation
      uint32_t channelIsClear = 0;
      uint32_t chanenlIsNotClear = 0;
      uint32_t numberOfReceivedBeacon = 0;
      uint16_t channelClearRate = 0;
      uint16_t isClose = 1;
      uint16_t dataRate = GetDataRateFromApplication()

  event void Boot.booted()
      initializeFuzzy();

  async command RadioRx.beaconReceived()
      numberOfReceivedBeacon++;
      if numberOfReceivedBeacon == 20
           numberOfReceivedBeacon = 0;
          channelClearRate =
                      (chanelIsClear /

(channelIsClear +    channelIsNotClear)) * 10;

  void waitBackoffDoneSlottedCsma()
      if Second CCA succeeded
          channelIsClear++;
      else
          channelIsNotClear++;

  void nextIterationSlottedCsma() Next iteration on slotted
CSMA
      uint16_t buf[BUF_SIZE];
      buf[] = array{
                          channelIsClear,
                          channelIsNotClear,
                          numberOfReceivedBeacon};
    if isClose == 1
```

```
            isClose = 0;
            call FatFs.mount();
            call FatFs.fopen("data.dat");
        call FatFs.fwrite(buf);

        fuzzyMaxDelay        =        optimizeTinyFuzzy(dataRate,
    channelClearRate);
        backoff = random(0, fuzzyMaxDelay) * unit backoff periods
        call ReliableWait.waitBackoff(backoff);
      ...
```

6.4 **Evaluations**

In this section we measure the reliability of our fuzzy-enabled MAC implemented on the TinyOS of the SHIMMER sensor platforms in terms of the average packets received at the coordinator device. For reliability measurements we need to let the sensor devices send the data at a specific rate to the coordinator device. We then have to average the number of successfully received packets at the coordinator device from all the sensor nodes in the network to have a figure of reliability offered by our fuzzy-enabled MAC.

Figure 6-4 illustrates the average number of received packets by all the five SHIMMER sensors at the coordinator device. The five sensor nodes have our fuzzy-enabled MAC implemented in their CSMA/CA, which is compared by a green stroke to the original IEEE 802.15.4 MAC (red stroke). We have defined an array on the coordinator device whose index represents the node's ID. The array's value for each node at any time indicates the total number of received packets from that node until that time instant. Every time a packet is received at the coordinator device, the array is updated with respect to the node's ID and its value will be saved at the coordinator's SD card with the node's ID. The X axis shows the number of times that data has been recorded on the coordinator's SD card. The Y axis represents the average number of packets received from all the sensors.

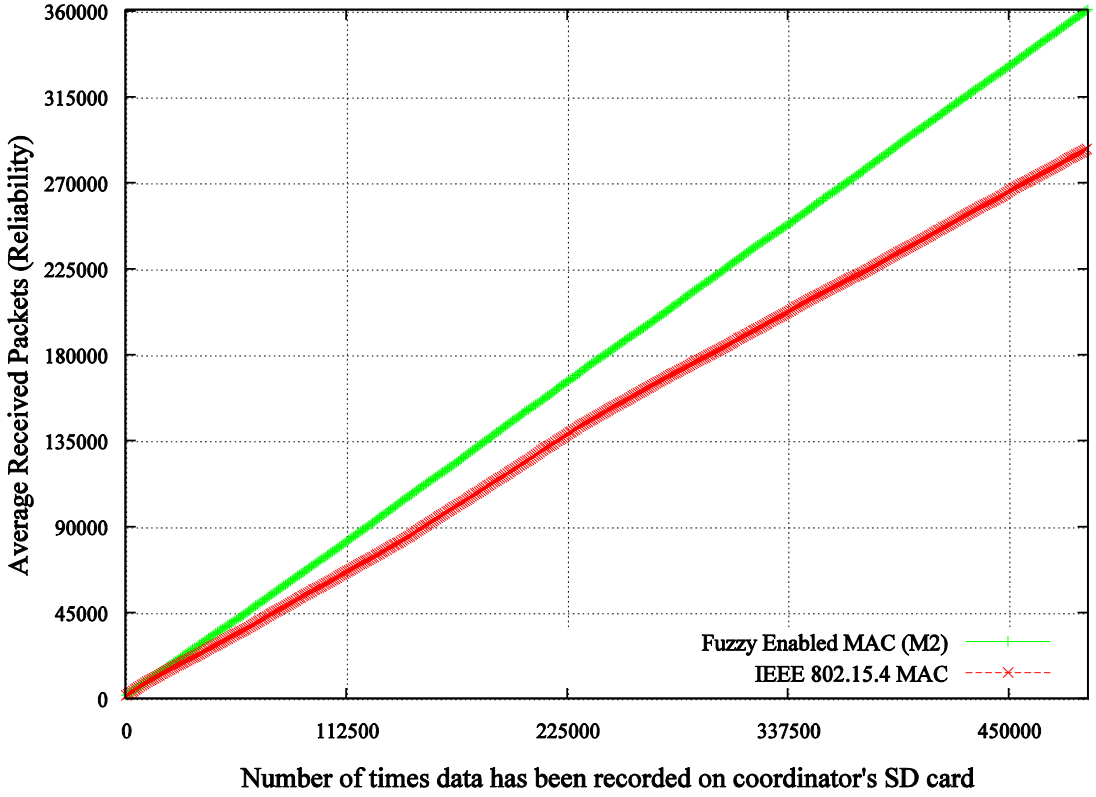


Figure 6-4: Average number of received packets by all the five SHIMMER sensors at the coordinator device; fuzzy-enabled MAC (green stroke) versus IEEE 802.15.4 MAC (red stroke); (X axis: Number of times data has been recorded on the coordinator's SD card).

It can be seen from the results that our fuzzy-enabled MAC outperforms the IEEE 802.15.4 MAC in terms of reliability of the received data at the coordinator device which is achieved by a more fair access to the shared channel among the five SHIMMER sensor nodes in the study. A difference of almost 20% is made in the average number of received packets from all the SHIMMER sensor nodes in our star topology at the coordinator device during an hour long experiment.

Figure 6-5 shows the number of successful CCA_2s and the number of times the CCA_2 was not a success in a SHIMMER sensor node that our fuzzy-enabled MAC runs on. This experiment was carried out for one of the SHIMMER sensor platforms during a three-minute experiment to reveal the difference between these two values when a SHIMMER sends out the data with our fuzzy-enabled MAC to coordinator device. The green stroke represents the number of times the channel was clear (a CCA_2 incident) and the red stroke shows the experience of a failed CCA_2. Each point in this diagram (red or green) implies the number of times a successful or

non-successful CCA_2 was experienced during the last 20 super frames. The number of successful CCA_2s is substantially higher than the unsuccessful CCA_2s which implies the fact that the proposed fuzzy-enabled MAC algorithm does help with an experience of an idle channel. A successful CCA_2 only takes place when the channel has been idle for two consecutive times, each time after waiting for a randomly generated backoff. Our fuzzy-enabled MAC increases the chances of such clear channel assessment by contributing to a traffic-aware and fair adjustment of the backoff window from which the required backoff period will be randomly generated from. This will yield to a more balanced and moderate waiting periods for each individual node as also described earlier in Chapter 2.

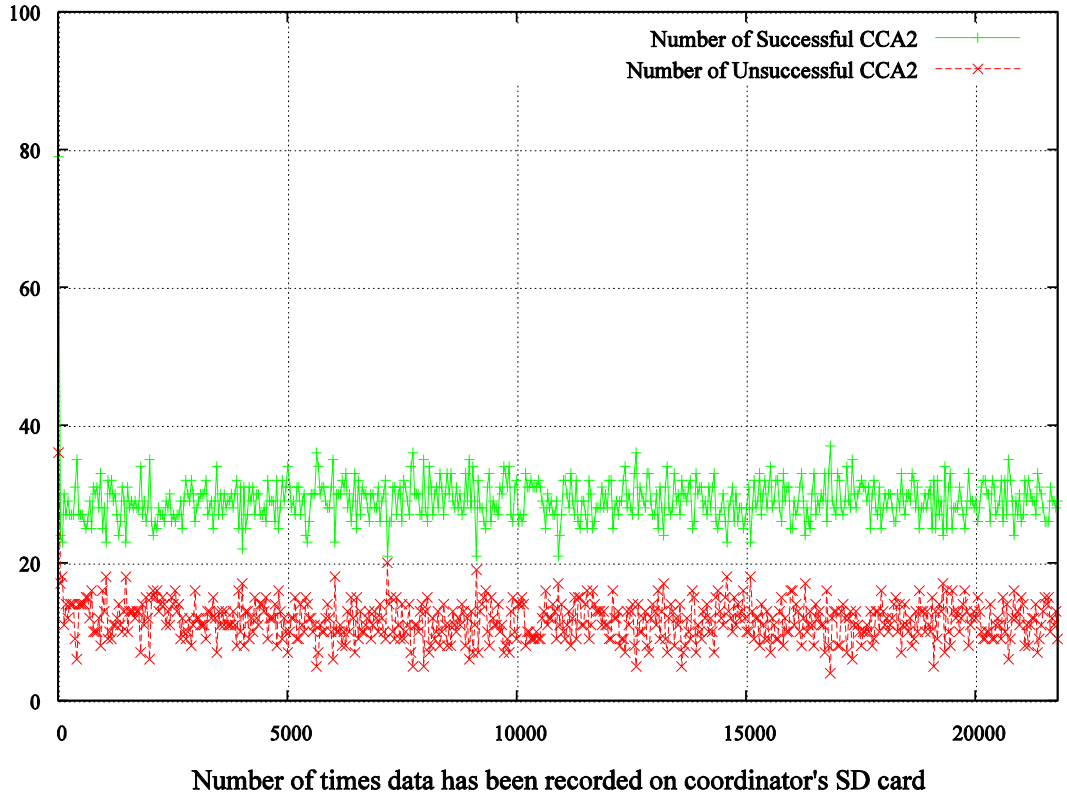


Figure 6-5: Successful CCA_2 (green stroke) versus unsuccessful CCA_2 (red stroke) with fuzzy-enabled MAC running on a SHIMMER sensor platform; (X axis: Number of times data has been recorded on the coordinator's SD card)

Figure 6-6 is similar to the previous figure except it shows the behavior of IEEE 802.15.4 MAC. By comparing these two figures it becomes clear why our algorithm has been more successful in sending the packets. The reason is a higher CCA_2 rate that affects the number of successfully sent packets from a SHIMMER sensor node. IEEE 802.15.4 performs almost the

same in terms of the successful and unsuccessful CCA_2s as shown in the figure. This implicitly implies a non-traffic-aware behavior of the MAC protocol which leads to no specific change of backoff periods with respect to nodes' past experiences in having access to the channel. In IEEE 802.15.4 the length of the generated backoff period would only depend on the previous channel-sensing outcome. If a failed clear channel assessment is faced the BE exponent will be incremented for the next backoff period generation and the sensor node has to perform the channel clear assessment for at least two more times in its best case.

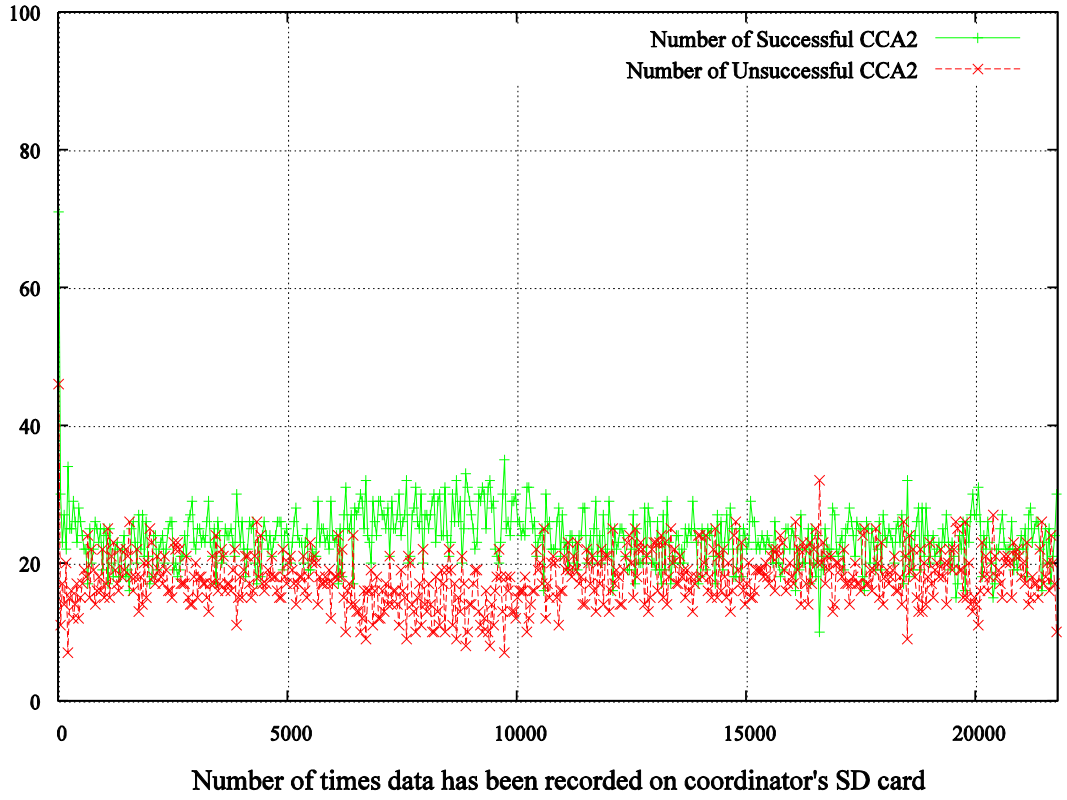


Figure 6-6: Successful CCA_2 (green stroke) versus unsuccessful CCA_2 (red stroke) with IEEE 802.15.4 MAC running on a SHIMMER sensor platform; (X axis: Number of times data has been recorded on the coordinator's SD card).

Figure 6-7 shows the values of $Channel_{Clear}Rate$ as one of the inputs of our fuzzy system. The green stroke shows the value of this parameter in our fuzzy-enabled MAC and the red stroke shows the value of this parameter for IEEE 802.15.4. $Channel_{Clear}Rate$ was calculated based on Method2 (M2) for our fuzzy-enabled MAC. In M2 the $Channel_{Clear}Rate$ is calculated over *all* the trials in every *n* super frames. This includes both successful and unsuccessful CCAs carried out during *n* super frame time interval. One interesting outcome of this comparison is the effect that $Channel_{Clear}Rate$ has on its next calculated values in our method. Since in IEEE

802.15.4 these values of $Channel_{Clear}Rate$ are not used for any purpose in the repeating loop of CSMA/CA, a fair and consistent pattern is not visible in its calculated $Channel_{Clear}Rate$ values. In fact the $Channel_{Clear}Rate$ parameter is calculated for IEEE 802.15.4 MAC protocol based on both the number of successful and also unsuccessful CCAs during *n* super frames but it is not integrated with the repeating loop of CSMA/CA algorithm. We have only calculated the value of this parameter for IEEE 802.15.4 MAC on this occasion to find out the difference of using $Channel_{Clear}Rate$ as one of the input parameters of our fuzzy algorithm in our method on how successful the next channel assessments will be.

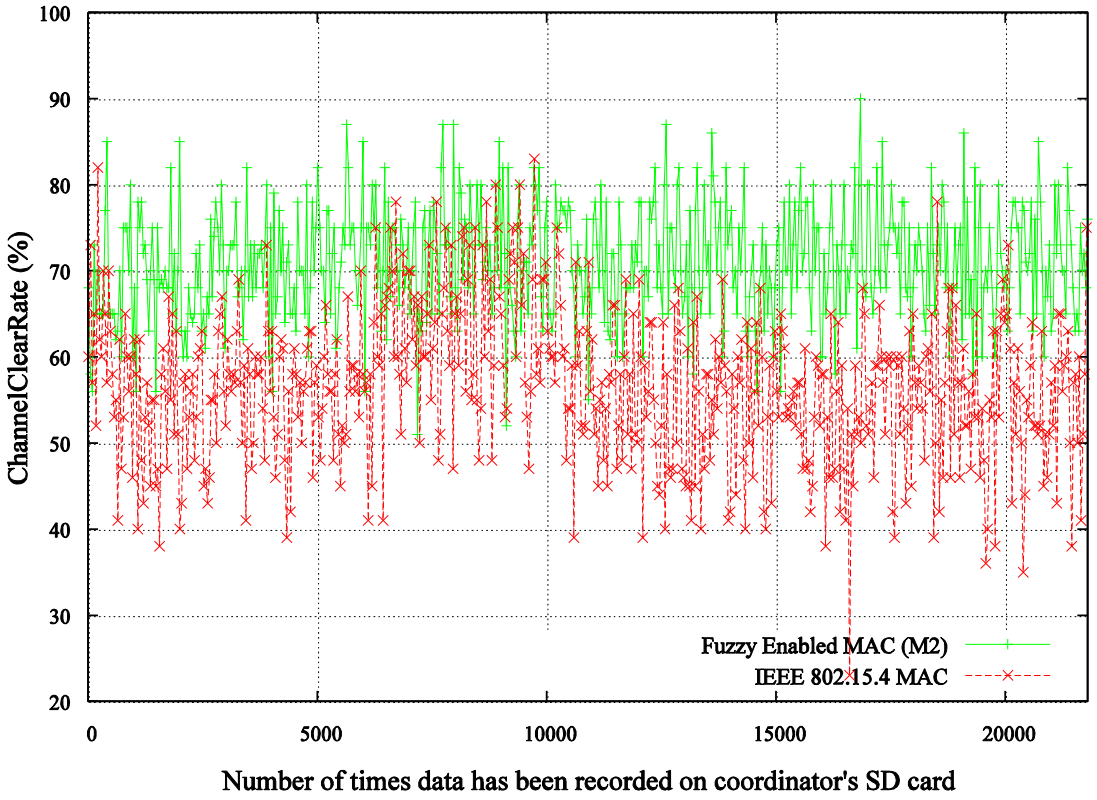


Figure 6-7: $Channel_{Clear}Rate$ values; fuzzy-enabled MAC versus IEEE 802.15.4 MAC; (X axis: Number of times data has been recorded on the coordinator's SD card).

Although the reliability evaluations in this section were done in lab environment they have contrasted our fuzzy enabled AMC against IEEE 802.15.4 MAC in the same conditions which makes the comparisons made reliable. Reliability was calculated based on all the packets received from all the five sensor nodes at the coordinator device when they ran the fuzzy-enabled MAC on them for packet scheduling. The CCA_2 parameter was a good parameter to examine as each successful CCA_2 represents a successful channel access which is indeed

directly associated with a better slot allocation performance of the MAC layer protocol engaged. This itself emphasize on the significance of a traffic aware and traffic adaptive dynamic backoff behavior presented in fuzzy-enabled MAC which has also been previously investigated and justified through simulation results.

6.5 Chapter Summary

An experimental study was carried out in this chapter as the last chapter of the thesis to implement the introduced fuzzy-enabled MAC in Chapter 4 on TinyOS and in the nesC language. Different challenges were encountered, shifting from the simulation environment to the real sensor platforms, which were discussed and addressed. Having more difficulty in viewing the real time values of input parameters, or saving such data on the sensor's SD card were amongst the few problems identified and resolved. The reliability metric was successfully measured for WBAN of five SHIMMER sensor platforms over an hour-long experiment (for several times) that proved a better performance. Although the feasibility study carried out in Section 5.2.2.1 shows a decent energy efficiency performance of the SHIMMER sensor platforms' batteries when running a fuzzy system, the energy measurements were not discussed at MAC layer as the research aims more towards a reliability-guaranteed MAC protocol that is specific to low data-rate monitoring applications. Most of the implementation works in this chapter were shown in the form of pseudo-codes, as it involved extensive programming to integrate the fuzzy algorithm into the CSMA/CA mechanism of TKN154, which is an implementation of the IEEE 802.15.4 MAC specification. The exploited and added modules were described to assist understanding the implementation stage as much as possible.

7. CHAPTER 7: RESEARCH CONTRIBUTIONS AND CONCLUSION

This thesis has been carried out within the context of the WBANs and their application area in healthcare. Different design criteria can be considered in an implementation of a WBAN which merely depend on the certain application's requirements. We narrowed our research on the design of a reliability-guaranteed WBAN structure to be specific to long-term monitoring of the vital signs in a periodic and low data-rate generated traffic. The significance of the study was described through the evidence given on the increasing trend in the elderly population around the globe which would demand an easy, long term monitoring of vital signs in the comfort of their homes. One example is the estimated figure in Bloom, et al. (2011), which states that the elderly population of over 65 year-olds only represents 11% of the current population (as at 2011): it is expected to soar to about 22% of the total population by 2050. We discussed how this rapid growth in the elderly population imposes new challenges to the world of traditional healthcare systems. An increase in the cost and demands of the workforce in hospitals is also the result of such a growth which declines in-hospital care systems as a perpetual solution in near future. A shift towards a reliable caring system where subjects can have easy and fast-enough access to their general practitioners in the comfort of their homes or work place, is becoming a necessity in the coming decades.

Extensive research has been done since the emergence of WBAN to address the challenges associated with their unique characteristics (briefly discussed in Sections 1.4.1–1.4.8). Different techniques have been utilized through the layers of the communication stack, each focusing on certain QoS metrics to fulfill, that mostly depend on the target application of a deployed WBAN. In a long-term monitoring of periodic traffic at relatively low data rates, the reliability of the received data mostly relies on the seamless communication links between the sensor nodes and their coordinator. Although energy preservation still remains as one of the challenges, it may not stand as the first crucial task to deal with.Reliability, however, stands as the first important factor when dealing with medical data as the sensors are the only evidence to a patient's current health status and are the only clues a physical practitioner has when monitoring a patient remotely.

In this thesis we have investigated the MAC protocol behavior of the IEEE 802.15.4 standard as the most-used standard for many of the today's WBAN structures. We described how

different MAC techniques are to make different tradeoffs suit the variety of situations that may be caused by the existing traffic type of a network and how they can also be mixed to create possibilities of handling diverse scenarios for different application types.Therefore the evaluation of each MAC technique should be based only on what certain QoS requirements that specific MAC is intended to look after, based on which it has been optimized to meet those certain criteria. Different categories of existing MAC protocols along with their advantages and disadvantages were discussed in Chapter 1. We described how the IEEE 802.15.4 MAC approach uses both contention- and schedule-based techniques in its super frame structure. We narrowed our fuzzy-enabled MAC designed into the beacon-enabled mode of access in IEEE 802.15.4, which uses a CSMA/CA mechanism to organize the access among all the sensor nodes to the shared medium. We then described the inefficiencies related to the backoff scheme of the CSMA/CA and proposed a fair-access mechanism by designing a dynamic backoff window for the random backoff period generator.

In Chapter 2 we investigated the nature of the traffic existing in the WBAN and WSN and discussed how the traffic in traditional wireless sensor networks, where all the sensor nodes in the network co-operate to do a common task, is different from that in wireless body area networks.WBAN structures suffer from heterogeneity in the applications that run on each sensor node (Latré et al., 2011). Each application has its own attributes in terms of data rate, priority, reliability, and delay criticality, which should be taken into consideration in the structure of a protocol. In our MAC algorithm the successful access rate for each sensor node is considered to be our main traffic-indicating parameter, which is derived in the continuing loop of the CSMA/CA performed in the CAP by each node. The motivation for deriving this parameter (which was discussed in Chapter 4) is the aggressive backoff mechanism in the CSMA/CA mechanism of the IEEE 802.15.4, which would sometimes lead to unbalanced waiting times assigned to the nodes of a WBAN. Heterogeneity of the application requirement for each node has motivated this work to treat each node individually, where the chances of transmission for each node are decided based on a fair-channel access mechanism. With a focus on low data-rate applications that do not normally involve sudden high data rates, our approach aimed to increase the reliability of the transmitted sensed data to the coordinator by assigning backoff times to each node that reflect the node’s past trials in having access to the channel. In our approach, no node will be left in severe need of channel access and no node will acquire the channel for a relatively long time.

7.1 **Contributions**

Understanding the generated traffic in a WBAN would help to tune the MAC protocol behavior to efficiently perform in any upcoming traffic situation. The inefficiency of the backoff algorithm in the CSMA/CA mechanism of IEEE 802.15.4 was described through identical and inefficient backoffs in Chapter 2. The event of collision in a contention periodwas described as when the current transmission ends successfully and two (or sometimes more) of the other nodes finish their backoff state *at the same time* and start the CSMA/CA process together. Whilst the standard introduces the backoff as a resolution to less probability of collisions (identical backoffs), it does not necessarily address the efficiency of the generated backoff time in non-identical backoff times, which shifts the problem to inefficient backoffs rather than identical ones. The algorithm mostly relies on incrementing the BE parameter upon a failure or decrementing it when there is a success. It was explained that a failure is not always reported because of an identical backoff, which leads to a collision, but also because a node cannot backoff more than a certain amount of time and none of its previously generated backoffs were good enough to let the device transmit. Identical and inefficient backoff times happen due to a diverse range of reasons that could be protocol or application related. Examples of application-induced causes of such a phenomenon were identified as a high number of nodes, high data rates, and different priorities the sent data have which were considerably associated with the running application. Whilst data rate and the number of nodes involved in a network are good descriptives of an application scenario and its characteristics, in a very realistic implementation, they cannot be considered tunable themselves as they describe an application's requirements. Nevertheless such parameters can be exploited into tuning protocol-related metrics that make CSMA/CA algorithm function for each transmission, which was the main motivation for choosing data rate as one of the inputs of our fuzzy system in the proposed MAC algorithm. However there are other factors that would contribute to the problem of inefficient backoff-period generation which are more protocol related. We discussed a variety of such parameters through an extensive literature review in Chapter 3 and had an indepth look into the involved parameters in the CFP and CAP of a super frame structure in beacon-enabled mode of access in IEEE 802.15.4 in Section 3.1. Studying the exploited parameters in the approaches taken by researchers in Chapter 3 led us to the selection of the second input of our fuzzy algorithm, which was an average value reflecting the successful rate of a sensor node in having access to the channel. We then designed a fuzzy system that could

feed in the effect of these two parameters and yield to a moderate tuning of the backoff window to improve fairness among nodes during their contention to the channel.

We proposed our fuzzy-enabled MAC for low to moderate periodic traffic behavior with reliability in mind as the main concern. A fair access to the shared wireless channel contributes greatly to the overall system's reliability performance. This fairness makes the channel access possible for all the heterogeneous sensor nodes in the network while examining their individual data rate value and also their success channel access rate. The main contributions of the proposed fuzzy-enabled MAC algorithm for the discussedproblems associated with IEEE 802.15.4 backoff scheme are:

- Higher level of reliability
- Lower level of delay
- Having no effect on the level of energy consumption for the fuzzy algorithm running on the sensor's side.

As the applications for a WBAN deal with humanlife, more levels of reliability and timely transmission of sensory data are needed that will conform to all these requirements. A timely transmission during a super frame in a beacon-enabled MAC refers to the need to assign a time slot as efficiently as possible. In our proposed MAC approach, this criteria is fulfilled by analysing the network traffic, considering the combined effect of two different parameters that would reflect how fairly the channel has been assigned to each sensor device. A dynamic, reliable, and traffic-adaptive MAC algorithm for wireless body area networks is presented in this thesis, in which a fuzzy-logic system decides about the length of the backoff period for each node in the network as the nodes contend to relay their data over the channel. The decision made by the fuzzy-logic system will determinethe maximum bound on the backoff windowin the beacon-enabled mode of IEEE 802.15.4 to moderate the waiting-time period for each node during their backoff procedure. The proposed methodology is first implemented and tested through simulations in Castalia, discussed in Chapter 5, and is later deployed on a real test bed (SHIMMER sensor platforms) for further analysis and investigations.

The results of the simulations carried out in Chapter 5 promise a better reliability level in terms of the average packets received at the coordinator device when nodes take advantage of the fuzzy-enabled MAC in their CSMA/CA mechanism. It has also been tested, both through

simulations and on the real SHIMMER sensor platforms, that the complexity of the proposed fuzzy algorithm does no harm to the lifetime of the battery when compared to a sensor node operating on the original IEEE 802.15.4 MAC protocol. A simplifying technique has also been incorporated into our fuzzy-enabled MAC that further ensures the efficiency of the proposed MAC in terms of the consumed energy.

7.2 Future Directions

Further research can be taken on testing other parameters as the inputs of the designed fuzzy algorithm. The chosen parameters, however, should reflect the application requirements. For example, priority can be chosen as a parameter of interest if the physiological sensor nodes in the WBAN are time critical and have been assigned applications related to acute health monitoring for patients suffering from epilepsy or severe heart diseases.

The energy measurements of the real implementations on SHIMMER sensor platforms at MAC layer have not been carried out as the proposed fuzzy-enabled MAC was designed to address the reliability for long-term monitoring applications of low data-rate nature and was not intended to improve energy efficiency. Only the computational complexity of the fuzzy algorithm has been tested to prove its similar energy spending behavior in contrast to IEEE 802.15.4 MAC. As discussed earlier in Section 1.4.8, in many of the reviewed pilot studies or short-term real implementations of a WBAN in hospital environments, changing and recharging the batteries of medical sensors remains one of the medical personnel's responsibilities. Although the research direction for many of the healthcare and sport applications is towards having a WBAN that would possibly need no change or recharging of the batteries, even in most of today's implementations of WBAN for medical use cases the batteries of the medical sensors will need to be changed at least once or twice a week, depending on the physiological activity they monitor. We therefore only performed the reliability evaluations in our testbed experiments, which proved a better percentage compared to the IEEE 802.15.4 MAC approach.

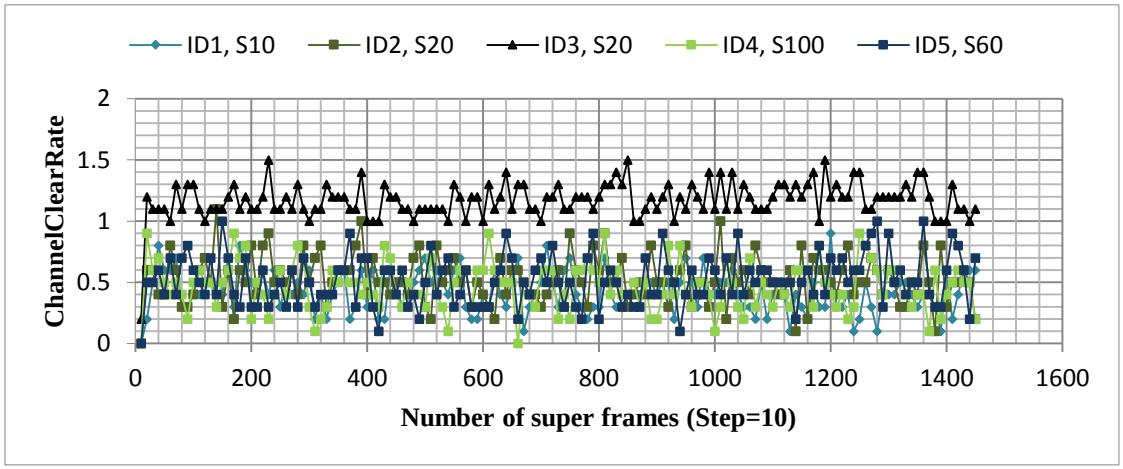
Although the data rates assigned to the RFD nodes in our star topology illustrate different values to each sensor node, the values do not change over the course of a simulation run or during their transmission in our testbed experiments. This does not mean that our algorithm is not capable of addressing dynamic data rates but we did not implement the data rate to be

dynamic for each individual node during any of the simulation or experimental studies in this thesis: these can be carried out in the future.

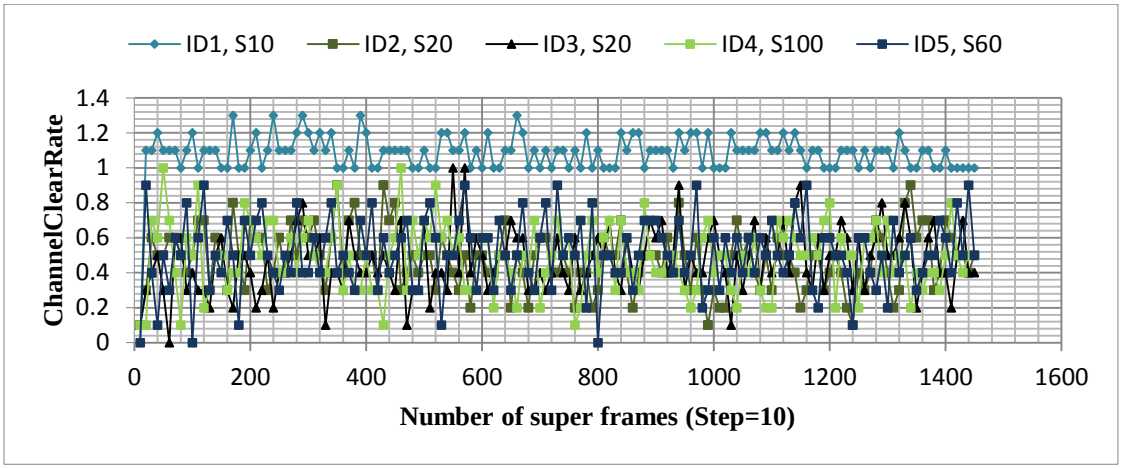
Although we followed the channel and mobility models existing in Castalia simulator, more sophisticated and close-to-reality models can be incorporated in the future as they emerge and become available in existing simulation tools.

APPENDIX I: *Channel$_{Clear}$Rate* Value Observations

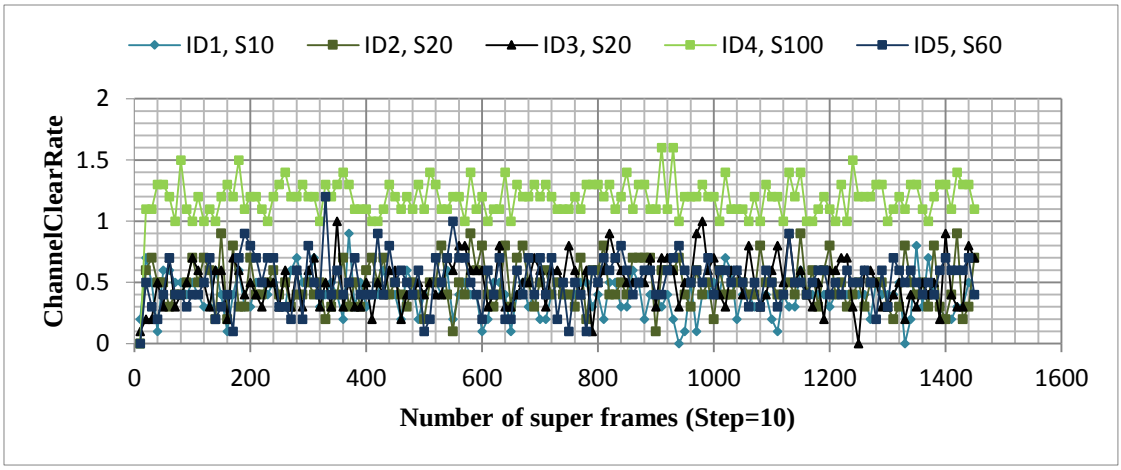
The figures below show the results of the six simulations done (out of 10) for the five existing nodes in our typical body area network and the values of the *Channel$_{Clear}$Rate* in percentage mounted on the Y axis of the graphs. These are the individual diagrams, the averages of which appear in Figures 4-5 to 4-9 in Section 4.2. The X axis is simply showing the number of the super frames ranging from zero (which is the beginning of the simulation) up to 1500 (as the very last super frame) with steps of *n* (for every *n* super frame). Based on these six repetitions of the simulations (with only one constant set of data rates), we can now make a fair decision for the min and max values of the first input of our fuzzy system. The strokes in each diagram are denoted by the node's ID and its assigned sample rate is denoted by S and the index showing the sample rate. For example 'ID1, S10' means Node1 with a sample rate of 10 packets per second.



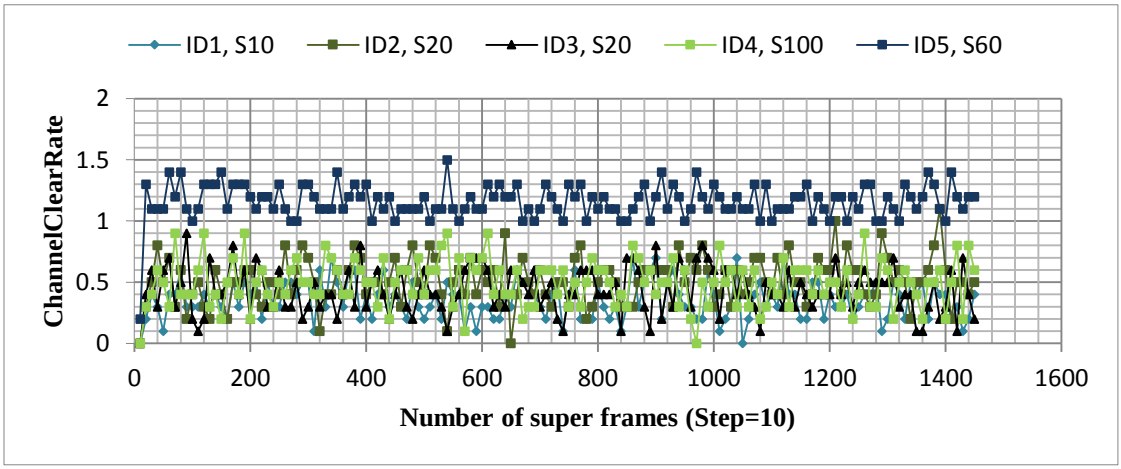
Figure_Apx1: Simulation run1, *Channel$_{Clear}$Rate*, every 10 super frames



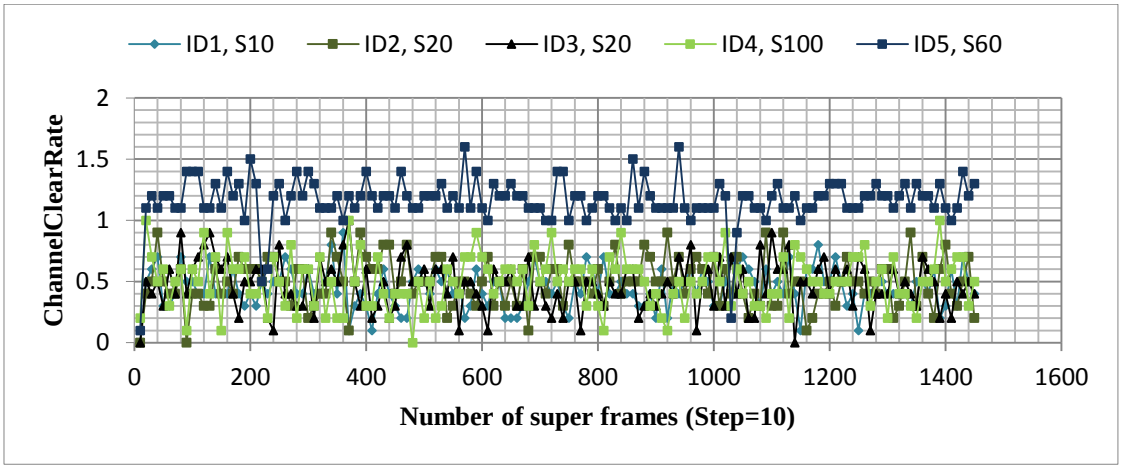
Figure_Apx2: Simulation run2, *Channel$_{ClearRate}$*, every 10 super frames



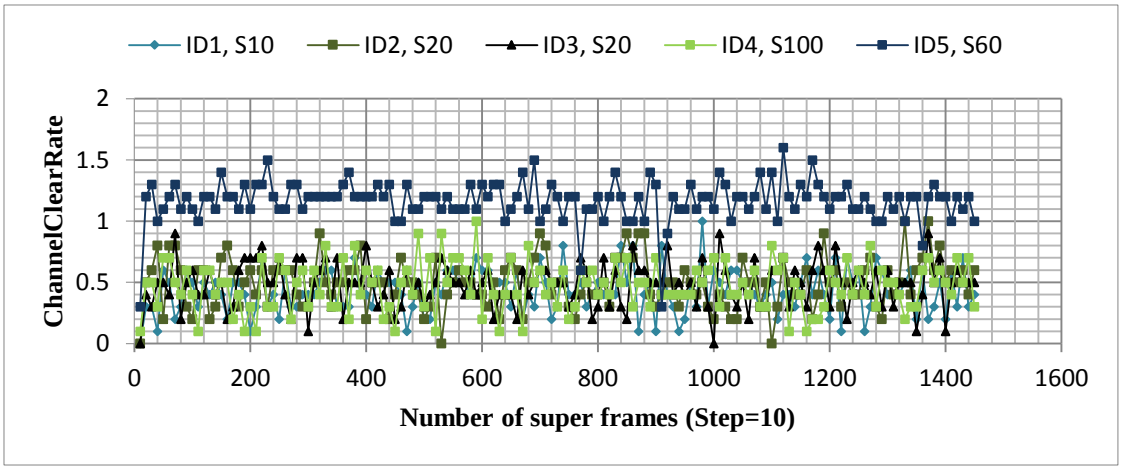
Figure_Apx3: Simulation run3, *Channel$_{ClearRate}$*, every 10 super frames



Figure_Apx4: Simulation run4, *Channel$_{ClearRate}$*, every 10 super frames

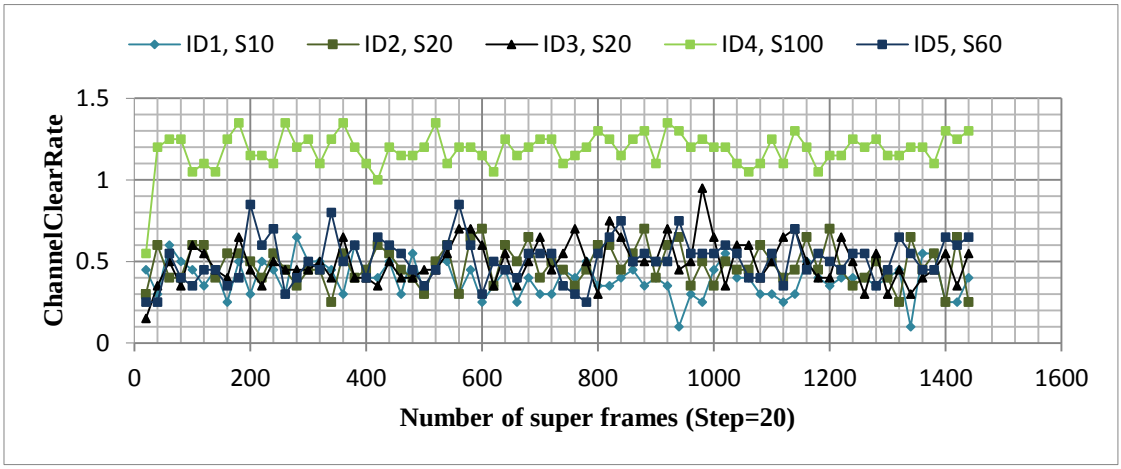


Figure_Apx5: Simulation run5, *Channel$_{ClearRate}$*, every 10 super frames

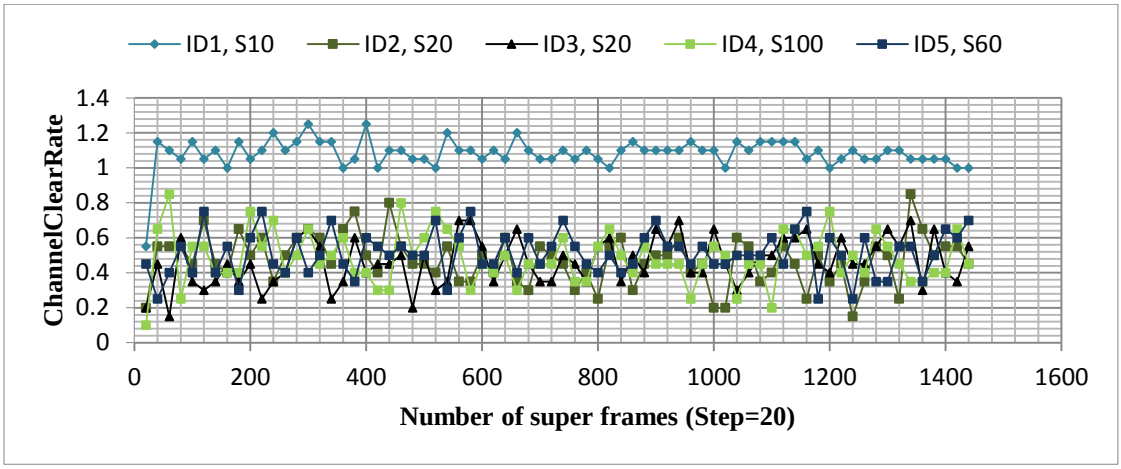


Figure_Apx6: Simulation run6, *Channel$_{ClearRate}$*, every 10 super frames

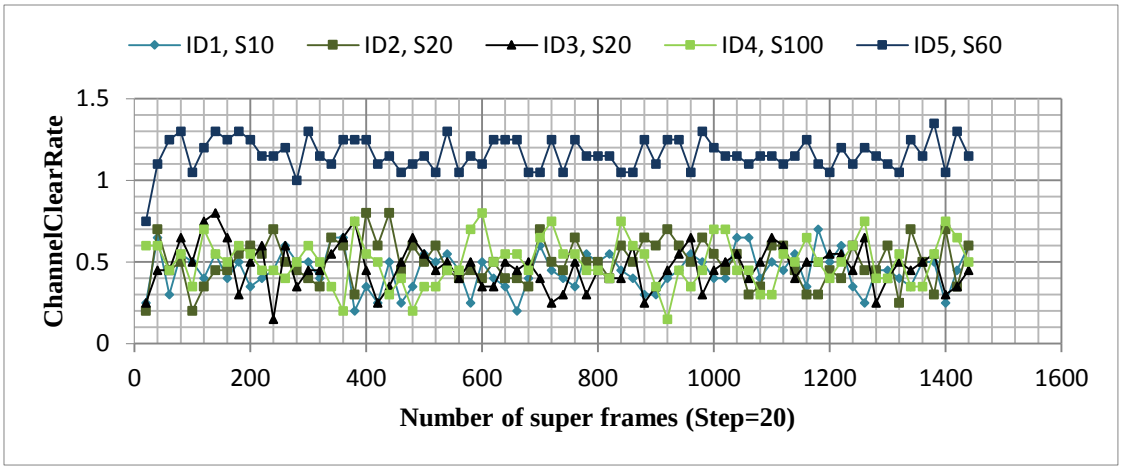
---



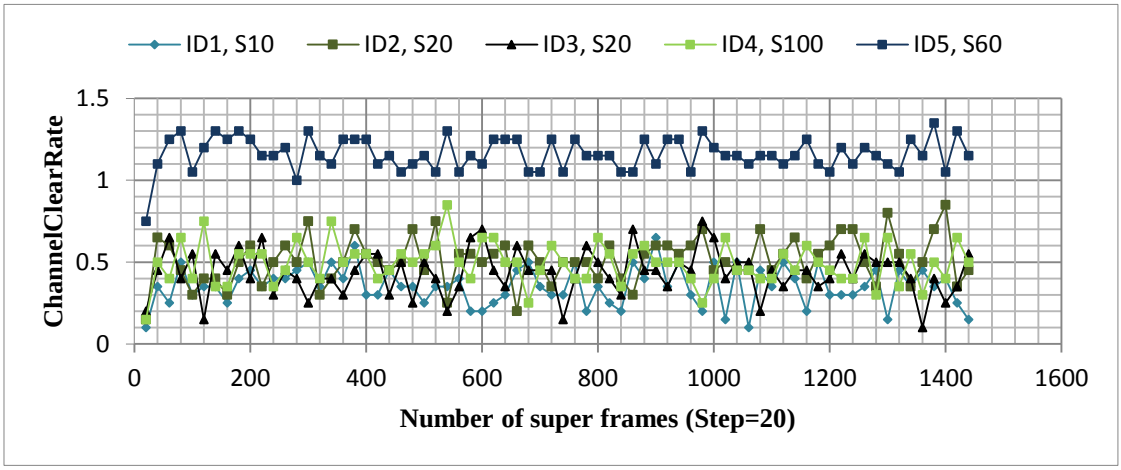
Figure_Apx7: Simulation run1, *Channel$_{ClearRate}$*, every 20 super frames



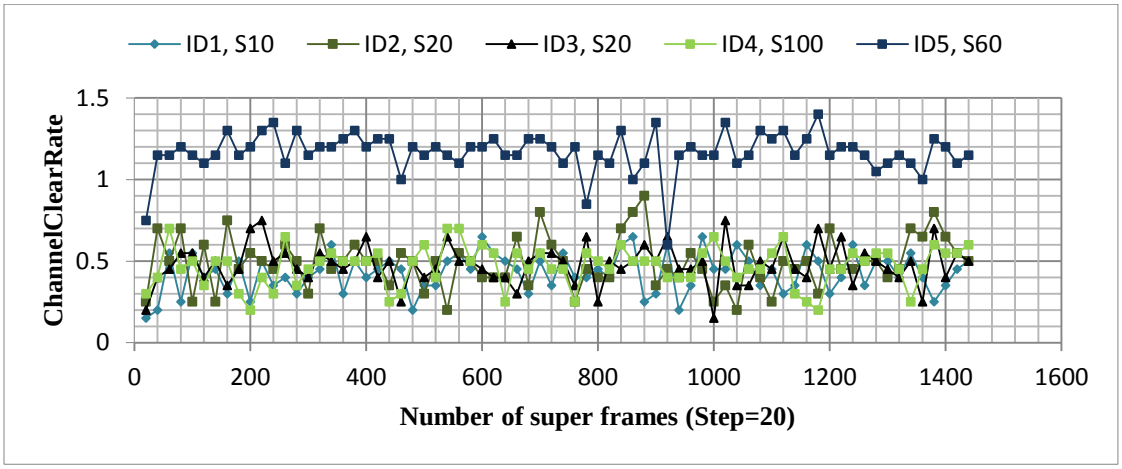
Figure_Apx8: Simulation run2, *Channel$_{Clear}$Rate*, every 20 super frames



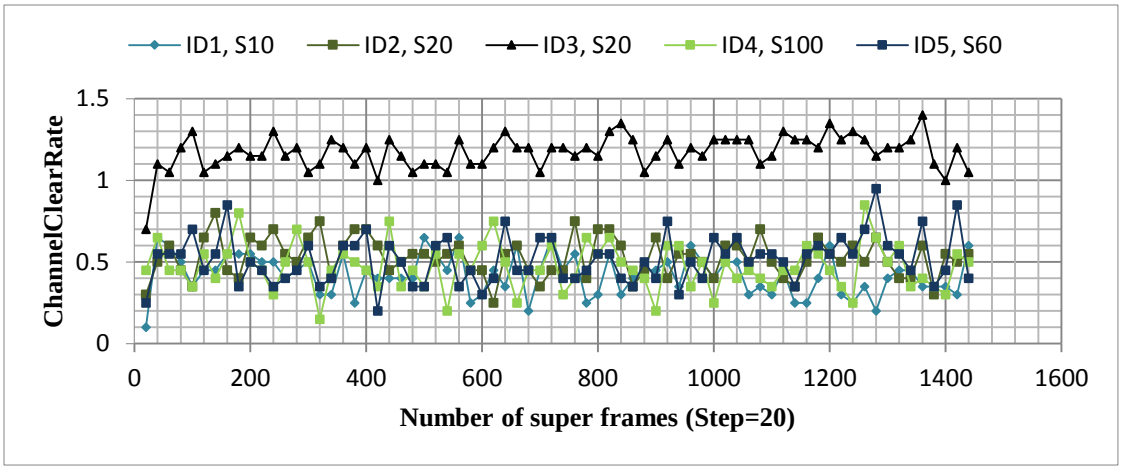
Figure_Apx9: Simulation run3, *Channel$_{Clear}$Rate*, every 20 super frames



Figure_Apx10: Simulation run4, *Channel$_{Clear}$Rate*, every 20 super frames

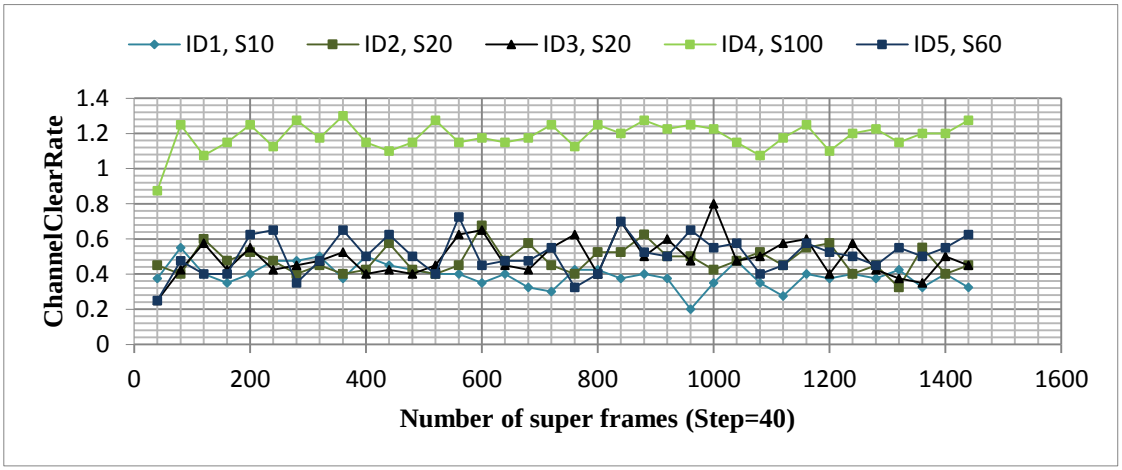


Figure_Apx11: Simulation run5, *Channel$_{ClearRate}$*, every 20 super frames

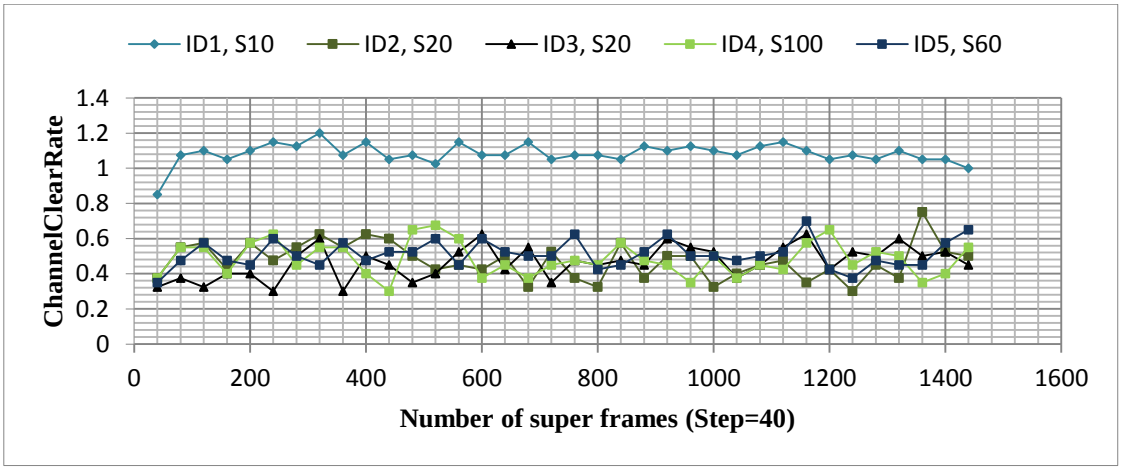


Figure_Apx12: Simulation run6, *Channel$_{ClearRate}$*, every 20 super frames

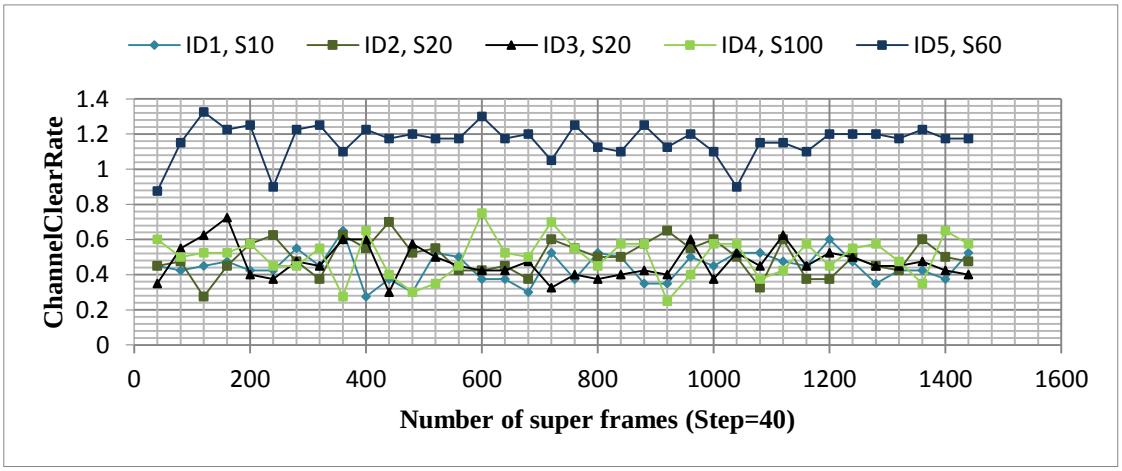
---



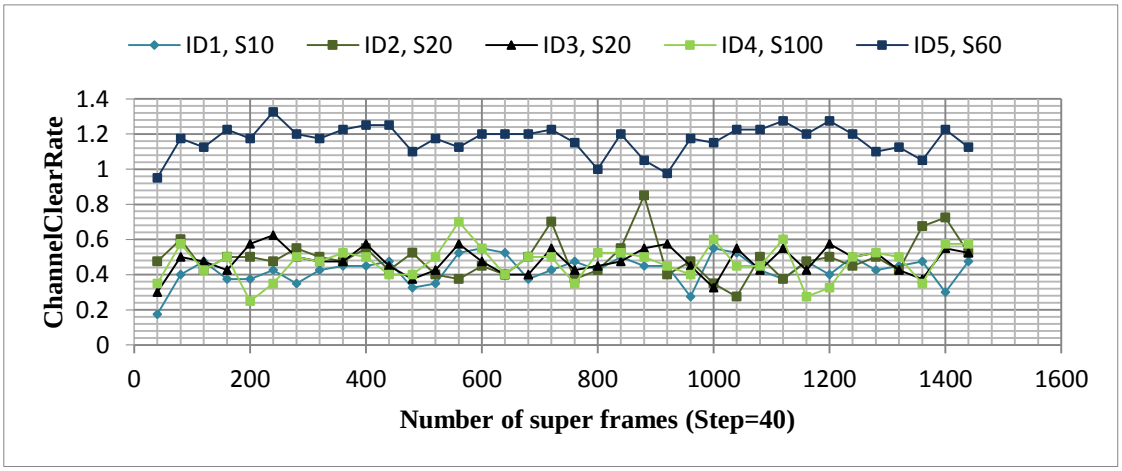
Figure_Apx13: Simulation run1, *Channel$_{ClearRate}$*, every 40 super frames



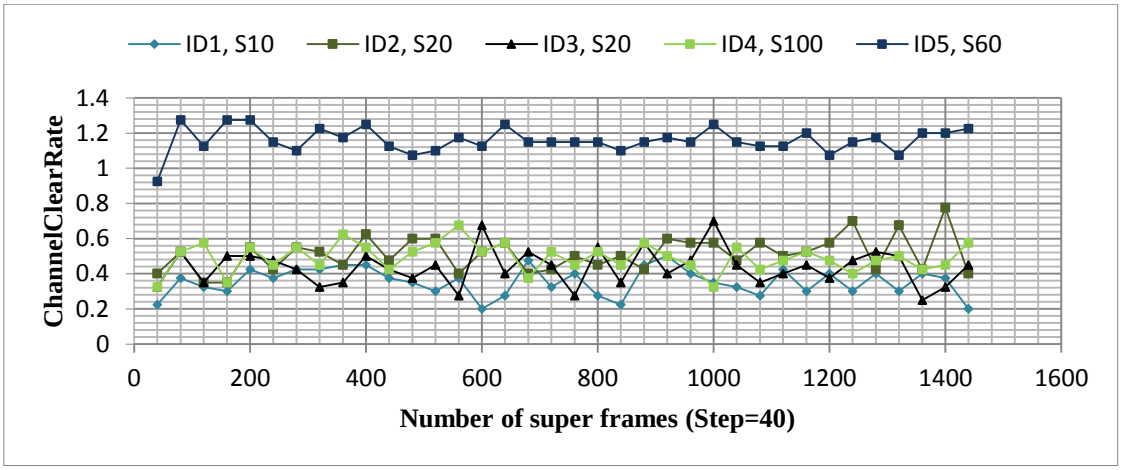
Figure_Apx14: Simulation run2, *Channel$_{ClearRate}$*, every 40 super frames



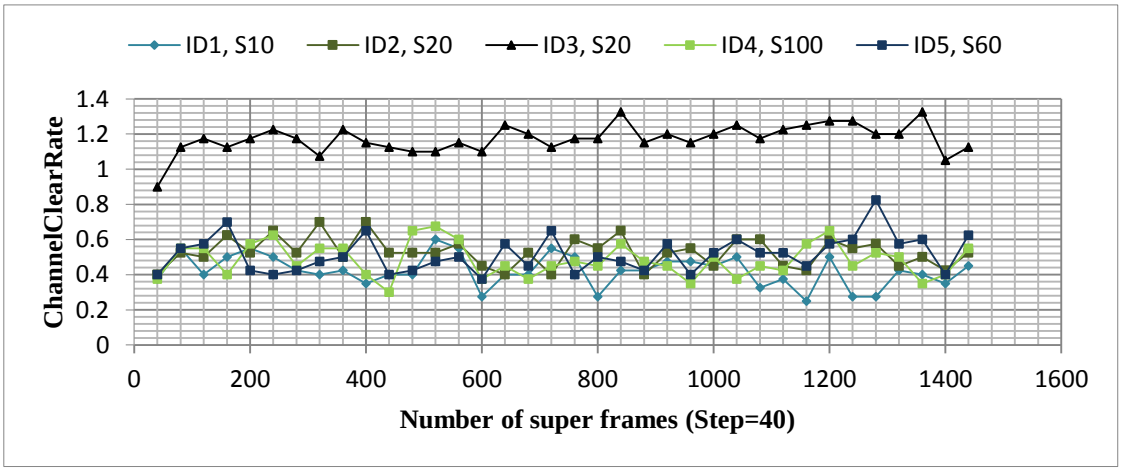
Figure_Apx15: Simulation run3, *Channel$_{ClearRate}$*, every 40 super frames



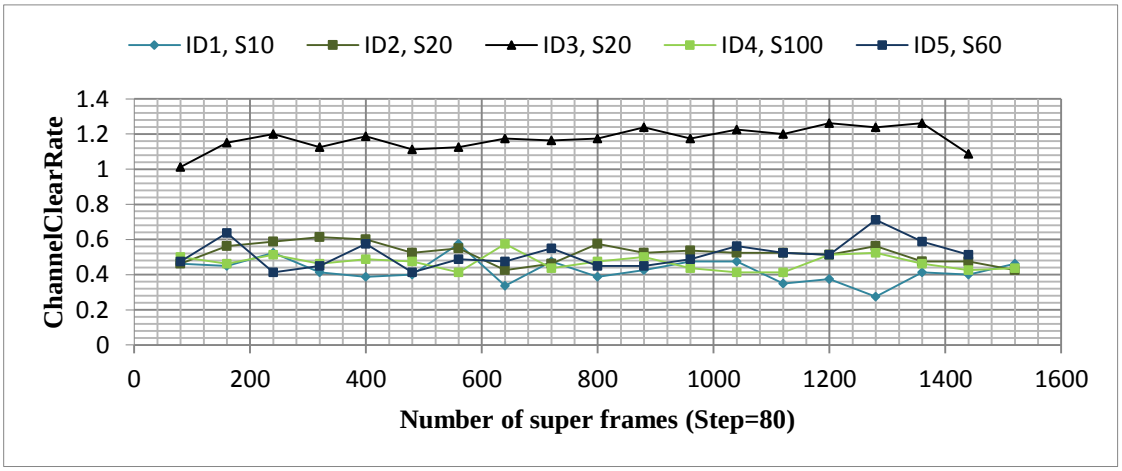
Figure_Apx16: Simulation run4, *Channel$_{ClearRate}$*, every 40 super frames



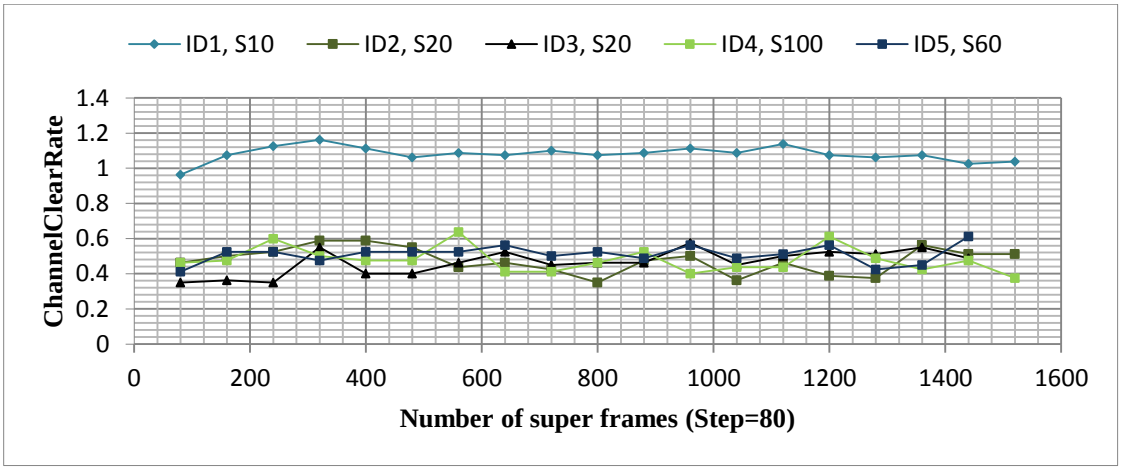
Figure_Apx17: Simulation run5, *Channel$_{ClearRate}$*, every 40 super frames



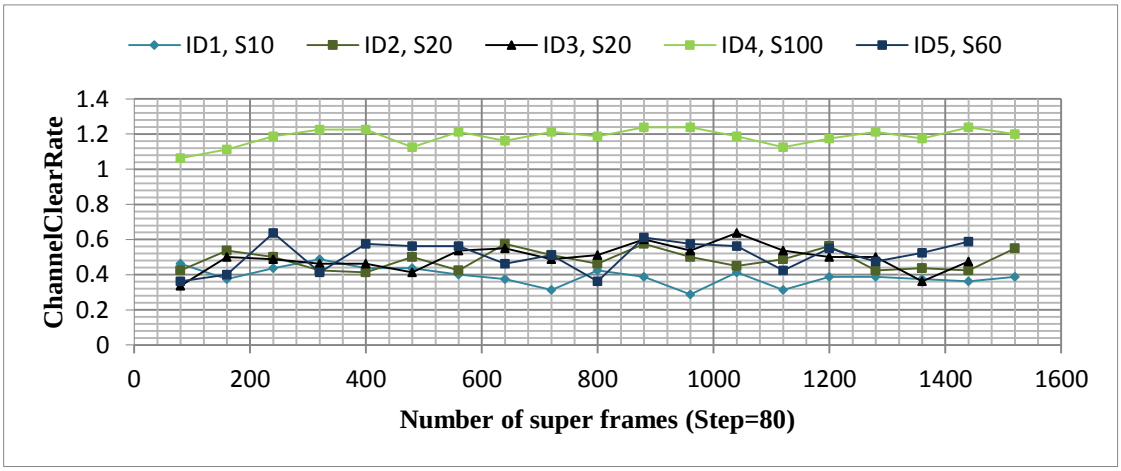
Figure_Apx18: Simulation run6, *Channel$_{ClearRate}$*, every 40 super frames



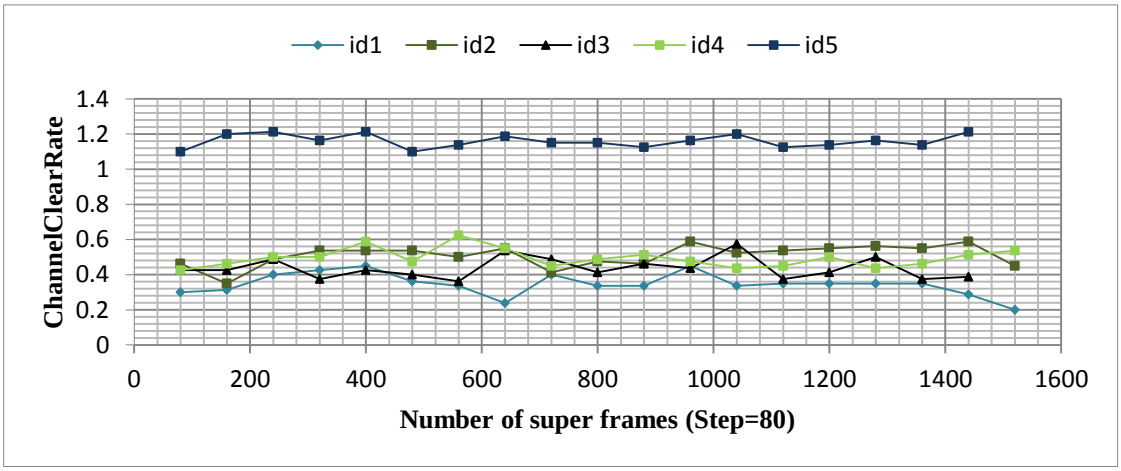
Figure_Apx19: Simulation run1, *Channel$_{ClearRate}$*, every 80 super frames



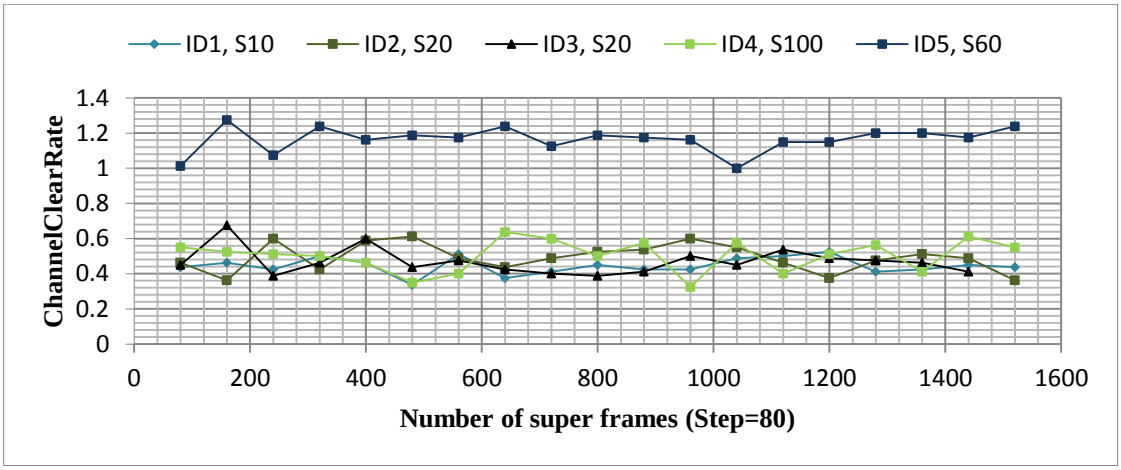
Figure_Apx20: Simulation run2, *Channel$_{ClearRate}$*, every 80 super frames



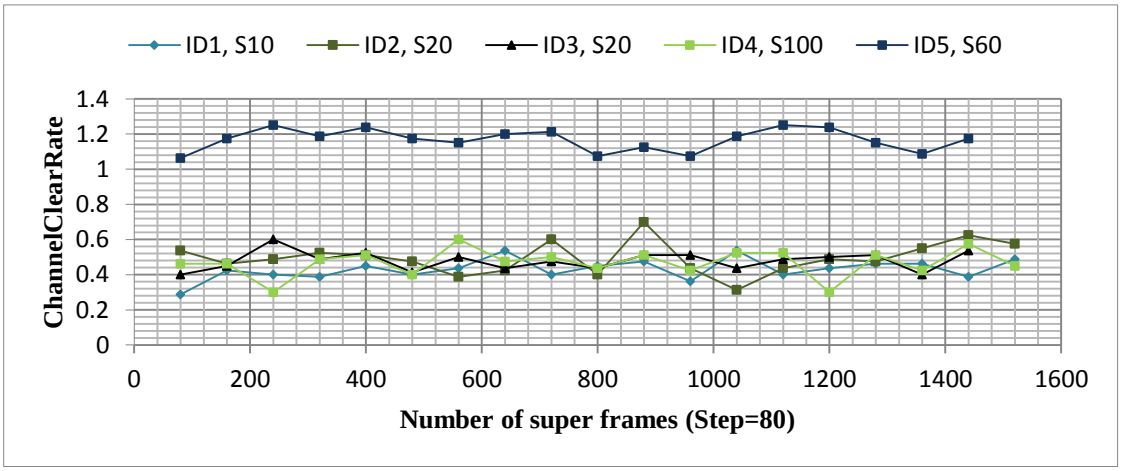
Figure_Apx21: Simulation run3, *Channel$_{ClearRate}$*, every 80 super frames



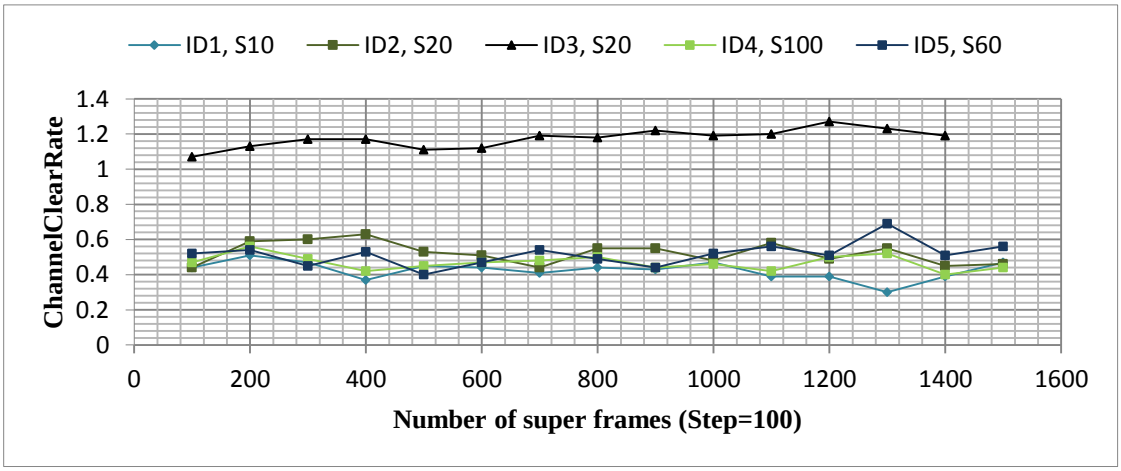
Figure_Apx22: Simulation run4, *Channel$_{ClearRate}$*, every 80 super frames



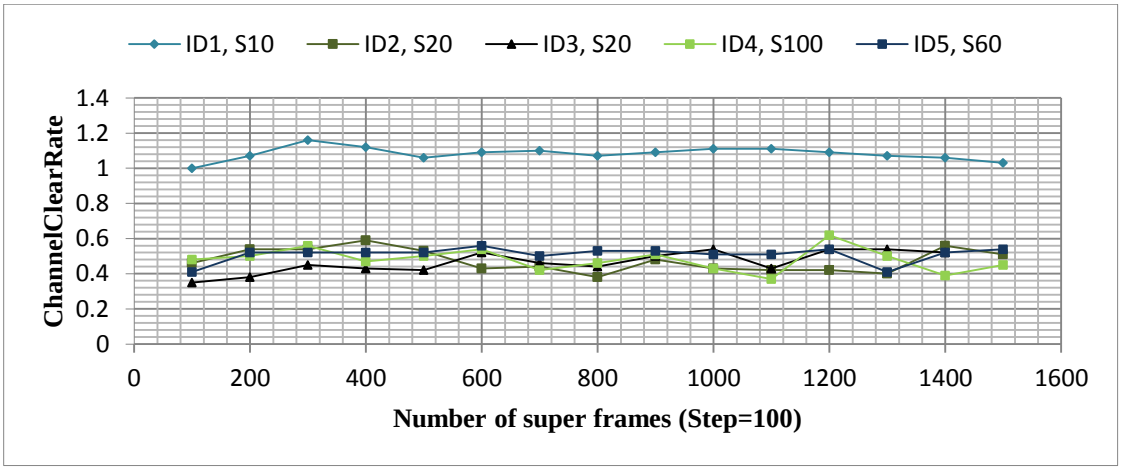
Figure_Apx23: Simulation run5, *Channel$_{ClearRate}$*, every 80 super frames



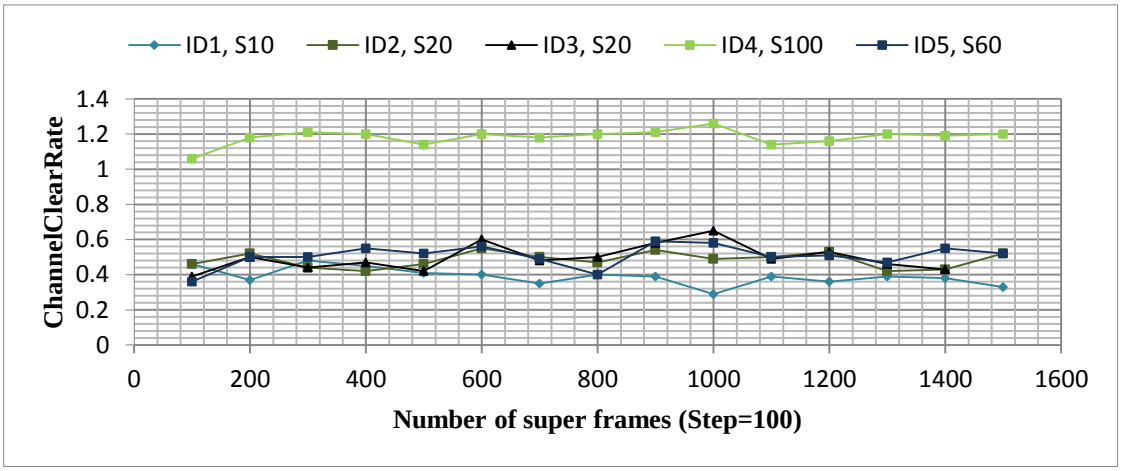
Figure_Apx24: Simulation run6, *Channel$_{ClearRate}$*, every 80 super frames



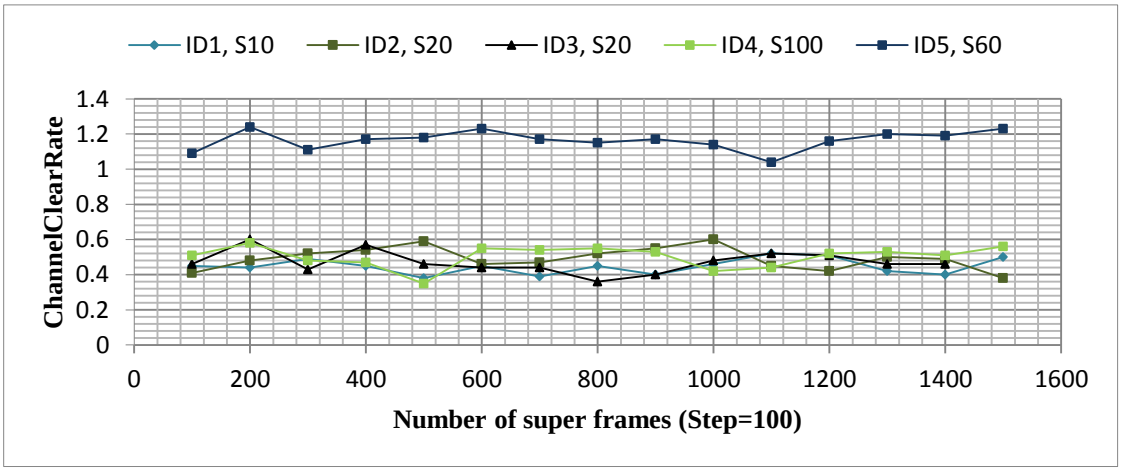
Figure_Apx25: Simulation run1, *Channel$_{ClearRate}$*, every 100 super frames



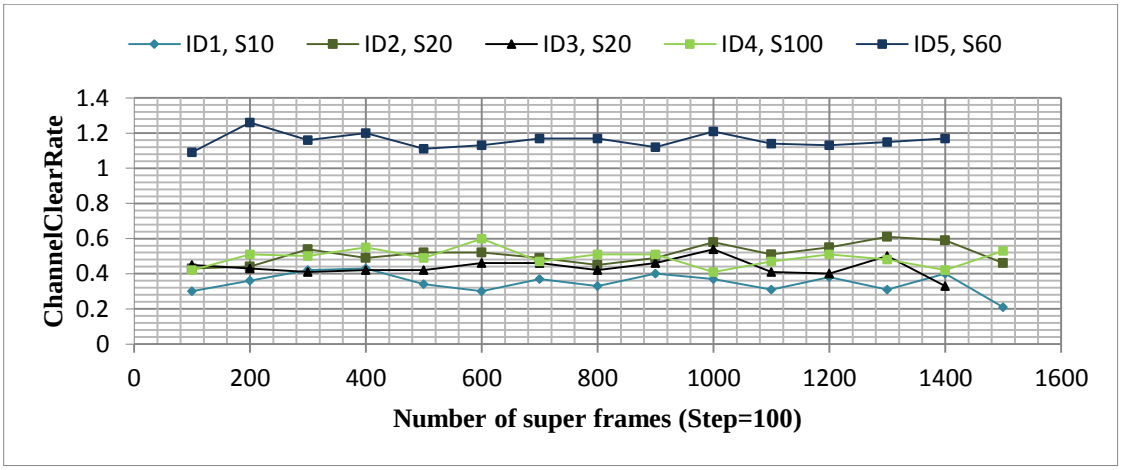
Figure_Apx26: Simulation run2, *Channel$_{ClearRate}$*, every 100 super frames



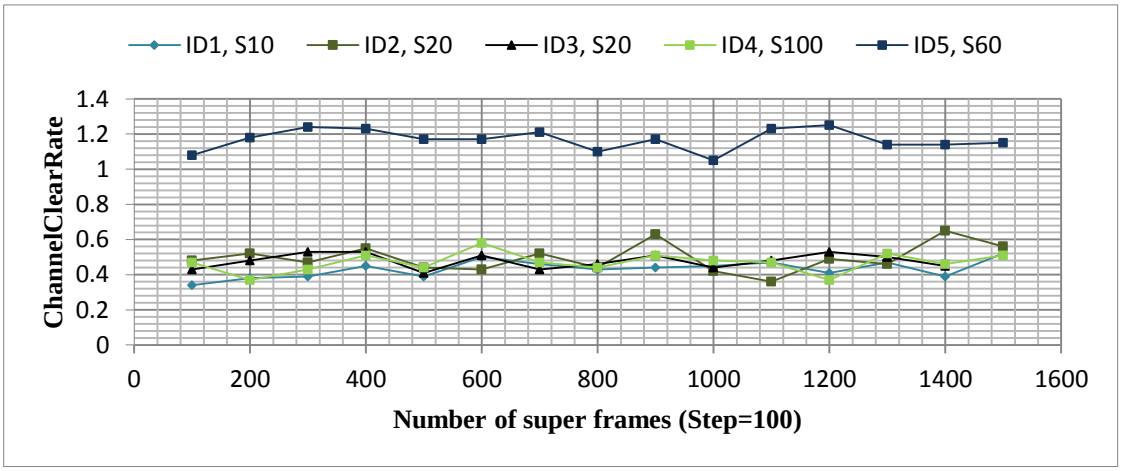
Figure_Apx27: Simulation run3, *Channel$_{ClearRate}$*, every 100 super frames



Figure_Apx28: Simulation run4, *Channel$_{ClearRate}$*, every 100 super frames



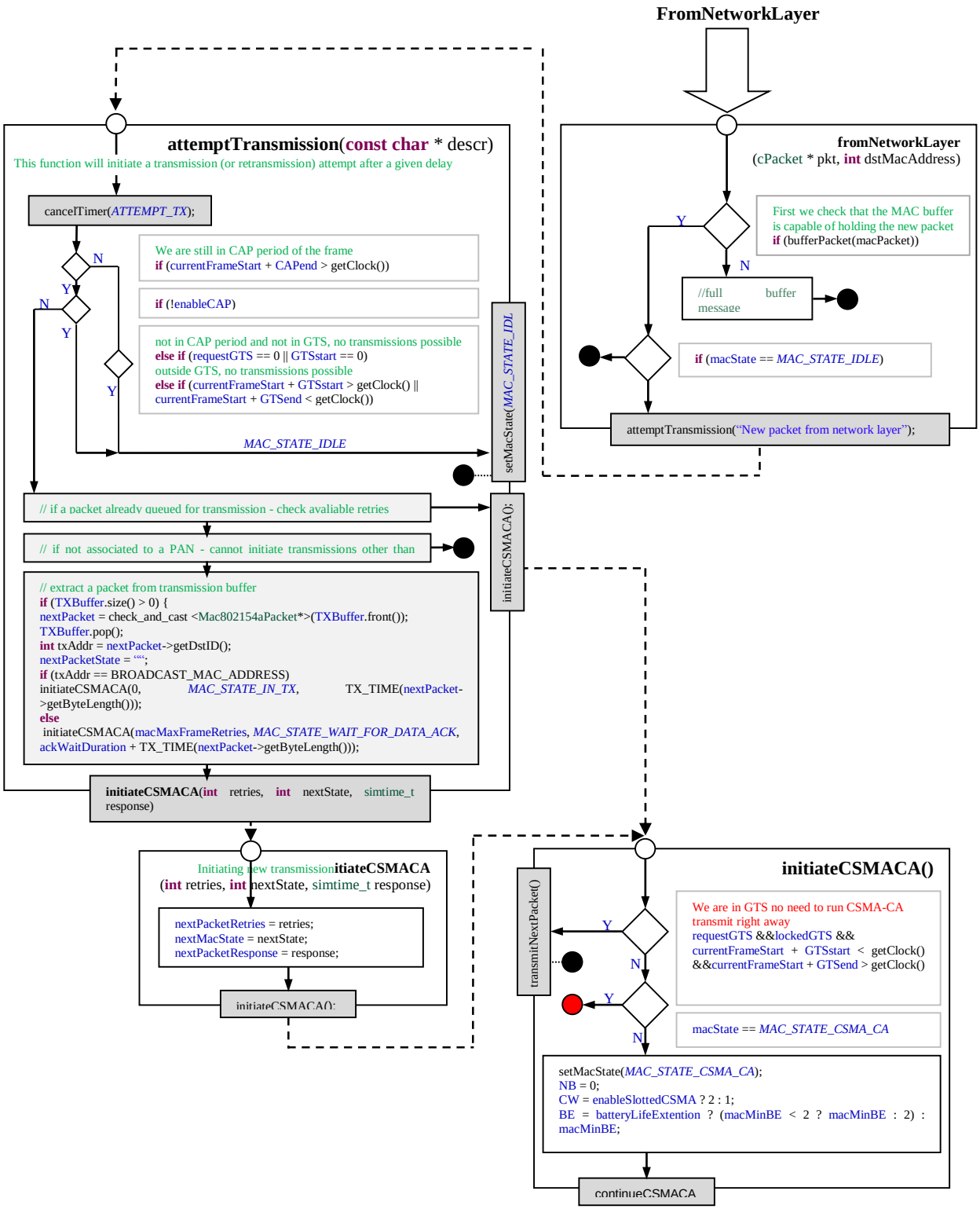
Figure_Apx29: Simulation run5, *Channel$_{ClearRate}$*, every 100 super frames

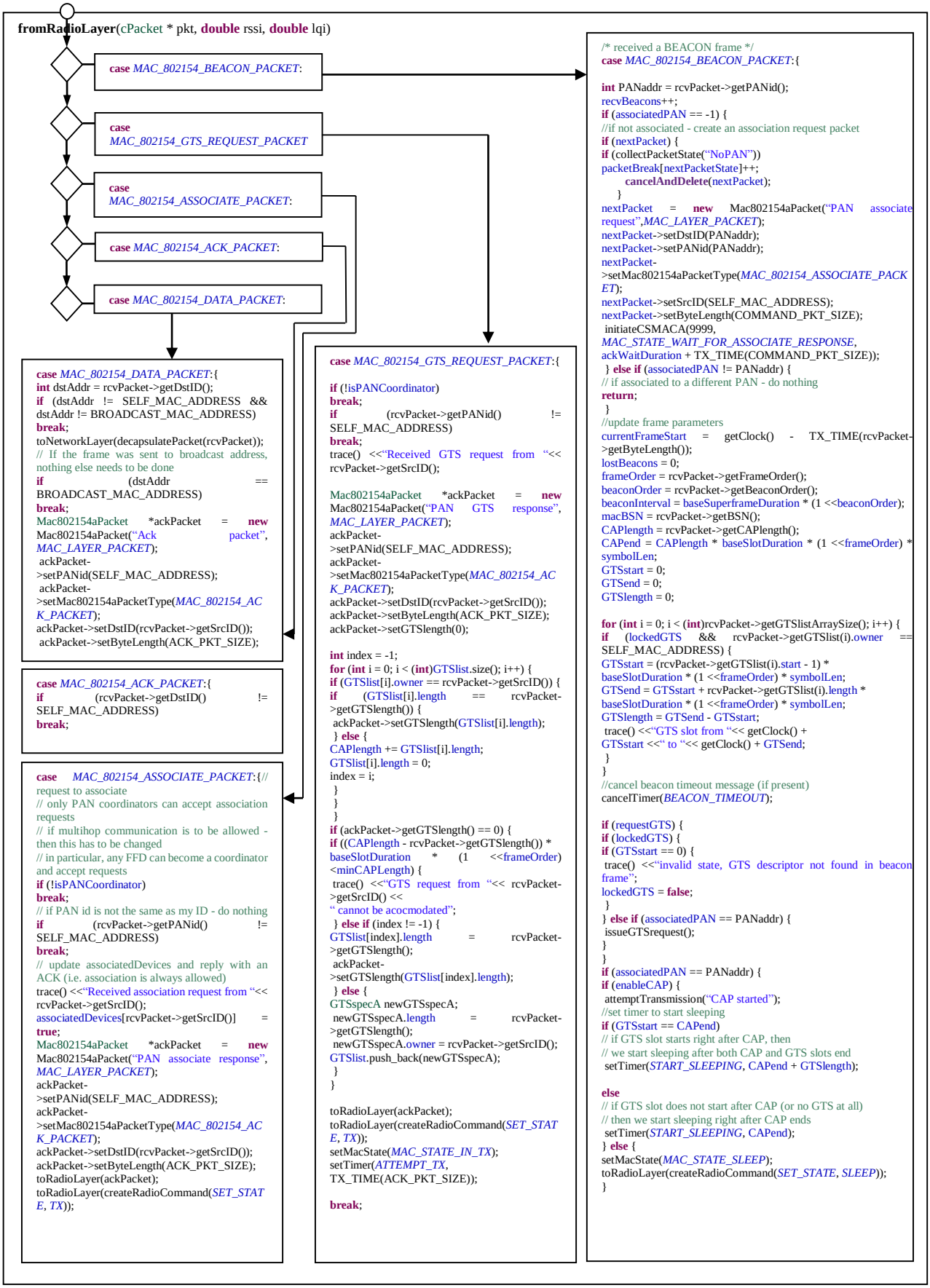


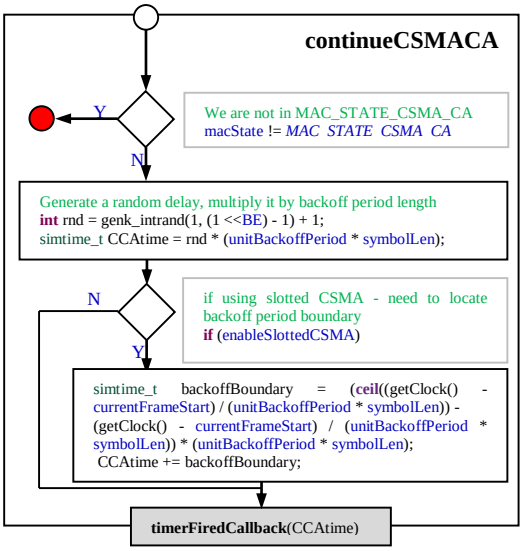
Figure_Apx30: Simulation run6, *Channel$_{ClearRate}$*, every 100 super frames

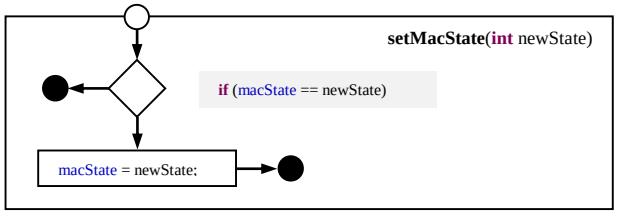
APPENDIX II: The MAC 802.15.4 Class in Castalia

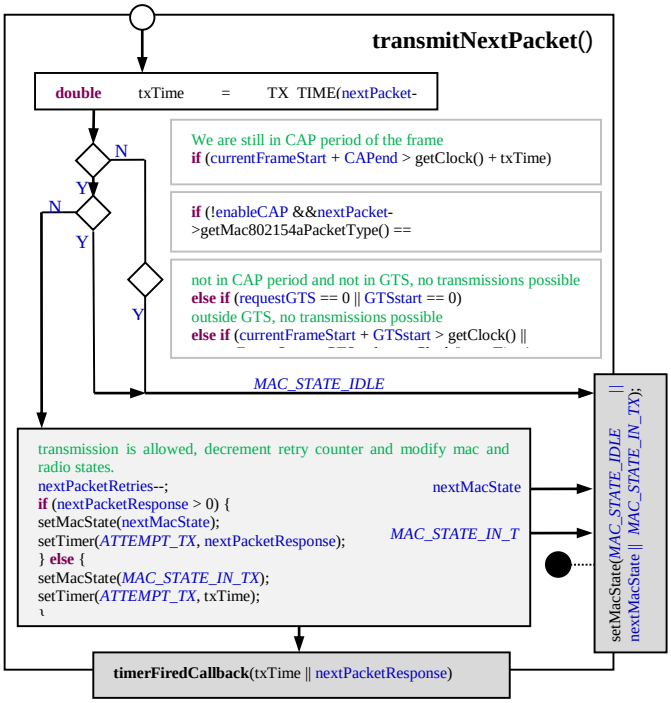
Narrowing down to the details of each step in our algorithm needs a thorough understanding of the IEEE 802.15.4 standard to exactly ensure the correct placement of the algorithm steps and the necessary changes to make in the related model built in the intended simulator to use, Castalia 3.2. Since the documentation of the standard's code in Castalia mostly relies on the comments inside the code, an exact match between the variables and primitives defined in both might not be possible, however Castalia's implementation of IEEE 802.15.4 is very easy to comprehend. It is indeed the best idea to start with the fundamentals of the standard in its draft and match them with the relevant parts of our implementation in Castalia later on. Here we have shown the block diagram of the IEEE 802.15.4 MAC structure according to its implementation in Castalia. This will give the reader a good reference to have a better understanding of the related pseudo-codes discussed in the thesis when implementing the fuzzy algorithm in the CSMA/CA module.

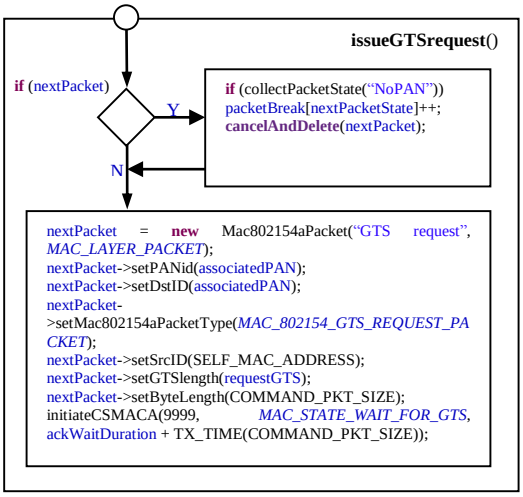

FromNetworkLayer
attemptTransmission(const char * descr)
This function will initiate a transmission (or retransmission) attempt after a given delay
cancelTimer(ATTEMPT_TX);
We are still in CAP period of the frame
if (currentFrameStart + CAPend > getClock())
if (!enableCAP)
not in CAP period and not in GTS, no transmissions possible
else if (requestGTS == 0 || GTSstart == 0)
outside GTS, no transmissions possible
else if (currentFrameStart + GTSstart > getClock() ||
currentFrameStart + GTSend < getClock())
MAC_STATE_IDLE
setMacState(MAC_STATE_IDL
// if a packet already queued for transmission - check avaliable retries
// if not associated to a PAN - cannot initiate transmissions other than
// extract a packet from transmission buffer
if (TXBuffer.size() > 0) {
nextPacket = check_and_cast <Mac802154aPacket*>(TXBuffer.front());
TXBuffer.pop();
int txAddr = nextPacket->getDstID();
nextPacketState = "";
if (txAddr == BROADCAST_MAC_ADDRESS)
initiateCSMACA(0, MAC_STATE_IN_TX, TX_TIME(nextPacket-
>getByteLength()));
else
initiateCSMACA(macMaxFrameRetries, MAC_STATE_WAIT_FOR_DATA_ACK,
ackWaitDuration + TX_TIME(nextPacket->getByteLength()));
initiateCSMACA(int retries, int nextState, simtime_t
response)
initiateCSMACA
(int retries, int nextState, simtime_t response)
nextPacketRetries = retries;
nextMacState = nextState;
nextPacketResponse = response;
initiateCSMACA();
fromNetworkLayer
(cPacket * pkt, int dstMacAddress)
First we check that the MAC buffer
is capable of holding the new packet
if (bufferPacket(macPacket))
//full buffer
message
if (macState == MAC_STATE_IDLE)
attemptTransmission("New packet from network layer");
initiateCSMACA()
transmitNextPacket()
We are in GTS no need to run CSMA-CA
transmit right away
requestGTS &&lockedGTS &&
currentFrameStart + GTSstart < getClock()
&&currentFrameStart + GTSend > getClock()
macState == MAC_STATE_CSMA_CA
setMacState(MAC_STATE_CSMA_CA);
NB = 0;
CW = enableSlottedCSMA ? 2 : 1;
BE = batteryLifeExtention ? (macMinBE < 2 ? macMinBE : 2) :
macMinBE;
continueCSMACA

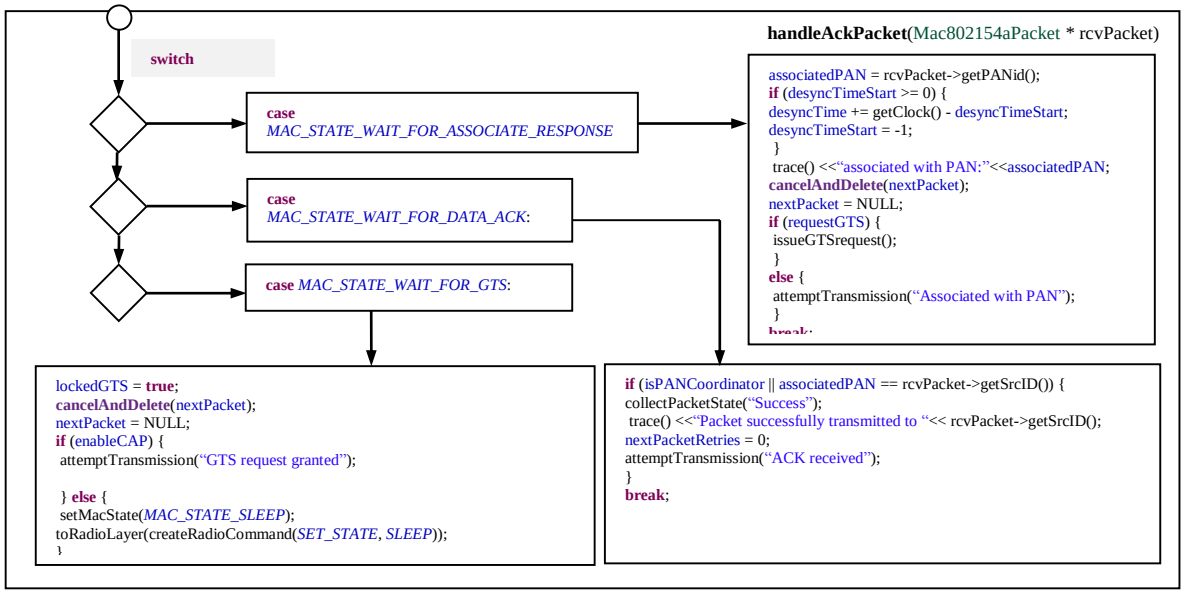

fromRadioLayer(cPacket * pkt, double rssi, double lqi)
case MAC_802154_BEACON_PACKET:
case MAC_802154_GTS_REQUEST_PACKET
case MAC_802154_ASSOCIATE_PACKET:
case MAC_802154_ACK_PACKET:
case MAC_802154_DATA_PACKET:
/* received a BEACON frame */
case MAC_802154_BEACON_PACKET:{
int PANaddr = rcvPacket->getPANid();
recvBeacons++;
if (associatedPAN == -1) {
//if not associated - create an association request packet
if (nextPacket) {
if (collectPacketState("NoPAN"))
packetBreak[nextPacketState]++;
cancelAndDelete(nextPacket);
}
nextPacket = new Mac802154aPacket("PAN associate request",MAC_LAYER_PACKET);
nextPacket->setDstID(PANaddr);
nextPacket->setPANid(PANaddr);
nextPacket->setMac802154aPacketType(MAC_802154_ASSOCIATE_PACKET);
nextPacket->setSrcID(SELF_MAC_ADDRESS);
nextPacket->setByteLength(COMMAND_PKT_SIZE);
initiateCSMACA(9999, MAC_STATE_WAIT_FOR_ASSOCIATE_RESPONSE, ackWaitDuration + TX_TIME(COMMAND_PKT_SIZE));
} else if (associatedPAN != PANaddr) {
// if associated to a different PAN - do nothing
return;
}
//update frame parameters
currentFrameStart = getClock() - TX_TIME(rcvPacket->getByteLength());
lostBeacons = 0;
frameOrder = rcvPacket->getFrameOrder();
beaconOrder = rcvPacket->getBeaconOrder();
beaconInterval = baseSuperframeDuration * (1 <<beaconOrder);
macBSN = rcvPacket->getBSN();
CAPlength = rcvPacket->getCAPlength();
CAPend = CAPlength * baseSlotDuration * (1 <<frameOrder) * symbolLen;
GTSstart = 0;
GTSend = 0;
GTSlength = 0;
for (int i = 0; i < (int)rcvPacket->getGTSlistArraySize(); i++) {
if (lockedGTS && rcvPacket->getGTSlist(i).owner == SELF_MAC_ADDRESS) {
GTSstart = (rcvPacket->getGTSlist(i).start - 1) * baseSlotDuration * (1 <<frameOrder) * symbolLen;
GTSend = GTSstart + rcvPacket->getGTSlist(i).length * baseSlotDuration * (1 <<frameOrder) * symbolLen;
GTSlength = GTSend - GTSstart;
trace() <<"GTS slot from "<< getClock() + GTSstart <<" to "<< getClock() + GTSend;
}
}
//cancel beacon timeout message (if present)
cancelTimer(BEACON_TIMEOUT);
if (requestGTS) {
if (lockedGTS) {
if (GTSstart == 0) {
trace() <<"invalid state, GTS descriptor not found in beacon frame";
lockedGTS = false;
}
} else if (associatedPAN == PANaddr) {
issueGTSrequest();
}
}
if (associatedPAN == PANaddr) {
if (enableCAP) {
attemptTransmission("CAP started");
//set timer to start sleeping
if (GTSstart == CAPend)
// if GTS slot starts right after CAP, then
// we start sleeping after both CAP and GTS slots end
setTimer(START_SLEEPING, CAPend + GTSlength);
else
// if GTS slot does not start after CAP (or no GTS at all)
// then we start sleeping right after CAP ends
setTimer(START_SLEEPING, CAPend);
} else {
setMacState(MAC_STATE_SLEEP);
toRadioLayer(createRadioCommand(SET_STATE, SLEEP));
}
case MAC_802154_GTS_REQUEST_PACKET:{
if (!isPANCoordinator)
break;
if (rcvPacket->getPANid() != SELF_MAC_ADDRESS)
break;
trace() <<"Received GTS request from "<< rcvPacket->getSrcID();
Mac802154aPacket *ackPacket = new Mac802154aPacket("PAN GTS response", MAC_LAYER_PACKET);
ackPacket->setPANid(SELF_MAC_ADDRESS);
ackPacket->setMac802154aPacketType(MAC_802154_ACK_PACKET);
ackPacket->setDstID(rcvPacket->getSrcID());
ackPacket->setByteLength(ACK_PKT_SIZE);
ackPacket->setGTSlength(0);
int index = -1;
for (int i = 0; i < (int)GTSlist.size(); i++) {
if (GTSlist[i].owner == rcvPacket->getSrcID()) {
if (GTSlist[i].length == rcvPacket->getGTSlength()) {
ackPacket->setGTSlength(GTSlist[i].length);
} else {
CAPlength += GTSlist[i].length;
GTSlist[i].length = 0;
index = i;
}
}
}
if (ackPacket->getGTSlength() == 0) {
if ((CAPlength - rcvPacket->getGTSlength()) * baseSlotDuration * (1 <<frameOrder) <minCAPLength) {
trace() <<"GTS request from "<< rcvPacket->getSrcID() <<
" cannot be acocmodated";
} else if (index != -1) {
GTSlist[index].length = rcvPacket->getGTSlength();
ackPacket->setGTSlength(GTSlist[index].length);
} else {
GTSspecA newGTSspecA;
newGTSspecA.length = rcvPacket->getGTSlength();
newGTSspecA.owner = rcvPacket->getSrcID();
GTSlist.push_back(newGTSspecA);
}
}
toRadioLayer(ackPacket);
toRadioLayer(createRadioCommand(SET_STATE, TX));
setMacState(MAC_STATE_IN_TX);
setTimer(ATTEMPT_TX, TX_TIME(ACK_PKT_SIZE));
break;
case MAC_802154_DATA_PACKET:{
int dstAddr = rcvPacket->getDstID();
if (dstAddr != SELF_MAC_ADDRESS && dstAddr != BROADCAST_MAC_ADDRESS)
break;
toNetworkLayer(decapsulatePacket(rcvPacket));
// If the frame was sent to broadcast address, nothing else needs to be done
if (dstAddr == BROADCAST_MAC_ADDRESS)
break;
Mac802154aPacket *ackPacket = new Mac802154aPacket("Ack packet", MAC_LAYER_PACKET);
ackPacket->setPANid(SELF_MAC_ADDRESS);
ackPacket->setMac802154aPacketType(MAC_802154_ACK_PACKET);
ackPacket->setDstID(rcvPacket->getSrcID());
ackPacket->setByteLength(ACK_PKT_SIZE);
case MAC_802154_ACK_PACKET:{
if (rcvPacket->getDstID() != SELF_MAC_ADDRESS)
break;
case MAC_802154_ASSOCIATE_PACKET:{// request to associate
// only PAN coordinators can accept association requests
// if multihop communication is to be allowed - then this has to be changed
// in particular, any FFD can become a coordinator and accept requests
if (!isPANCoordinator)
break;
// if PAN id is not the same as my ID - do nothing
if (rcvPacket->getPANid() != SELF_MAC_ADDRESS)
break;
// update associatedDevices and reply with an ACK (i.e. association is always allowed)
trace() <<"Received association request from "<< rcvPacket->getSrcID();
associatedDevices[rcvPacket->getSrcID()] = true;
Mac802154aPacket *ackPacket = new Mac802154aPacket("PAN associate response", MAC_LAYER_PACKET);
ackPacket->setPANid(SELF_MAC_ADDRESS);
ackPacket->setMac802154aPacketType(MAC_802154_ACK_PACKET);
ackPacket->setDstID(rcvPacket->getSrcID());
ackPacket->setByteLength(ACK_PKT_SIZE);
toRadioLayer(ackPacket);
toRadioLayer(createRadioCommand(SET_STATE, TX));

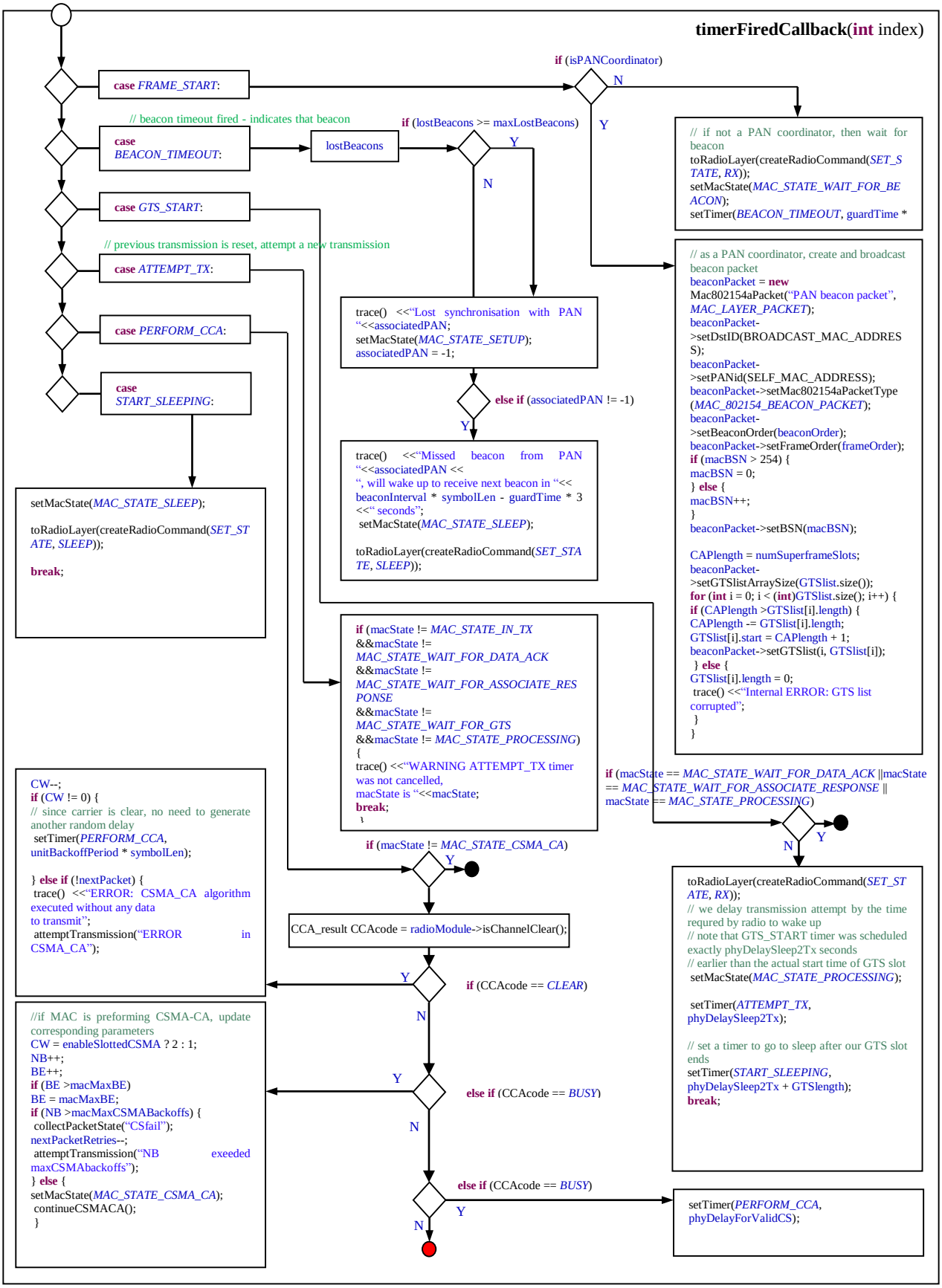

continueCSMACA
Y
We are not in MAC_STATE_CSMA_CA
macState != MAC_STATE_CSMA_CA
N
Generate a random delay, multiply it by backoff period length
int rnd = genk_intrand(1, (1 <<BE) - 1) + 1;
simtime_t CCAtime = rnd * (unitBackoffPeriod * symbolLen);
N
if using slotted CSMA - need to locate backoff period boundary
if (enableSlottedCSMA)
Y
simtime_t backoffBoundary = (ceil((getClock() - currentFrameStart) / (unitBackoffPeriod * symbolLen)) - (getClock() - currentFrameStart) / (unitBackoffPeriod * symbolLen)) * (unitBackoffPeriod * symbolLen);
CCAtime += backoffBoundary;
timerFiredCallback(CCAtime)


setMacState(int newState)
if (macState == newState)
macState = newState;


transmitNextPacket()
double txTime = TX_TIME(nextPacket-
N
We are still in CAP period of the frame
if (currentFrameStart + CAPend > getClock() + txTime)
Y
N
if (!enableCAP &&nextPacket->getMac802154aPacketType() ==
Y
not in CAP period and not in GTS, no transmissions possible
else if (requestGTS == 0 || GTSstart == 0)
outside GTS, no transmissions possible
else if (currentFrameStart + GTSstart > getClock() ||
Y
MAC_STATE_IDLE
transmission is allowed, decrement retry counter and modify mac and radio states.
nextPacketRetries--;
if (nextPacketResponse > 0) {
setMacState(nextMacState);
setTimer(ATTEMPT_TX, nextPacketResponse);
} else {
setMacState(MAC_STATE_IN_TX);
setTimer(ATTEMPT_TX, txTime);
nextMacState
MAC_STATE_IN_T
setMacState(MAC_STATE_IDLE || nextMacState || MAC_STATE_IN_TX);
timerFiredCallback(txTime || nextPacketResponse)


issueGTSrequest()
if (nextPacket)
Y
if (collectPacketState("NoPAN"))
packetBreak[nextPacketState]++;
cancelAndDelete(nextPacket);
N
nextPacket = new Mac802154aPacket("GTS request", MAC_LAYER_PACKET);
nextPacket->setPANid(associatedPAN);
nextPacket->setDstID(associatedPAN);
nextPacket->setMac802154aPacketType(MAC_802154_GTS_REQUEST_PACKET);
nextPacket->setSrcID(SELF_MAC_ADDRESS);
nextPacket->setGTSlength(requestGTS);
nextPacket->setByteLength(COMMAND_PKT_SIZE);
initiateCSMACA(9999, MAC_STATE_WAIT_FOR_GTS, ackWaitDuration + TX_TIME(COMMAND_PKT_SIZE));


handleAckPacket(Mac802154aPacket * rcvPacket)
switch
case MAC_STATE_WAIT_FOR_ASSOCIATE_RESPONSE
associatedPAN = rcvPacket->getPANid();
if (desyncTimeStart >= 0) {
desyncTime += getClock() - desyncTimeStart;
desyncTimeStart = -1;
}
trace() <<"associated with PAN:"<<associatedPAN;
cancelAndDelete(nextPacket);
nextPacket = NULL;
if (requestGTS) {
issueGTSrequest();
}
else {
attemptTransmission("Associated with PAN");
}
case MAC_STATE_WAIT_FOR_DATA_ACK:
case MAC_STATE_WAIT_FOR_GTS:
lockedGTS = true;
cancelAndDelete(nextPacket);
nextPacket = NULL;
if (enableCAP) {
attemptTransmission("GTS request granted");
} else {
setMacState(MAC_STATE_SLEEP);
toRadioLayer(createRadioCommand(SET_STATE, SLEEP));
if (isPANCoordinator || associatedPAN == rcvPacket->getSrcID()) {
collectPacketState("Success");
trace() <<"Packet successfully transmitted to "<< rcvPacket->getSrcID();
nextPacketRetries = 0;
attemptTransmission("ACK received");
}
break;



Figure_Apx31: IEEE 802.15.4 MAC implementation in Castalia simulator

References

Abramson, N. (1970). The ALOHA System-Another Alternative for Computer Communications. *In Proc. AFIPS Fall Joint Computer Conference, 37,* 281-285.

Ahmad, A. Riedl, A., Naramore, W.J., Nee-Yin C.,& Alley, M.S. (2009). Scenario-Based Traffic Modeling for Data Emanating from Medical Instruments in Clinical Environment. *2009 WRI World Congress on Computer Science and Information Engineering*, 529-533. doi: 10.1109/CSIE.2009.969.

Ahola, T.,Korpinen, P., Rakkola, J., and Ramo, T. (2007). Wearable FPGA Based Wireless Platform. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2281-2291. doi: 10.1109/IEMBS.2007.4352782

Bachlin, M., Forster, K., & Troster, G. (2009). SwimMaster: AWearable Assistant for Swimmer. *Proc. of the 11th International Conference on Ubiquitous Computing*, 215–224.

Bandyopadhyay, L. K., Chaulya, S. K., &Mishra, P. K. (2009). *Wireless Communication in Underground Mines: RFID-based Sensor Networking.* NewYork: Springer.

Battery University. (2011). The secrets of battery runtime [Article]. Retrieved from http://batteryuniversity.com/learn/article/the_secrets_of_battery_runtime

Bauer, P., Sichitiu, M., Istepanian, R., Premaratne, K. (2000). The Patient: Wireless Distributed Sensor Networks for Patient Monitoring and Care. *Proceedings of Information Technology in Biomedicine*, 17-21. doi: 10.1109/ITAB.2000.892341.

Bechler, M., Franz, W.J. & Wolf, L. (2003). Mobile Internet Access in FleetNet. *Proceedings of 13thConference on Communication in Distributed Systems, Germany.*

Bharghavan, V., Demers, A., Shenker, S. & Zhang, L. (1994). MACAW: A Media Access Protocol for Wireless LAN's. *SIGCOMM '94 Proceedings of the Conference on Communications Architectures, Protocols and Application*, 212-225. doi: 10.1145/190314.190334

Bianchi, G. (2000). Performance analysis of the IEEE 802.11 Distributed Coordination Function, *IEEE Journal on Selected Areas in Communications, (18)*3, 535-547.

Bilstrub, K. (2008). A Preliminary Study of Wireless Body Area Networks. [Technical Report IDE0854]. Retrieved from http://www.diva-portal.org/smash/get/diva2:239215/FULLTEXT01.pdf

Bin, Z., (2008). *IEEE P802.15 Wireless Personal Area Networks*.

Retrieved from
https://mentor.ieee.org/802.15/documents?n=9&is_group=0006

Bloom, E. D., Boersch-Supan, A., McGee, P., & Seike, A. (May 2011). *Population Aging: Facts, Challenges, and Responses* [Working Paper Series].
Retrieved from Harvard
Initiative for Global Health website:
http://www.hsph.harvard.edu/pgda/WorkingPapers/2011/PGDA_WP_71.pdf

Bonnici, T.,Orphanidou, C., Vallance, D., Darrell, A. (2012). Testing of Wearable Monitors in a Real-World Hospital Environment: What Lessons Can Be Learnt?. *Ninth International Conference on Wearable and Implantable Body Sensor Networks (BSN),* 79-84.doi: 10.1109/BSN.2012.31.

Boulis, A., & Tselishchev, Y. (2010). Contention vs. Polling: A Study in Body Area Networks MAC Design.*Proc. of the Fifth International Conference on Body Area Networks (BodyNetS 2010).* Corfu, Greece, 151-157.

Boulis, A. (2011). Castalia, A Simulator for Wireless Sensor Networks and Body Area Networks [User's Manual]. Available from
http://castalia.npc.nicta.com.au/pdfs/Castalia%20-%20User%20Manual.pdf

Burrati, Ch., D'Erriko, R., Maman, M., Martelli, F., Rosini, R., Verdone, R. (2011). Design of a Body Area Network for Medical Applications: The WiserBAN Project. *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies,* doi: 10.1145/2093698.2093862.

Calì, F., Conti, M. & Gregori, E. (2000). Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. *IEEE/ACM Transactions on Networking (TON), 8*(6), 785-799.

Cheng, L., Bourgeois, A.G., & Zhang, X. (2007). A New GTS Allocation Scheme for IEEE 802.15.4 Networks with Improved Bandwidth Utilization. *International Symposium on Communications and Information Technologies*, 1143-1148. doi: 10.1109/ISCIT.2007.4392189.

Chipara, O., Lu, Ch., Bailey, T. C., Roman, G. (2010). Reliable Clinical Monitoring Using Wireless Sensor Networks: Experiences in a Step-down Hospital Unit.*SenSys'10, Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems,* 155-168. doi: 10.1145/1869983.1869999.

Cho, C., Pak, J., Kim, J., Lee, I. & Han, K. (2006). A Random Backoff Algorithm for Wireless Sensor Networks. *LECTURE NOTES IN COMPUTER SCIENCE, 4003,* 108-117.

Cohen, J. E. (2003). Human Population: The Next Half Century. *Science Magazine, 302*(5648), 1172-1175.

Cotton, S. L. & Scanlon, W. G. (2009). Characterization and Modelling of On-body Spatial Diversity within Indoor Environments at 868 MHz. *IEEE Trans. Wireless Communication, 8*(1), 176–185.

Culmaraes, J., Ferrelra, C., Cabral, P. (2008). Pediatric Epilepsy Remote Monitoring. In *Vodafone Portegal*.
Retrieved from
http://www.vodafone.pt/main/A+Vodafone/EN/Fundacao/Projects/Health/epilepsy-monitoring.htm

Curtis, D. W., Pino, E. J., Bailey, J. M., Shih, E. I., Waterman, J., Vinterbo, S. A., Stair, T. O., Guttag, J. V, Greenes RA, Ohno-Machado L. (2008). SMART: an integrated wireless system for monitoring unattended patients. *PubMed, 15*(1), 44-53.

Dishong, T. D., & McGrowth, M. (2010). *Wireless Sensor Networks for Healthcare Applications* (1st ed.).
Boston: Artech House.

Edgar, H. & Callaway, J. (2005). The wireless sensor network MAC. *Handbook of Sensor Networks: Algorithms and Architectures,* 244-245.

El-Hoiydi, A., &Decotignie, J.D. (2004). WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. *Ninth International Symposium on ComputersandCommunications,* 244-251. doi: 10.1109/ISCC.2004.1358412.

Espina**,** J., Falck, T., & Mülhens, O. (2006). Body Sensor Networks**,** Network Topologies, Communication Protocols, and Standard. doi: 10.1007/1-84628-484-8_5.

FreeScale IEEE 802.15.4 Std/ZigBee. (2010). Software Selector Guide. [Application Note].
Retrieved from
http://cache.freescale.com/files/rf_if/doc/app_note/AN3403.pdf

Gao, J., Hu, J., & Min, G. (2008). A New Analytical Model for Slotted IEEE 802.15.4 Medium Access Control Protocol in Sensor Networks. In CMC '08. *WRI International Conference on Communications and Mobile Computing*, 427-431. doi: 10.1109/CMC.2009.342

Gao, T., Pesto, C., Selavo, L., Chen, Y., Ko, J., Kim, J., Terzis, A., Watt, A., Jeng, J., Chen, B., Lorincz, K., & Welsh, M. (2008). Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results. *Proc. of the HST,* Waltham, MA, USA.

Gay, D., Levis, P., Behren, R., Welsh, M., Brewer, E., & Culler, D. (2003). The nesC Language: A Holistic Approach to Networked Embedded Systems. *Proc. of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation*, 1-11.

Ghaboosi, K.,Pahlavan, K., &Pomalaza-Raez, C.A. (2011). A Cooperative Medical Traffic Delivery Mechanism for Multi-hop Body Area Networks. *IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2239 – 2243. doi: 10.1109/PIMRC.2011.6139916.

Ghildiyal, A., Godara, B., & Amara, A. (2011). Design of an ultra low power MAC for a heterogeneous in-body sensor network. *BodyNets '11 Proceedings of the 6th International Conference on Body Area Networks*, 60-66.

Golmie, N., Cypher, D., & Rebala, O. (2005). Performance Analysis of Low Rate Wireless Technologies for Medical Applications.*Computer Communications, 28*(10), 1266-1275.

Gong, H., Liu, M., Mao, Y., Chen, L. & Xie, L. (2005). Traffic Adaptive MAC Protocol for Wireless Sensor Network, *LECTURE NOTES IN COMPUTER SCIENCE, 3619*, 1134-1143.
Springer.

Gyselinckx, B.,Van Hoof, C., Ryckaert, J., &Yazicioglu, R.F. (2005). Human++: Autonomous Wireless Sensors for Body Area Networks. *Proceedings of the IEEE Conference on Custom Integrated Circuits*, 13-19. doi: 10.1109/CICC.2005.1568597

Hadid, N., Guitton, A., & Misson, M. (2009). Adaptive Slotted CSMA/CA Algorithm for the Traffic Accumulated During the Inactive Period. *Proceedings of the ACM Symposium on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks*, 143-146. doi: 10.1145/1641876.1641903.

Hanson, M. A., et al. (2009). Body Area Sensor Networks: Challenges and Opportunities. *Computer J, 42*(1), 59-60.

Hauer, J. (2011, June 1). *TestData Application*. [C++ source code].
Retrieved from
https://github.com/tinyos/tinyos-main/tree/master/apps/tests/tkn154/beacon-enabled/TestData

Hauer, J. (2012, November). *TKN15.4, A Platform-independent IEEE 802.15.4-2006 MAC Implementation*. [README.txt].
Retrieved from
https://github.com/tinyos/tinyos-main/tree/master/tos/lib/mac/tkn154

Heinzelman, W., Chandrakasan, A., Balakrishnan, H. & MIT, C. (2000). Energy-efficient Communication Protocol for Wireless Microsensor Networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. doi*:* 10.1109/HICSS.2000.926982

Hillman, K. M., Bristow, P. J., Chey, T., Daffurn,Jacques, K., Norman, T., S. L.,Bishop, G. F., & Simmons, G. (2001). Antecedents to Hospital Deaths. *Internal Medicine Journal, 31*(6), 342-348.

Hok Lim, H., & Qiu, B. (2001). Fuzzy Logic Traffic Control in Broadband Communication Networks. *The 10th IEEE International Conference on Fuzzy Systems,* 99-102. doi: 10.1109/FUZZ.2001.1007256.

Huq, M.A.,Dutkiewicz, E., Fang, G., Ren P. (2012). MEB MAC: Improved Channel Access Scheme for Medical Emergency Traffic in WBAN. *2012 International Symposium on Communications and Information Technologies (ISCIT),* 371-376. doi: 10.1109/ISCIT.2012.6380924.

Hurni, P., &Braun, T. (2010). MaxMAC: A Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks. In Lecture Notes in Computer Science (Vol. 5970, pp. 289-305).

Hussain, M., & Kwak, K. S. (2009). Positioning in Wireless Body Area Network using GSM", *International Journal of Digital Content Technology and its Applications, 3*(3).

Hwang, L. J., Sheu, S. T., Shih, Y., & Cheng, Y. C. (2005). Grouping Strategy for Solving Hidden Node Problem in IEEE 802.15.4 LR-WPAN. *Proc. of theFirst International Conference on Wireless Internet (WICON'05).*

IEEE 802.15.4 User Guide. (2006). Available from http://www.atmel.com/Images/doc5182.pdf

IEEE 802.2: Logical Link Control. (1998). Available from http://standards.ieee.org/about/get/802/802.2.html

IEEE P802.15.6/D01: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) used in or around a body. (2010).

IEEE, Std. 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Personal Area Networks (WPAN). (2003).

IRB, Institutional Review Board Services, (n. d.).

Retrived 24th June, 2014, from
http://www.irbservices.com/irbservices/Home.html

Isikman, A. O., Cazalon, L., Chen, F., and Li, P. (2011). *Body Area Networks*. (Final report of Group 6 of the Course SSY145 Wireless Networks). Gothenburg, Sweden: Chalmers University of Technology.

Jain, R. (2007). *Channel Models, A Tutorial* [Tutorial]. Retrieved from http://www.cse.wustl.edu/~jain/cse574-08/ftp/channel_model_tutorial.pdf

Javaid, N., Khan, NA., Shakir, M., Khan, MA., Bouk, SH., & Khan, ZA. (2013). Ubiquitous HealthCare in Wireless Body Area Networks-ASurvey. *Journal of Basic and Applied Science, 3*(4), 747-759.

Jovanov, E., Price, J., Raskovic, D., Kavi, K., Martin, T., Adhami, R. (2000). Wireless personal area networks in telemedicalenvironment. *Proceeding of the International Conference on Information Technology Applications in Biomedicine*, 22-27. doi: 10.1109/ITAB.2000.892342.

Kara, A., & Bertoni, H. L. (2001). Blockage/Shadowing and Polarization Measurements at 2.45 GHz for Interference Evaluation between Bluetooth and IEEE 802.11 WLAN. *Int. Symposium on Antennas Propagation*, 376–379.

Karn, P. (1990). MACA- A New Channel Access Method for Packet Radio. *9th Computer Networking Conference on ARRL/CRRL Amateur Radio.* 134-140.

Kleinrock, L. & Tobagi, F. (1975). Random Access Techniques for Data Transmission Over Packet-switched Radio Channels. *Proceedings of the National Computer Conference and Exposition* ACM New York, NY. 187-201. doi: 10.1145/1499949.1499984

Ko, J., Cho, Y., & Kim, H. (2006). Performance Evaluation of IEEE 802.15.4 MAC with Different Backoff Ranges in Wireless Sensor Networks. *10th IEEE Singapore International Conference on Communication systems*, 1-5. doi: 10.1109/ICCS.2006.301525.

Ko, J., Musaloiu-Elefteri, R., Lim, J., Chen, Y., Terzis, A., Gao, T., Destler, W., Selavo, L. (2008). MEDiSN: Medical Emergency Detection in Sensor Networks.*SenSys'08, Proceedings of the 6th ACM conference on Embedded network sensor systems*, 361-362. doi:10.1145/1460412.1460452.

Koubaa, A., M. Alves , B. Nefzi and Y.-Q. Song. (2006). Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Ttime-critical Events in Wireless Sensor Networks. *Proc. of Workshop Real-Time Networks (RTN 2006), Satellite Workshop to ECRTS*.

Koubaa, A., Alves, M., Tovar, E. (2007). Time Sensitive IEEE 802.15.4 Protocol. In *Sensor Networks and Configuration* (pp. 19-49).

Berlin Heidelberg: Springer.

Koubaa, A., Chaudhry, S., Gaddour, O., Chaari, R., Al-Elaiwi, N., Al-Soli, H., & Boujelben, H. (2011). Z-Monitor: Monitoring and Analyzing IEEE 802.15.4-based Wireless Sensor Networks. *36th IEEE Conference on Local Computer Networks (LCN 2011).* doi: http://doi.ieeecomputersociety.org/10.1109/LCN.2011.6115575

Koubâa, A., Severino, R., Alves, M., & Tovar, E. (2009). H-NAMe: A Hidden-Node Avoidance Mechanism for Wireless Sensor Networks. *Proc. of the 8th IFAC International Conference on Field Buses and Networks in Industrial and Embedded Systems'09*.

Kurose, J.F., Ross, K.W. & Ross, K. (2003). Chapter 6: Delay and Loss in Packet-Switched Networks. *In Computer networking: a top-down approach featuring the Internet*. Addison-Wesley Reading, MA.

Kwak, K. S., Ullah, S., and Ullah, N. (2011, February 28). *An Overview of IEEE 802.15.6 Standard*, [Article].
Retrived from
http://arxiv.org/PS_cache/arxiv/pdf/1102/1102.4106v1.pdf

Lamahewa, T., Hanlen, L., Miniutti, D., Smith, D., Rodda, D., & Gilbert, B. (2010*). BAN sleeping channel: Implications for Relays* [IEEE document].
Retrieved from
https://mentor.ieee.org/802.15/dcn/10/15-10-0306-00-0006-ban-sleeping-channel-implications-for-relays.pdf

Lamprinos, I. E., Prentza, A., Sakka, E., & Koutsouris, D. (2005). Energy-efficient MAC Protocol for Patient Personal Area Networks. *27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS*, 3799–3802. doi:

Latré, B., Braem, B., Moerman, I., Blondia, C., & Demeester, P. (2011). A survey on wireless body area networks. *Wireless Networks Journal*, *17*(1), 1-18.

Latré, B., Braem, B., Moerman, I., Blondia, C., Reusens, E., Joseph, W., Demeester, P. (2007). A Low-delay Protocol for Multihop Wireless Body Area Networks. *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007.*1-8. doi: 10.1109/MOBIQ.2007.4451060.

Latré, B., De Poorter, E., Moerman, I., Demeester, P. (2007). Lecture Notes in Computer Science: MOFBAN: A Lightweight Modular Framework for Body Area Networks. Volume 4808/2007. doi: 10.1007/978-3-540-77092-3_53.

Lee, H., Cerpa, A., & Levis, P. (2007). Improving Wireless Simulation Through Noise Modeling. *IPSN '07 Proceedings of the 6th International Conference on Information*

Processing in Sensor Networks, 21-30. doi: 10.1145/1236360.1236364.

Lee, Y. D., & Chung, W. Y. (2009). Wireless Sensor Network Based Wearable Smart Shirt for Ubiquitous Health and Activity Monitoring. *Sensors and Actuators B: Chemical, ESEVIER, 140*(2), 390-395.

Lee, C., Lee, H., & Choi, S. (2010). An Enhanced MAC Protocol of IEEE 802.15.4 for Wireless Body Area Networks. *Proc. Of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*.

Levis, Ph. (2005, July). *Hardware Platforms for TinyOS*. [README.txt]. Retrieved from https://github.com/tinyos/tinyos-main/tree/master/tos/platforms

Li, C., Wang, L., Li, J., Zhen, B., Li, H., & Kohno, R. (2009). Scalable and Robust Medium Access Control Protocol in Wireless Body Area Networks. *IEEE PIMRC*, Tokyo, 2127–2131.

Li, H., & Tan, J. (2005). An Ultra-low-power Medium Access Control Protocol for Body Sensor Network. *62nd IEEE Conference on Vehicular Technology*,2342-2346. doi: 10.1109/VETECF.2005.1558967

Li, H., & Tan, J. (2010). Heartbeat-Driven Medium-Access Control for Body Sensor Networks. *IEEE Transactions on Information Technology In Biomedicine*, Vol. 14.

H. Lin and G. Campbell. (1993). Using DQRAP (Distributed Queuing Random Access Protocol) for Local Wireless Communications. *In Proc. of Wireless '93*, Calgary, Canada, 625–635.

Liu, B., Yan, Z., & Chen, C. W. (2011). CA-MAC: A Hybrid Context-aware MAC Protocol for Wireless Body Area Networks. *IEEE 13th International Conference on e-Health Networking, Applications and Services*, 213-216. doi: 10.1109/HEALTH.2011.6026748.

Lorincz, k., Chen, B. Rong, Challen, G. W., Chowdhury, A. R., Patel, S., Bonato, P., & Welsh. M. (2009). Mercury: AWearable Sensor Network Platform for High-fidelity Motion Analysis. *SenSys '09*.

Lv, J., Zhang, X., Han, X. & Fu, Y. (2007). A Novel Adaptively Dynamic Tuning of the Contention Window (CW) for Distributed Coordination Function in IEEE 802.11 Ad hoc Networks. *International Conference on Convergence Information Technology*, 290-294. doi: 10.1109/ICCIT.2007.146.

Mahalik, N. P. (2007). Fundamentals, Standards, Platforms, and Applications. In Sensor Networks and Configuration (Eds.), (pp. 29). eBook.

Mahtab Alam, M., Berder, O., Menard, D., & Sentieys, O. (2012). Latency-Energy

Optimized MAC Protocol for Body Sensor Networks. *BSN '12 Proceedings of the 2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks,* 67-72. doi: 10.1109/BSN.2012.8.

Marinkovic, S.J. Popovici, E.M., Spagnol, C., Faul, S., &Marnane, W.P. (2009). Energy-Efficient Low Duty Cycle MAC Protocol for Wireless Body Area Networks. *IEEE Transactions on Information Technology in Biomedicine, 13*(6), 915-925.

Marinkovic, S., Spagnol, C., & Popovici, E. (2009). Energy-Efficient TDMA-Based MAC Protocol for Wireless Body Area Networks. *Third International Conference* on *Sensor Technologies and Applications, SENSORCOMM '09,* 604-609. doi: 10.1109/SENSORCOMM.2009.99.

Mathworks: MATLAB. (2014). The Language of Technical Computing. Retrieved from http://www.mathworks.com.au/products/matlab/

Milenkovic, A., Otto, C., & Jovanov, E. (2006). Wireless sensor networks for personal health monitoring: issues and an implementation. *Computer Communication*, 2521–2533.

Miluzzo, E., Zheng, X., Fodor, K., & Campbell, A. T. (2008). Radio Characterization of 802.15.4 and its Impact on the Design of Mobile Sensor Networks.*Proc. of the 5th European conference on Wireless sensor networks EWSN'08.*

Ming, C., Wong & Pin Hsu, W. (2010). An Additional Clear Channel Assessment for IEEE 802.15.4 Slotted CSMA/CA Networks. *2010 IEEE International Conference on Communication Systems (ICCS)*, 62-66.

Mouzehkesh, N. &Zia, T.A. (2011). A Dynamic Backoff Approach in Wireless Sensor Networks for Environmental Monitoring. *Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 247-252. doi: 10.1109/ISSNIP.2011.6146532.

Muneer Bani, Y., Marwa, S., Wail, M., & Khamayseh, Y. (2012). Fibonacci Backoff Algorithm for IEEE 802.15.4/ZigBee. *Network Protocols & Algorithms, 4*(3), 62-78.

Munir, Saad A., Bin, Y., Biao, R., & Jian, M. (2007). Fuzzy Logic Based Congestion Estimation for QoS in Wireless Sensor Network. IEEE Wireless Communications and Networking Conference, WCNC '07, 4336 – 4341. doi: 10.1109/WCNC.2007.791.

Muqattash, A. (May, 2012). System and Method for MAC Layer Clock Drift Compensation [Patent: US21020127902A1]. Available from http://www.google.com/patents/US20120127902

Nabi, M., Geilen, M., & Basten. T. (2011). MoBAN: A configurable mobility model for

wireless body area networks. *Proc. of the 4th Int'l Conf. On Simulation Tools and Techniques (SIMUTools). ICST.*

Nasir, Q. & Albalt, M. (2008). History Based Adaptive Backoff (HBAB) IEEE 802.11 MAC Protocol. *6th Annual Communication Networks and Services Research Conference, CNSR'08,* 533-538. doi: 10.1109/CNSR.2008.20.

National ICT Australia Ltd (NICTA). (2014). Available from http://www.nicta.com.au/

O'Donovan, T., Sreenan, C., Sammon, D., O'Reilly, P., & O'Connor, K. A. (2009). A Context Aware Wireless Body Area Network (BAN). *Proc. of the Pervasive Health Conference.*

Obayashi, S., & Zander, J. (1998). A Body-Shadowing Model for Indoor Radio Communication Environments, Antennas, and Propagation. *IEEE Transations on Research & Development Centre, Toshiba Corporation, and Kawasaki*. 920-927.

Omeni, O.C., Toumaz Technol. Ltd., Didcot, Eljamaly, O., &Burdett, A.J. (2007). Energy Efficient Medium Access Protocol for Wireless Medical Body Area Sensor Networks. *4th IEEE/EMBS International Summer School and Symposium on Medical Devices and Biosensors, 29-32. doi: 10.1109/ISSMDBS.2007.4338285.*

OMNet++ Network Simulation Framework. (2014). Retrieved from http://www.omnetpp.org/

OPNET Simulator v11. (n. d). Retrieved from http://www.opnet.com

Osterlind, F., Wirstrom, N., Tsiftes, N., Finne, N., Voigt, T., & Dunkels, A. (2010). StrawMAN: Making Sudden Traffic Surges Graceful in Low-power Wireless Networks. *HotEmNets '10 Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*. doi: 10.1145/1978642.1978660

Otal, B., & Alonso, L. (2009). Highly Reliable Energy-Saving MAC for Wireless Body Sensor Networks in Healthcare Systems. *IEEE Journal on Selected Areas In Communications, 27*(4).

Papadimitratos, P. Mishra, A., & Rosenburgh, D. (2005). A Cross-layer Design Approach to Enhance 802.15.4. *2005 IEEE Military Communications Conference, MILCOM '05*, 1719-1726. doi:10.1109/MILCOM.2005.1605922.

Park, P., Fischione, c., & Johansson, K. (2010). Adaptive IEEE 802.15.4 Protocol for Energy Efficient, Reliable and Timely Communications.*Proceedings of the 9th ACM/IEEE*

International Conference on Information Processing in Sensor Networks, 327-338. doi:10.1145/1791212.1791251.

Peng, Y., Wu, H., Cheng, S. & Long, K. (2002). A new self-adapt DCF algorithm. *IEEE Global Telecommunications Conference, GLOBECOM'02*, 87-91. doi: 10.1109/GLOCOM.2002.1188047.

Pollin, S., Ergen, M., Ergen, S., Bougard, B., DerPerre, L., Moerman, I., Bahai, A., Varaiya, P., & Catthoor, F. (2008). Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer. *IEEE Trransactions on Wireless Communications*, 3359-3371. doi: 10.1109/TWC.2008.060057.

Prabh, K., & Hauer, J.H. (2011). Opportunistic Packet Scheduling in Body Area Networks. *Proc. of the European Conference on Wireless Sensor Networks (EWSN)*, Bonn, Germany.

Prakash Rao, V., &Marandin, D. (2006). Adaptive Backoff Exponent Algorithm for Zigbee (IEEE 802.15.4). Next Generation Teletraffic and Wired/Wireless Advanced Networking, *Lecture Notes in Computer Science* (pp. 501-516). Springer Berlin Heidelberg.

Rada-Vilela, J. (2013). Fuzzy-Lite, A Fuzzy Logic Library written in C++. Retrieved from http://www.fuzzylite.com/

Rada-Vilela, J. (Producer). (2013, October 27). Fuzzylite: A Fuzzy Logic Control Library in C++. [Video]. Retrieved from https://www.youtube.com/watch?v=rSAlWPyaA34

Rahman, A., Kennedy, P., Simmonds, A., & Edwards, J. (2008). Fuzzy Logic Based Modeling and Analysis of Network Traffic. *8th IEEE International Conferenceon Computer and Information Technology*, 652-657. doi: 10.1109/CIT.2008.4594752

Raptis, P., Vitsas, V., & Paparrizos, K. (2008). Packet Delay Metrics for IEEE 802.11 Distributed Coordination Function. *ACM Journal of Mobile Networks and Application December, 14*(6), 772-781.

Ren, Q., & Liang, Q. (2006). Energy-efficient medium access control protocols for wireless sensor networks.*EURASIP Journal on Wireless Communications and Networking*, *2006*(2), 5-5. doi: 10.1155/WCN/2006/39814.

Ren, Z., Zhou, G., Pyles, A. J., Keally, M., Mao, W. & Wang, H. (2011). Body T2: Throughput and Time Delay Performance Assurance for Heterogeneous BSNs. *IEEE INFOCOM 2011*, Shanghai, China.

Sahinoglu, Z., & Guvenc, I. (2011).*ZigBee networks and low-rate UWB communications.*

Cambridge University Press.

Salcic, Z. (2014). Fuzzy Logic Application-Specific Processor for Traffic Control in ATM Network.

Sathyan, T., Humphrey, D., & Hedley, M. (2011). WASP: A System andAlgorithms for Accurate Radio Localization Using Low-CostHardware.*IEEE Transactions on Systems, Man, and Cybernetics- Part C,41*(2), 211-222.

Sayrafian-Pour, K., Yang, W., Hagedorn, J., Terrill,J., Yazdandoost, K. Y., Hamaguchi K. (2010). Channel Models for Medical Implant Communication.*International Journal of Wireless Information Networks*,*17*(3-4), 105-112.

Schmidth, R., Norgall, T., Morsdof, J., Bernhard, J., & Von Der Grun, T. (2002). Body Area Network BAN: AKey Infrastructure Element for Patient-centered Medical Applications. *Journal of BioMed Tech, 2002*(47), 365-368.

Shimmer Research Group. (2014). [Fact Sheet]. Shimmer Connect Rev 0.10a User Manual. Retrived from http://www.shimmersensing.com/images/uploads/docs/Shimmer_Connect_User_Manual_Rev0.10a.pdf

Shimmer Research Group. (2014). [Fact Sheet]. Shimmer Dock 2 Quick Start Rev 1bUser Manual.
Retrieved from
http://www.shimmersensing.com/images/uploads/docs/Shimmer_Dock_User_Guide_rev1.6.pdf

Shimmer Research Group. (2014). [Fact Sheet]. Shimmer Sensing LabVIEW Instrument Driver Library User Manual Revision 2.1a.
Retrieved from
http://www.shimmersensing.com/images/uploads/docs/ShimmerSensing_LabVIEW_Library_User_Manual_rev2.1a.pdf

Shimmer Research Group. (2014). [Fact Sheet]. Wireless EMG Sensor Module Specification Fact Sheet.
Retrieved from
http://www.shimmersensing.com/images/uploads/docs/Wireless_EMG_Sensor_Module_Spec_Sheet.pdf

SHIMMER Sensing Technology. (2014). Available from http://www.shimmersensing.com/

SMA-WiBAN. (2010). *MAC and Security Baseline Proposal*.; *IEEE 802.15 Documents, Document no. 196*, rev.2.
Retrieved from
https://mentor.ieee.org/802.15/documents?is_group=0006

Su, H., & Zhang, X. (2009). Battery-dynamics Driven TDMA MAC Protocols for Wireless Body-area Monitoring Networks in Healthcare Applications. *IEEE J. Sel. Areas Communication*‘09. *27*(4), 424–434.

Sun, M., Sun, K. & Zou, Y. (2009). Analysis and Improvement for 802.15.4 Multi-hop Network. *Proc. of the 2009 International Conference on Communications and Mobile Computing.*

Timmons, N.F., &Scanlon, W.G. (2004). Analysis of the Performance of IEEE 802.15.4 for Medical Sensor Body Area Networking. *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004,* 16-24. doi: 10.1109/SAHCN.2004.1381898.

TinyOS, an Operating System Designed for Low Power Wireless Devices. (2013). Available from http://www.tinyos.net/

TKN15.4. (2014). Available from https://code.google.com/p/tinyos-main/source/browse/trunk/tos/lib/mac/tkn154/?r=5572

Tselishchev,Y., Libman,L., Boulis, T. (2011). Reducing Transmission Losses in Body Area Networks using Variable TDMA Scheduling. *The 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2011),* 374-381.

Turon, M. (2005, February). *The tos/chips Directory*. [README.txt]. Retrieved from https://github.com/tinyos/tinyos-main/tree/master/tos/chips

Ullah, S., & Sup Kwak, K. (2010). Performance Study of Low Power MAC Protocols for Wireless Body Area Networks. *21st IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications Workshops*, 112-116. doi: 10.1109/PIMRCW.2010.5670417

Van Dam, K., Pitchers, S., & Barnard, M. (2001). Body Area Networks: Towards a Wearable Future. *Proceedings of WWRF kick off meeting*, *17*(1), 1-18. doi:10.1007/s11276-010-0252-4.

Viot, G. (1993, February). “Fuzzy logic in C,” Dr. Dobb’s Journal, special issue on Cognitive Computing, pp. 40-49 and 94. Available: http://www.chebucto.ns.ca/Science/AIMET/archive/ddj/fuzzy_logic_in_C [2013]

Wang, L. (1997). A Course in Fuzzy Systems and Control Prentice Hall International, Inc.

Wang, P., & Akyildiz, I. F. (2011). Spatial Correlation and Mobility-Aware Traffic Modeling

for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking, 19*(6), 1860-1873.

Wang, Q. (2010). Traffic Analysis & Modeling in Wireless Sensor Networks and Their Applications on Network Optimization and Anomaly Detection. *Journal of Network Protocols and Algorithms, 2*(1), 74-92.

Willig, A. (2006). Wireless Sensor Networks: Concept, Challenges and Approaches. *Elektrotechnik und Informationstechnik, 123*(6). 224-231.

Wong, A. C. W., Mcdonagh, D., Omeni, O.,Nunn, C., Hernandez-Silveira, M., & Burdett, A. J. (2009). Sensium: An Ultra-Low-Power Wireless Body Sensor Network Platform: Design & Application Challenges. *AnnualInternational Conference of the IEEESociety in Engineering, Medicine and Biology, EMBC 2009*, doi: 10.1109/Iembs.2009.5334001, 6576.

Xi, P., Guo, H. &Shu. C. (2011). Human Body Shape Prediction and Analysis Using Predictive Clustering Tree. *Proc. Of The 2011 International Conference On 3D Imaging, Modeling , Processing, Visualization and Transmission (3DIMPVT)*. 196 – 203.

Xijun, C., Meng, M. Q. H., & Hongliang, R. (2005). Design of Sensor Node Platform for Wireless BiomedicalSensor Networks. Unpublished.

Yan, L., Zhong, L., & Jha, N. (2007). Energy Comparison and Optimization for Wireless Body Area Network Technologies. *BodyNets '07 Proceedings of the ICST 2nd International Conference on Body Area Networks.*

Ye, W., Heidemann, J., & Estrin, D. (2002). Sensor-MAC (S-MAC): Medium Access Control for Wireless Sensor Networks. *Proc. of the21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, 1567-1576. doi: 10.1109/INFCOM.2002.1019408.

Ye, W., Heidemann, J. & Estrin, D. (2004). Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking, 12*(3), 493-506 .

Ying Lei, X., Choi, Y., Park, S., & Rhee, S. (2012). GTS Allocation for Emergency Data in Low-rate WPAN. *18th Asia-Pacific Conference on Communications (APCC)*, 792-793. doi: 10.1109/APCC.2012.6388216.

Yuce, M. R. (2010). Implementation of wireless body area networks for healthcare systems. *Sensor and Actuators: A Physical, 162*(1).

Zadeh, L. A. (June 1965). Fuzzy Sets. *Information and Control*, Elsevier, *8*(3), 338-353.

Zhang, Y. & Dolmans, G. (2009). A New Priority-guaranteed MAC Protocol for Emerging Body Area Networks. *Proc. of the 2009 Fifth International Conference on Wireless and Mobile Communications*.

Zhisheng, Y. & Liu, B. (2011). A Context Aware MAC Protocol for Medical Wireless Body Area Network. *7th International Conference onWireless Communications and Mobile Computing (IWCMC)*, 2133-2138. doi: 10.1109/IWCMC.2011.5982864.

Zhisheng Y.,Bin L., & Chang W. C. (2012). QoS-driven Scheduling Approach Using optimal Slot Allocation for Wireless Body Area Networks. *IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 267-272. doi: 10.1109/HealthCom.2012.6379419

Zhou, G., Lu, J., Wan, C.Y., Yarvis, M. D., & Stankovic, J. A. (2008). BodyQoS: Adaptive and Radio-gnostic QoS for Body Sensor Networks. *Proc. of the IEEE INFOCOM '08, Phoenix, AZ*.

Zhuo, S., Song, Y., Wang, Z., &Wang, Zh. (2012). Queue-MAC: A Queue-length Aware Hybrid CSMA/TDMA MAC Protocol for Providing Dynamic Adaptation to Traffic and Duty-cycle Variation in Wireless Sensor Networks. *9th IEEE International Workshop on Factory Communication Systems (WFCS)*, 105-114. doi: 10.1109/WFCS.2012.6242552.

ZigBee Alliance. (2014). Retrieved from http://www.zigbee.org/

Zuo, J., Xin Ng, S., & Hanzo, L. (2010). Fuzzy Logic Aided Dynamic Source Routing in Cross-Layer Operation Assisted Ad Hoc Networks. *IEEE 72nd Conference on Vehicular Technology Conference (VTC 2010-Fall)*,1090-3038. doi: 10.1109/VETECF.2010.5594508.