PERFORMANCE EVALUATION OF AN UNDERWATER ACOUSTIC SENSOR NETWORK COMMUNICATION PROTOCOL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DERYA ACAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2014

Approval of the thesis:

PERFORMANCE EVALUATION OF AN UNDERWATER ACOUSTIC SENSOR NETWORK COMMUNICATION PROTOCOL

submitted by **DERYA ACAR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan ____________________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Semih Bilgen ____________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Cüneyt Bazlamaçcı ____________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Semih Bilgen ____________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Ece Schmidt ____________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Altan Koçyiğit ____________________
Information Systems, METU

Assoc. Prof. Dr. Murat Erten ____________________
Executive Committee Member, İnnova A.Ş.

Date: 06.02.2014

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: DERYA ACAR

Signature :

ABSTRACT

PERFORMANCE EVALUATION OF AN UNDERWATER ACOUSTIC SENSOR NETWORK COMMUNICATION PROTOCOL

Acar, Derya
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Semih Bilgen

February 2014, 57 pages

Underwater acoustic sensor networks constitute a popular research issue, regarding which researchers have especially focused on energy efficiency problem. Methods to minimize energy consumption have been investigated in terms of network layers, especially data link and routing layer. However, RPUASN (Remotely Powered Underwater Acoustic Sensor Networks), which is a recent proposal, solves this problem by including acoustic energy sources and sensor nodes which can harvest the emitted energy in the environment. Sink Powered Underwater Acoustic Sensor Networks (SPUASN) is a particular kind of RPUASN, in which the sink node also operates as acoustic energy source. X-PACCA which is a recently proposed protocol for SPUASN is a cross layer solution including data link, routing and transport layers. In this study, to be able to evaluate the performance characteristics of X-PACCA protocol, an event-based simulator is developed in MATLAB. To validate the simulator, CSMA and DACAP protocols are implemented in this simulator, and their latency and throughput characteristics are investigated and compared with published references. Then, X-PACCA protocol is implemented in this simulator and its performance is evaluated in terms of average end-to-end delay and packet delivery ratio. The present work extends earlier similar studies and shows that parameter value choice for X-PACCA protocol is important to achieve low latency and high packet delivery ratio.

Keywords: Underwater Acoustic Sensor Networks, Remotely Powered Underwater Acoustic Sensor Networks (RPUASN), Sink Powered Underwater Acoustic Sensor Networks (SPUASN), Cross Layer Power Adaptive CSMA/CA (X-PACCA), event-based network simulator design

ÖZ

BİR SUALTI AKUSTİK ALGILAYICI AĞI İLETİŞİM PROTOKOLÜNÜN PERFORMANS DEĞERLENDİRMESİ

Acar, Derya
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Semih Bilgen

Şubat 2014, 57 sayfa

Sualtı akustik algılayıcı ağları araştırmacıların özellikle enerji verimliliği problemine odaklandığı popüler bir araştırma konusudur. Ağ katmanlarında, özellikle MAC ve yönlendirme katmanında enerji tüketimini minimize eden yöntemler araştırılmıştır. Fakat yeni bir öneri olan uzaktan beslemeli UASN (RPUASN) bu problemi ortama akustik enerji yayan düğümler ve ortamdaki enerjiyi kullanabilen algılayıcı düğümler koyarak çözmüştür. Alıcı tarafından beslemeli UASN (SPUASN), alıcı düğümün enerji kaynağı olarak görev aldığı RPUASN'nin özel bir türevidir. SPUASN için önerilmiş X-PACCA protokolü MAC, yönlendirme ve iletim katmanlarını kapsayan yeni bir çalışmadır. Bu çalışmada X-PACCA protokolünün performans özelliklerini değerlendirebilmek için MATLAB ortamında olay tabanlı bir simülatör geliştirilmiştir. Simülatörü geçerli kılmak için simülatör içerisinde CSMA ve DACAP protokolleri uygulanmış ve onların gecikme ve çıktı verimi performansları araştırılmış ve yayınlanan referanslar ile karşılaştırılmıştır. Daha sonra X-PACCA protokolü bu simülatör içerisinde uygulanmış ve ortalama uçtan uca gecikme ve paket gönderim oranı kriterleri bazında performansı değerlendirilmiştir. Bu çalışma daha önce yapılmış benzer çalışmaları ilerletmekte ve X-PACCA protokolünün düşük gecikme ve yüksek paket gönderim oranı başarımı için parametre değer seçiminin önemini göstermektedir.

Anahtar Kelimeler: Sualtı Akustik Algılayıcı Ağları, Uzaktan Beslemeli Sualtı Akustik Algılayıcı Ağlar, Alıcı Beslemeli Sualtı Akustik Algılayıcı Ağlar, Katmanlar Arası Güç Uyarlamalı CSMA/CA, olay tabanlı ağ simülatörü tasarımı

To My Family

ACKNOWLEDGMENTS

I wish to express my sincere thanks to my thesis supervisor Prof. Dr. Semih Bilgen for his valuable guidance and encouragement.

I would like to thank Alper Bereketli for his helpful responses to my questions related to my thesis work and his advices on writing a simulator program in Matlab.

In addition, I am also thankful to TÜBİTAK UZAY and my colleagues for their friendship and encouragement.

Finally, I owe my particular thanks to my family for their love, support, patience and understanding during my studies. Without their support, it would be impossible to complete this study.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgment
AUV	Autonomous Underwater Vehicle
BER	Bit Error Rate
BFSK	Binary Frequency Shift Keying
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DACAP	Distance Aware Collision Avoidance Protocol
FDMA	Frequency Division Multiple Access
MAC	Medium Access Control
PER	Packet Error Rate
RPUASN	Remotely Powered Underwater Acoustic Sensor Networks
RTS	Request to Send
RVS	Receiving Voltage Sensitivity
SINR	Signal to Interference and Noise Ratio
SIR	Signal to Interference Ratio
SNR	Signal-to-Noise Ratio
SPUASN	Sink Powered Underwater Acoustic Sensor Networks
TDMA	Time Division Multiple Access
UASN	Underwater Acoustic Sensor Networks
WFA	Wait for ACK
X-PACCA	Cross Layer Power Adaptive CSMA/CA

CHAPTER 1

INTRODUCTION

Underwater acoustic sensor networks (UASNs) which can serve for many application areas of underwater is coming into prominence increasingly. Due to the fact that the underwater environment has a more challenging structure when compared to terrestrial region, sensor network design for underwater requires different considerations. For example, acoustic signalization is used for communication because of its being more convenient than radio or optic signals. However, its speed is low, which is approximately 1500m/s, so propagation delay becomes an important issue to take into account in protocol design of network layers. Other challenges are discussed in Chapter 2.

Energy efficiency is a critical problem for UASN as in the case of terrestrial sensor networks. Furthermore, underwater sensor nodes are more expensive than terrestrial sensor nodes [15] so that extending lifetime of underwater nodes by consuming less energy is more important. Many protocol studies, especially data link and routing layer protocol propositions which claim to provide high energy efficiency for UASN are realized. However, generally there is a trade-off between energy efficiency and packet latency because the mechanism behind energy-efficient protocols mostly depends on sleep cycles. Thus, it becomes troublesome for delay sensitive applications.

Energy efficiency problem for UASN is solved with the RPUASN (Remotely Powered Underwater Acoustic Sensor Network) proposal in [3] which shows that it is possible to supply energy to sensor nodes with acoustic signals emitted from a remote acoustic energy source. This novel concept enables new protocol designs because enery efficiency is no more a problem. X-PACCA [4] which is a recently proposed protocol for RPUASN is a cross layer solution including data link, routing and transport layers. In [4] and [17], X-PACCA protocol is implemented in Aqua-Sim and some important simulation studies are realized for evaluation of the performance characteristics and comparison with two MAC layer protocols, Slotted FAMA and UWAN-MAC.

In the present study, to be able to further evaluate the performance characteristics of X-PACCA, an event-based simulator in MATLAB is developed for underwater acoustic sensor networks. While Aqua-Sim has been used quite often for underwater sensor network studies, MATLAB is known to provide an easy-to-use infrastructure that can easily be extended to cater for diverse problem settings and issues. Hence, in the present study, using the MATLAB environment has been preferred. For the verification of the simulator, CSMA and DACAP protocols are implemented in this simulator to compare the simulation results with the results published in [8] by applying identical scenarios in terms of latency and throughput characteristics. Afterwards, X-PACCA protocol is implemented in the simulator together with the SPUASN concept. Some simulation scenarios in [4] are applied to compare the results. Then, further evaluation studies are realized for X-PACCA protocol in terms of the effects of some parameters on protocol performance and MAC layer performance comparison with two known UASN MAC protocols, which are CSMA and DACAP. Besides providing a user-friendly and versatile simulator based on MATLAB infrastructure, some performance aspects of X-PACCA hitherto unexplored are thus uncovered.

The rest of this work is organized as follows: Chapter Two outlines a review of the relevant literature, Chapter Three is devoted to the prepared simulation infrastructure and repetition of the results reported in the literature, Chapter Four presents the performance evaluation of X-PACCA protocol performed with the MATLAB simulator and Chapter Five concludes the study commenting on the results and suggesting relevant future work.

CHAPTER 2

LITERATURE SURVEY

2.1 Design Issues

UASN which is composed of sensor nodes and vehicles deployed over a specific region to realize a collaborative monitoring task is a popular research issue. Some basic application areas of underwater sensor networks are ocean sampling networks, environmental monitoring, undersea explorations, disaster prevention, assisted navigation, distributed tactical surveillance and mine reconnaissance [1]. The communication basically depends on acoustic communication because radio waves do not propagate well in water at comparatively high frequencies and optical waves undergo scattering. There are some challenges for underwater sensor network design, which are caused by the characteristics of acoustic communication channel. These are:

- limited bandwidth,
- distortions due to multipath and fading,
- higher and variable propagation delays than terrestrial RF channels,
- high bit error rates,
- temporary losses of connectivity,
- limited battery power and impossibility to recharge the batteries with solar energy,
- possibility to fail being exposed to fouling and corrosion [1].

Due to all of these reasons, both network topology organization and protocol stack organization is important. In the underwater environment, placement of sensor nodes which are static or mobile may be modeled in two or three dimensions according to the application. There usually is a surface sink station to gather the information from the network in underwater and to transfer it to an outside terminal which can be an onshore station or a satellite. Moreover, one or more mobile autonomous underwater vehicles (AUVs) may be deployed in the network in order to realize some missions such as taking the position data from GPS, supporting coverage in the case of connectivity loss and controlling the sampling rate according to operating conditions.

2.1.1 Acoustic Sensor Nodes

An acoustic sensor node is composed of some basic parts which are CPU/ onboard controller, acoustic modem, sensor, memory and power supply [1]. Sensors collect data and send it to the CPU. Then, the data can be stored in the memory or send to another node in the network by acoustic modem. Storing data in memory may be desired in the case of interruptions in the communication caused by the nature of underwater environment. Acoustic sensor nodes are expensive devices compared to terrestrial sensor nodes because acoustic transceivers have a more complex structure than RF transceivers, and acoustic sensor nodes require a hardware protection mechanism from harsh underwater conditions. Due to expensiveness, they can be deployed sparse, not densely as in the case of terrestrial sensor nodes. Moreover, acoustic sensor nodes consume more energy than terrestrial sensor nodes because of higher distances and more complex signal processing at the receivers. To reduce energy consumption, they may spend most of the time asleep. However, in [3] energy problem is solved with remotely powered underwater acoustic sensor networks (RPUASNs).

2.1.2 RPUASN Nodes

In the literature, energy efficiency problem for both terrestrial and underwater acoustic sensor networks has been discussed extensively. Many energy efficient MAC or routing layer algorithms have been proposed. For example, putting the sensor node into sleep mode and awake mode in cycles, namely duty cycling is a popular method. However, such methods are suitable for delay tolerant applications. It is important to keep sensor nodes mostly in awake mode for time critical applications. On the other hand, passive underwater sensor networks which harvest energy from environment overcome energy efficiency problem so that they can be used in such applications. Remotely Powered Underwater Acoustic Sensor Network (RPUASN) which is proposed in [3] is a kind of passive underwater sensor network in which the sensor nodes can harvest energy from a remote acoustic energy source. In a RPUASN node, a harvesting and power unit exist in addition to the other units such as communication and processing units. In the harvesting unit, an array of n hydrophones is available to produce electricity from acoustic signals emerging from the acoustic energy source. In the power unit, a dc converter is used to obtain the necessary voltages for the node and a storage capacitor is used to store the energy coming from the dc converter. In [3], the feasibility study of the proposed network model and its performance investigation study in terms of coverage and connectivity are presented.

Although RPUASN is a novel concept, a protocol stack proposal and performance study for it has been given in [4]. More information about the proposed protocol which is cross layer power adaptive CSMA/CA (X-PACCA) protocol is given in section 2.1.4

2.1.3 Protocol Stack for UASN

Protocol stack for underwater sensor nodes is basically composed of physical layer, data link layer, network layer, transport layer and application layer. Furthermore, there are three planes for the stack, which are power management plane aiming to minimize energy consumption, coordination plane providing coordinated works such as data aggregation and localization plane providing localization information for sensor nodes when necessary [1]. It is necessary to design all layers in an appropriate way to underwater channel characteristics.

Conducting communication with acoustic channels is a challenging issue so physical layer considerations for UASN is important. An acoustic system can operate in the frequency range of 10kHz and 15kHz and the total available bandwidth is only 5kHz [6]. Speed of sound is also very low, around 1500m/s. It varies with temperature, salinity and density to a significant extent. Due to these variations sound waves propagate on curved paths [2]. Transmitter may become inaudible because of this. Attenuation in underwater environment is heavily dependent on communication frequency. If frequency increases, loss also increases.

Noise is also a significant problem for acoustic communication. The dominant noise source in the frequency range of 200Hz and 50kHz is the wind acting on the sea surface [2].

Multipath effect for acoustic channel is caused by two basic reasons, namely sound reflection and sound refraction. Sound reflection occurs at the surface, bottom and objects in underwater. Sound refraction is caused by sound speed variation with depth which becomes more significant for deep water channels.

Another problem in the physical layer is Doppler effect caused by the motion of transmitter and receiver. In fact, this problem is an inevitable one because there is always a motion in underwater caused by waves, currents and tides. The basic problem of Doppler effect to the network is difficulty in the design of synchronization algorithms.

Energy efficient communication is very important for UASN and it can be provided by some ways. For example, transmitting at a higher bit rate saves a significant amount of energy [6]. High bit rates also reduce collision rate. Another way is reducing the number of retransmissions.

For data link layer, many MAC protocols have been proposed for underwater acoustic sensor networks being inspired of terrestrial MAC protocols. However, noticeably the recent research takes into account most of the challenges of the acoustic communication channel when designing a protocol for underwater. The challenges concerning with acoustic channel such as limited bandwidth, high and variable delays, multipath and fading problems, high bit error rates, connectivity losses requires special protocol design considerations. Moreover, limited battery power operation of sensor nodes necessitates an energy efficient MAC protocol. To achieve this aim, minimizing collisions as possible is important because retransmissions reduce energy efficiency considerably. Also, some protocols based on low duty cycles try to decrease energy consumption by causing the sensor nodes to go to sleep mode periodically. In this scheme, neighbor nodes should know the schedules of each other in order to wake up to receive the packet that a neighbor node is transmitting. Therefore, synchronization is necessary for such protocols but providing global schedules is very difficult for underwater acoustic channels due to high and variable delays. However, locally synchronized schedules can be a solution as in the case of [7]. MAC protocols proposed for underwater acoustic sensor networks in recent research can be divided into two groups, which are scheduled protocols such as TDMA and contention-based protocols such as CSMA. Table 2.1 indicates the MAC protocols proposed for UASN.

Table 2.1: UASN MAC Protocols

Protocol	**Classification**	**Main Characteristics**	**Specific Aim**
UWAN-MAC [7]	Schedule based	• Requires synchronization • Low duty cycles • Suitable for delay tolerant applications	To minimize energy consumption
STUMP [9]	Schedule based	• Uses propagation delay information to improve scheduling	To increase channel utilization
DACAP [13]	Contention based	• Uses RTS/CTS handshaking • Waiting period to avoid collisions	To increase throughput by minimizing collisions
COPE-MAC [11]	Contention based	• Uses broadcast channel reservation packet instead of RTS/CTS handshaking • Uses cyber carrier sensing technique by mapping physical channel to virtual one	To increase throughput

Table 2.1 (cont'd)

Protocol	**Classification**	**Main Characteristics**	**Specific Aim**
Slotted FAMA [18]	Schedule based	• Uses RTS/CTS handshaking • Uses carrier sensing	To increase throughput by avoiding collisions
BTB-TDMA [10]	Schedule based	• Uses propagation delay information • Designed for mobile nodes • Gives time bounds for data transfer to decrease delay	To provide better channel utilization
SF-MAC [19]	Contention based	• Uses RTS/CTS handshaking	To overcome fairness problem
DOTS [20]	Both contention and schedule based	• Uses RTS/CTS handshaking • Overhearing propagation delay and transmission schedule of neighbor nodes to provide concurrent transmissions	To provide better channel utilization
NAMAC [21]	Contention based	• Noise aware protocol • Uses RTS/CTS handshaking	To overcome noise factor by switching the frequency band in multi-band modems
HRMAC [22]	Both contention and schedule based	• hybrid solution by combining channel reservation and scheduling • Spectrum spreading technology is used	To decrease the end-to-end delay

Table 2.1 (cont'd)

Protocol	**Classification**	**Main Characteristics**	**Specific Aim**
CBTDB [23]	Schedule based	• cluster based model • avoids to use RTS/CTS or ACK packets	To improve channel throughput by reducing collisions
T-Lohi [24]	Contention based	• tone based reservation • designed for single hop contention • promote fairness	To provide energy efficiency
BiC-MAC [25]	Contention based	• Uses RTS/CTS handshaking • After channel reservation bidirectional data transmission in terms of bursts	To improve channel utilization
TLPC [26]	Contention based	• Uses RTS/CTS handshaking • Uses two power levels to avoid collisions	To overcome control/data collision and large interference range collision problems

There are also some other studies evaluating the performances of MAC protocols proposed for UASN. In [14], an effective network density to minimize energy consumption is explored for low duty cycle underwater MAC protocols. Acoustic channel is modeled by taking into account losses and effective number of nodes which is the number of nodes awake at a time instant. It is shown that there is an optimal effective node density to minimize the energy consumption in a low duty cycle MAC protocol. Also, an optimal duty cycle can be found in a given network density because $\rho_{eff} = \alpha\rho$ where α is the duty cycle and ρ is the actual network density. Therefore, this study [14] can shed light on design of low duty cycle MAC protocols for underwater networks both to provide minimum energy consumption and to see the relation between some performance metrics such as delay, reliability, collision rate and effective node density. Another important study realized about underwater MAC protocols is [8] which is the first study providing an extensive comparison between underwater MAC protocols with at-sea experiments. The results of these experiments are important to observe the difference between simulation and real case. In [8], it is very significant that there is a considerable gap between simulation and at-sea experiment results. The basic reason for this difference is that simulations do not include acoustic modem limitations and delays. Even if these overhead and delays are modeled in simulation, there is still some gap between actual case and simulation due to some practical effects of the experiments which cannot be easily included into the simulations. However, taking into account acoustic modem limitations and delays is important to get simulation results close to the actual case in terms of throughput and packet delay.

For network layer, the proposed routing protocols for terrestrial ad hoc networks are not useful due to mobility and scalability reasons. The terrestrial routing protocols can be grouped in three, which are proactive routing protocols, reactive routing protocols and geographical routing protocols. Proactive and reactive routing protocols are not suitable because of large signaling overhead. Thus, different routing solutions are proposed in the literature as in Table 2.2.

Table 2.2: UASN Routing Protocols

Protocol	**Classification**	**Main Characteristics**	**Specific Aim**
SEANAR [5]	Energy efficient and topology aware routing	• Routing decisions are made with degree information of neighbor nodes • A greedy approach	To provide energy efficiency
QELAR [27]	Energy efficient and adaptive routing	• Even distribution of residual energy of sensor nodes	To prolong the lifetime of the network

Table 2.2 (cont'd)

Protocol	**Classification**	**Main Characteristics**	**Specific Aim**
VAPR [28]	Pressure routing	• Uses depth information • Greedy directional forwarding	To provide efficient routing in the presence of voids
UHRP [29]	Hybrid routing	• Support transmissions from both localized and unlocalized nodes • Reduces routing overhead	To maintain routing in the presence of unlocalized regions
DCR [30]	Geographic routing	• Provides depth control of nodes for topology control	To improve network connectivity
DBR [31]	Depth based routing	• No need for full location information and localization service • Suitable for dynamic topology • Requires inexpensive depth sensors	To provide routing without localization
VBF [32]	Energy efficient routing	• Vector based forwarding • Location information is used • Redundancy is supported	To provide energy efficient, scalable and robust routing
AURP [33]	AUV aided routing	• Uses AUV (autonomous underwater vehicle) to collect data from gateway nodes and forward it to the sink node	To provide high delivery ratio and low energy consumption
HydroCast [34]	Pressure based routing	• Requires pressure sensor for depth information • Limits co-channel interference	To provide reliable data transport

2.1.4 CSMA

CSMA (Carrier Sensing Multiple Access) which provides a basis for many MAC protocols is a widely-used protocol for UASN studies. The flow chart for this protocol implemented according to the description in [8] is as in Figure 2.1. The algorithm is as follows. When a packet is desired to be sent, the channel is examined whether it is idle or not. If it is idle, the packet is sent. If it is busy, a random backoff delay is applied. The backoff delay is randomly chosen in [0, T] where T = $2^{txRetry}$(2maxDelay + dataTime + ackTime). After sending the data packet, ACK is waited until the time (2propDelay + ackTime) where propDelay is propagation delay between the sender and receiver. If ACK is not received in this period, the packet is retransmitted by choosing backoff time in twofold of the previous transmission. If the channel can not be accessed or the packet can not be transmitted successfully until the maximum number of retries, the packet is dropped.

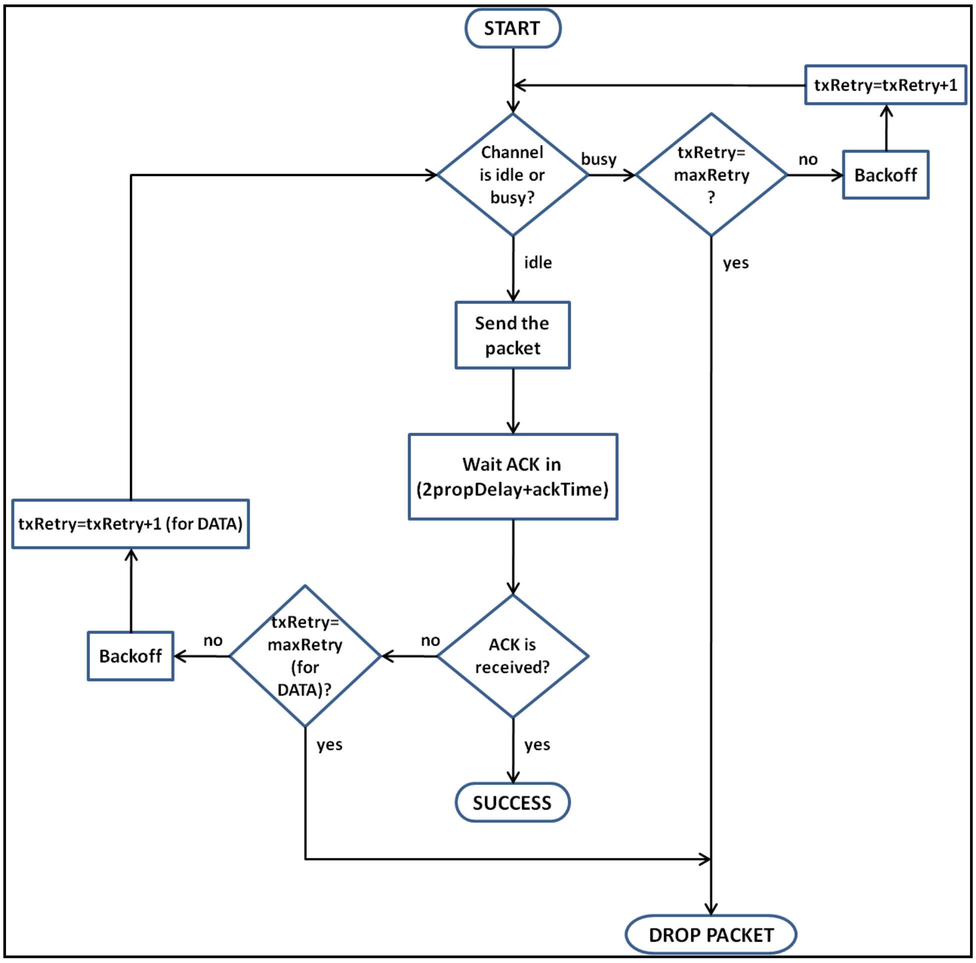

Figure 2.1: Flowchart of CSMA protocol

2.1.5 DACAP

DACAP (Distance Aware Collision Avoidance Protocol) which is proposed for UASN in [13] is an RTS-CTS handshake based MAC protocol. Figure 2.2 shows the flowchart of DACAP protocol, which is formed according to the description in [8]. The algorithm is as follows. When a packet is desired to be sent, the channel is examined whether it is idle or not. If it is idle, RTS packet is sent. If it is busy, a random backoff delay is applied. The receiver node sends a CTS packet and waits for the data transmission. After receiving the CTS packet, the sender node waits a specific time which is T_w. During this waiting period, if the receiver node receives an RTS packet from another node, it sends a warning packet to the sender node. If the sender node receives a warning packet, it postpones the transmission according to an exponential backoff mechanism. If the channel can not be accessed or the packet can not be transmitted successfully until the maximum number of retries, the packet is dropped. In [13], two versions of DACAP protocol are described as with and without acknowledgement packet. In [8], the version is not mentioned so the version with ACK is assumed.

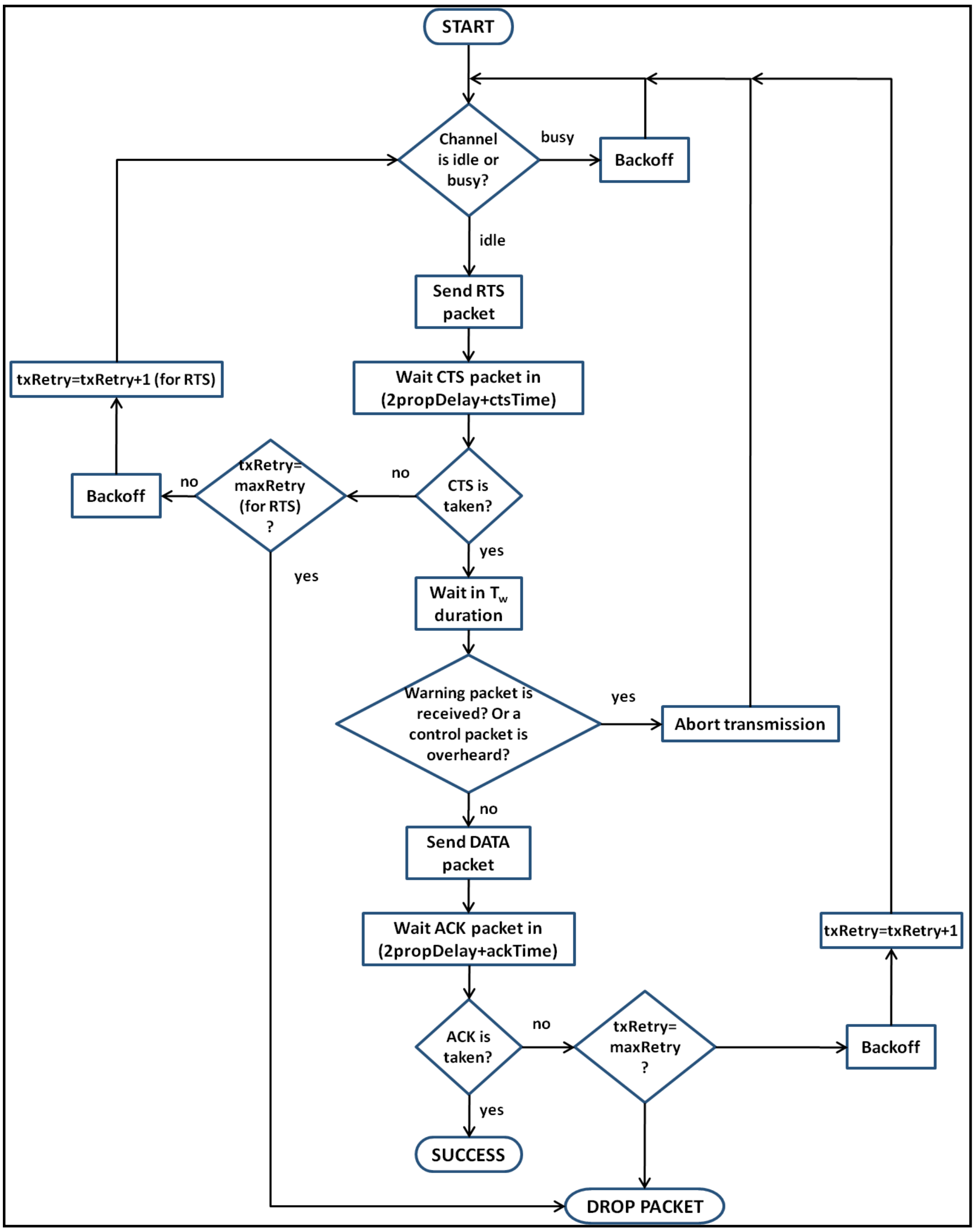

Figure 2.2: Flowchart of DACAP protocol

2.1.6 Cross Layer Power Adaptive CSMA/CA

In [4], a protocol stack, namely Cross Layer Power Adaptive CSMA/CA (X-PACCA) has been proposed for Sink Powered Underwater Acoustic Sensor Networks (SPUASN) which is a specific sort of RPUASN. In SPUASN, the sink node is also the acoustic energy source supplying energy to the sensor nodes. X-PACCA protocol which takes advantage of this configuration gives the nodes closer to the sink more priority than the farther nodes in order to access to the channel. It is achieved by adjusting the window size of relay nodes inversely proportional to the harvested power and giving the transmission chance of a packet to the relay node which has more harvested power than the previous node. In addition, the packets processed or relayed by a node are recorded for a specific time so retransmission of the same packet is prevented. Thus, a more efficient data transmission is realized by preventing redundant paths and retransmissions. Moreover, end-to-end delay decreases because the nodes closer to the sink will have more harvested power. Also, success probability increases due to prevention of redundant paths.

Another good side of X-PACCA protocol is to provide end-to-end reliability by acknowledgement packets sent by the sink node. This is also useful to prevent the retransmissions of the same packet by the other nodes.

Due to all of these properties of X-PACCA protocol, it supports MAC, network and transport layer functions. Furthermore, it is appropriate for harsh underwater conditions due to avoiding redundant transmissions and reducing congestion.

In [17], the impacts of several parameters on the performance of X-PACCA protocol are investigated in appropriate scenarios by using spherical cone deployment in terms of average end-to-end delay and success probability. The investigated parameters are window size, slot length, backoff constant (k), vertex angle and node density. In addition, the MAC and routing layer performances of X-PACCA are investigated by a comparison study with two UASN MAC protocols which are UWAN-MAC and Slotted FAMA and two routing layer protocols DBR and VBF. In this study, as distinct from [17] the impacts of WFA threshold value and PER value on the performance of X-PACCA protocol in spherical cone deployment are explored in terms of average end-to-end delay and success probability. Also, its performance in spherical deployment is examined by deploying different number of nodes in the same volume. Finally, its MAC performance is compared with CSMA and DACAP protocols in terms of average end-to-end delay and receive throughput.

CHAPTER 3

SIMULATION STUDIES WITH THE DEVELOPED SIMULATOR

3.1 Simulation and Comparison Study of CSMA and DACAP

CSMA and DACAP protocols which hold an important place in the literature of underwater acoustic sensor networks are selected for a simulation and comparison study in this paper. The reason for choosing them is that they are contention based protocols which have more appropriate access mechanism for underwater channel than scheduled protocols or FDMA. To compare performance metrics of these protocols, in the scope of the present study, an event-based simulator has been developed using MATLAB for underwater acoustic sensor networks. To test the reliability of the simulator, some simulation studies were realized by applying the same simulation scenarios in article [8] and comparing throughput efficiency and latency values. In this section the modeled simulator is described first, then the simulation scenarios and parameters are given. Finally, the results obtained are presented and discussed.

3.1.1 Simulator Developed Using MATLAB for UASNs

The main difference between an acoustic network model and a terrestrial network model is due to the physical layer modeling. Propagation delays are negligible for radio channel while they are significant for acoustic channel in terms of both the low speed of sound and attenuation with frequency. The speed of sound is taken as 1500 m/s. Thorp's formula is used for absorption coefficient which is α(f) in dB/km for f in kHz:

$$10\log\alpha(f) = \frac{0.11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + \frac{2.75f^2}{10^4} + 0.003 \tag{3.1}$$

Attenuation in underwater acoustic channel over a distance d for a signal of frequency f is given by:

$$A(d,f) = d^k\alpha(f)^d \tag{3.2}$$

where α(f) is absorption coefficient and k is spreading factor. k is taken as 1.5 which is the practical spreading value while k = 1 stands for cylindrical spreading and k = 2 stands for spherical spreading.

Noise is another important physical layer consideration for underwater acoustic networks so it was also modeled in the simulator. Ambient noise in underwater channel is caused by four factors which are turbulence, shipping, waves and thermal noise. For different frequency ranges the dominant noise factor is different. At frequencies lower than 10Hz the main noise source is turbulence. The shipping factor becomes dominant in the frequency range 10Hz-100Hz. Waves caused by wind are the key noise contributor in the frequency range 100Hz-100kHz. At frequencies higher than 100kHz, the major noise factor is thermal noise. The power spectral density for these noise factors in dB re uPa per Hz as a function of frequency in kHz are as follows respectively:

$$10\log N_t(f) = 17 - 30\log f \tag{3.3}$$

$$10\log N_s(f) = 40 + 20(s-0.5) + 26\log f - 60\log(f + 0.03)$$

$$10\log N_w(f) = 50 + 7.5\sqrt{w} + 20\log f - 40\log(f + 0.4)$$

$$10\log N_{th}(f) = -15 + 20\log f$$

The total ambient noise is:

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \tag{3.4}$$

Wind speed is taken as 4 m/s and shipping parameter s is taken as 0.5 for simulations.

In addition to attenuation and noise modeling, interference is also modeled in the simulator. If a packet is received with the signal to interference ratio which is lower than the threshold value, it means a collision and the packet should be retransmitted. Also, if a packet is received with the signal to noise ratio which is lower than the threshold value, it means the packet could not be taken by the receiver correctly, and it should be retransmitted. The threshold values for SIR (signal to interference ratio) and SNR (signal to noise ratio) are as follows:

SIR threshold = 15 dB
SNR threshold = 20 dB

Topology used in the simulator is static network model. Nodes can be distributed to the known positions or random points in a 3D region. Traffic is generated according to a Poisson process and each generated packet is sent to a destination which is randomly chosen among all nodes. Single hop scenarios were implemented in which nodes can communicate to each other directly.

Further information about the simulator can be found in Appendix A.

3.1.2 Simulation Study for CSMA

CSMA MAC protocol was modeled using the MATLAB simulator and the single hop scenario in [8] was implemented. In this scenario, there are 3 source nodes and a sink node whose locations are known and static. Figure 3.1 shows the locations of the nodes in 2D in our simulation study. Figure 3.2 shows the node locations in [8] and in the single hop scenario M1, M2 and M3 are source nodes while GB1 is the sink node.

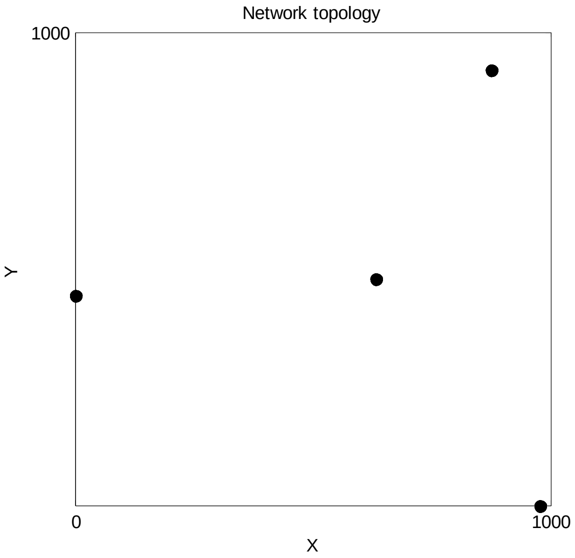

Figure 3.1: Node locations in 2D

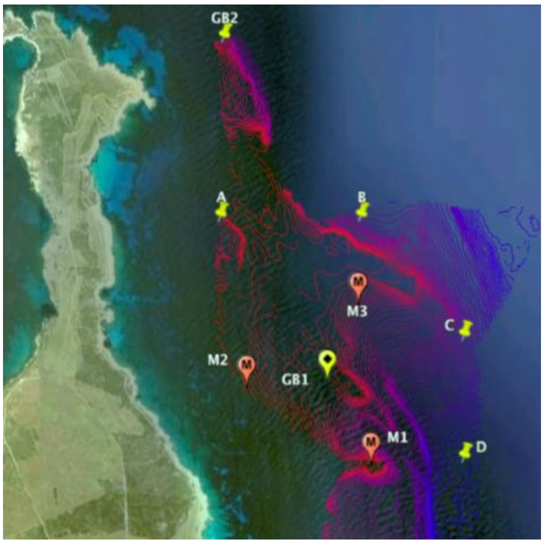

Figure 3.2: Node locations in [8]

Traffic is generated according to a Poisson process with network-wide rate λ packets per seconds. Performance graphics are obtained in terms of packets per packet time which is $\lambda' = \lambda T_{data}$. T_{data} is the transmission delay of a data packet and λ' is varied between 0.08 and 0.22 packets per packet time. The parameter values which were set according to [8] are as in Table 3.1.

Table 3.1: Parameter values for CSMA evaluation

Parameter	**Value**
Carrier frequency	24 kHz
Transmission bit rate	80bps
Transmission power	180dB re 1uPa at 1m
Packet size	32Bytes
ACK size	13bits
Retransmission attempts	4
MAC queue size	50
Load size	0.08 - 0.22 packets per packet time
Simulation Time (s)	100,000 s

Investigated performance metrics for this protocol are throughput efficiency and latency which are defined as:

> *Throughput Efficiency: is the ratio of the successfully received packets to the total generated packets.*
>
> *Packet Latency: is the average time difference between the instant a packet is generated and the instant it is received by the sink node in terms of seconds.*

The simulations are realized similar to the case in which additional acoustic modem delays are not taken into account and it is named as 'CSMA-SimNoDelay' case in [8]. Figure 3.3 show the comparison of throughput efficiency graphs obtained in the present study and in [8] . As can be seen from the figures, throughput efficiency values obtained using our simulator are very close to the ones of 'CSMA-SimNoDelay' case in [8].

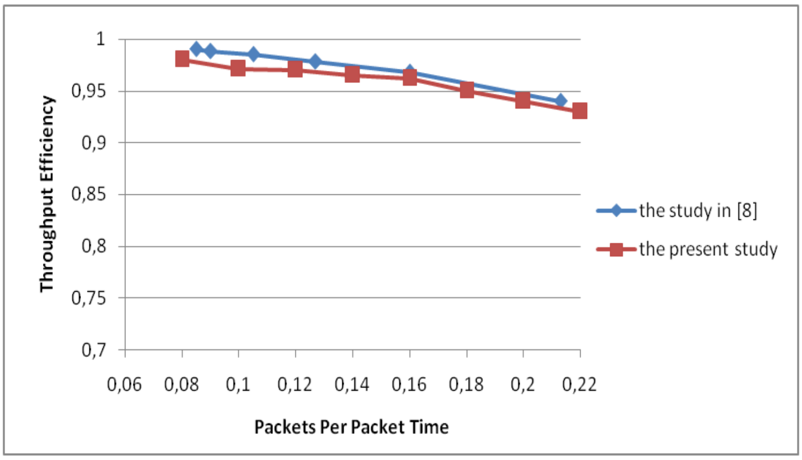

Figure 3.3: CSMA Throughput Efficiency

Figure 3.4 show the comparison of packet latency graphs obtained in our study and in [8]. As can be seen from the figures, packet latency values obtained using our simulator are very close to the ones of 'CSMA-SimNoDelay' case in [8].

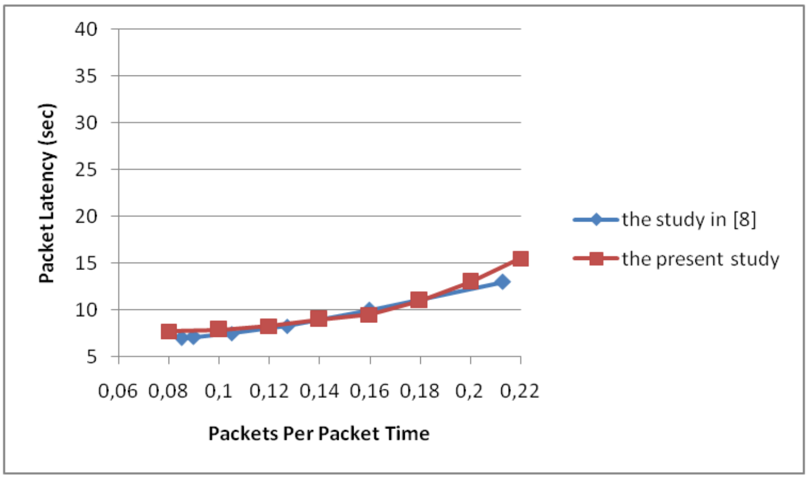

Figure 3.4: CSMA Packet Latency

There is a slight difference for throughput efficiency and packet latency values between our simulation results and the results of [8], which can be resulted from the different channel models used in the present study and in [8]. In the present study, Thorp approximation formulas are used to model propagation while in [8] Bellhop channel model is used. As stated in [35], Bellhop model which uses ray tracing method is a more accurate and complex propation model than Thorp approximation and the propagation models employed in simulations affect the performance of higher level protocols. Thus, it is expected to see some differences on the results.

3.1.3 Simulation Study for DACAP

The second protocol modeled using the MATLAB simulator is DACAP MAC protocol. The single hop scenario in [8] was implemented as described in section 3.1.2 with 3 static source nodes and a static sink node.

Traffic is generated according to a Poisson process with network-wide rate λ packets per seconds as in the case of the simulation study of CSMA protocol. However, $\lambda^{'}$ is varied between 0.08 and 0.14 packets per packet time in this case according to [8]. The parameter values for DACAP simulation are as in Table 3.2.

Table 3.2: Parameter values for DACAP evaluation

Parameter	**Value**
Carrier frequency	24 kHz
Transmission bit rate	80bps
Transmission power	180dB re 1uPa at 1m
Packet size	32Bytes
RTS/CTS/WARNING/ACK size	13bits
Retransmission attempts	7
MAC queue size	50
Load size	0.08 - 0.14 packets per packet time
Simulation Time (s)	100,000 s
T_{min}	T (T: maximum propagation delay)
T_{W_min}	0

Throughput efficiency and latency results are investigated with simulation studies.

The simulations are realized similar to the case in which additional acoustic modem delays are not taken into account and it is named as 'DACAP-SimNoDelay' case in [8]. Figure 3.5 and Figure 3.6 show the comparison of throughput efficiency and packet latency results obtained in our study and in [8]. As can be seen from the figures, throughput efficiency and packet latency values obtained using our simulator are very close to the ones of 'DACAP-SimNoDelay' case in [8]. However, there is an insignificant difference between the results. Reasons for this difference can be some chosen parameter values related to DACAP protocol such as T_{min}, T_{W_min}, the length of backoff times and waiting time. These values are not stated in [8], so some assumptions are made for them in our study. Also, the difference of channel models used in [8] and in our study can lead to some differences.

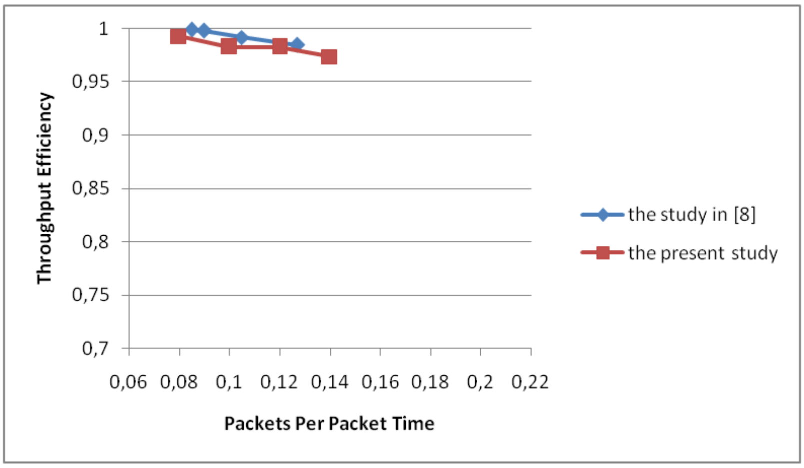

Figure 3.5: DACAP Throughput Efficiency

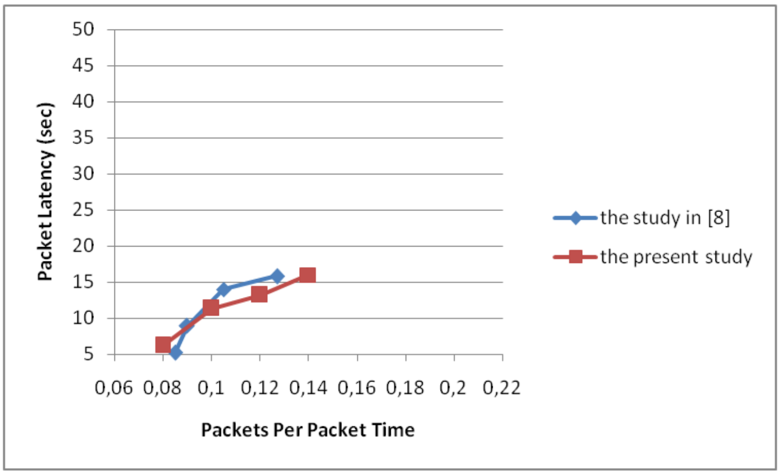

Figure 3.6: DACAP Packet Latency

CHAPTER 4

PERFORMANCE EVALUATION OF X-PACCA PROTOCOL

X-PACCA protocol which is proposed for SPUASN has been investigated in [4] and [17]. In the present study, X-PACCA protocol is implemented in the developed underwater MATLAB simulator to be able to realize some of the simulation studies in [4] and to extend the performance evaluation to provide more comprehensive results regarding the X-PACCA protocol.

In this section, all simulations are repeated until average delay values stay within ±10% confidence interval with 99% confidence level. Other simulation parameter results such as success probability are stated depending on the simulation repetitions executed for average delay.

4.1 The MATLAB Implementation of X-PACCA

To implement X-PACCA protocol in MATLAB, firstly the physical layer of the MATLAB simulator is adapted according to RPUASN environment by using the formulas mentioned in [3]. The harvested power is calculated using (4.5). Transmission power is determined according to (4.6) and it is assigned to the transmission power level label of data packets.

$$SL = 170.8 + 10\log_{10}P_{elec} + 10\log_{10}\eta + DI \tag{4.1}$$

$$DI = 20\log_{10}\frac{60\pi}{\theta} \tag{4.2}$$

$$AL = 20\log_{10}R + \alpha(f_s)R \tag{4.3}$$

$$RL = SL - AL \tag{4.4}$$

$$P_{harv} = 0.7n\frac{10^{(RL+RVS)/10}}{4R_p} \tag{4.5}$$

$$MYPL = \beta P_{harv} \tag{4.6}$$

where

SL: power level of the acoustic power source
RL: received power level by the sensor nodes
AL: attenuation power level from the acoustic power source to the sensor nodes
P_{harv}*: harvested power level by the sensor nodes*
MYPL: transmission power level of a data packet
P_{elec}*: electrical input power of the acoustic power source*
η*: electro-acoustic power conversion efficiency*
DI: directivity index of the acoustic power source
θ*: vertex angle of the spherical cone shaped power transmission*
R: distance between the acoustic power source and a source node in m
f_s*: transmission frequency of acoustic power*
$\alpha(f_s)$*: absorption coefficient in dB/m*
n: number of hydrophones
RVS: receiving voltage sensitivity of a hydrophone
Rp: hydrophone impedance
β*: transmission power level constant* $0< \beta<1$

Also, the following three queue structures related to X-PACCA protocol are implemented:

- MTQ (MAC Transmit Queue): The packets which will be transmitted are hold in this queue until transmission.
- IGS (Ignore to Send Queue): The IDs of packets which are processed or relayed are stored in this queue until a threshold time so that if the same packet is received again, it is ignored.

- WFA (Wait for ACK Queue): In this queue, data source nodes hold the packets waiting ACK after a packet transmission with timeout durations.

Moreover, packet reception code in physical layer is changed with PER_{req} implementation instead of SIR and SNR threshold value. When a packet is received by a node, BER is calculated after calculating SINR value using the formulas (4.7), (4.8) and (4.9), which are derived from [39], assuming BFSK modulation:

$$C = B \times \log_2(1 + SINR) \tag{4.7}$$

$$\frac{E_b}{N_0} = \frac{B}{C} \times SINR \tag{4.8}$$

$$BER = Q(\sqrt{E_b/N_0}) \tag{4.9}$$

$$PER = 1 - (1 - BER)^n \tag{4.10}$$

Packet error rate can be calculated from BER (bit error rate) value by (4.10) where n indicates the packet size. If the calculated PER is smaller than a threshold value PER_{req}, the packet is accepted as it is received successfully, otherwise it is ignored as it is assumed to be fully corrupted.

Secondly, the X-PACCA algorithms which are event sensing, packet reception and backoff algorithms mentioned in [4] are implemented in the MATLAB simulator. Also, the topology implementation is changed and spherical cone deployment is implemented where nodes are randomly distributed in a spherical cone with a R_{max} and θ (vertex angle) value. Figure 4.1 shows a sample of the 3D random deployment in spherical cone obtained from the MATLAB simulator.

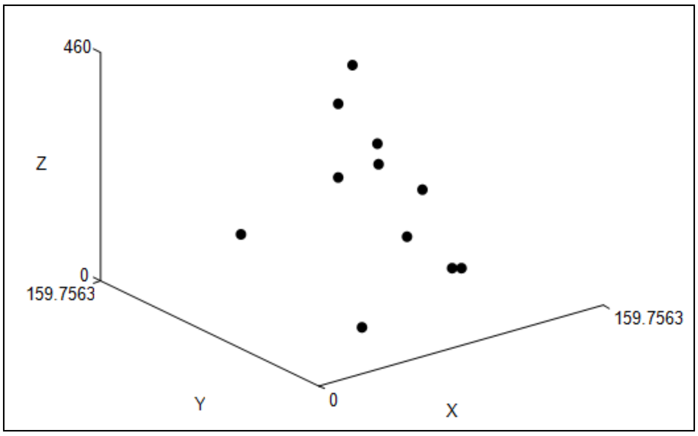

Figure 4.1: Randomly deployed nodes in spherical cone (10 source nodes, 1 sink node)

Another change applied to the MATLAB simulator is that the packet generation is adapted to the per node packet generation with Poisson arrivals. The main parameter values used in the simulations unless otherwise stated are as in Table 4.1.

Table 4.1: Main parameter values used in X-PACCA simulations

Parameter	Value
Communication frequency	25 kHz
Acoustic power source frequency	10 kHz
P_{elec}	10kW
Electro-acoustic power conversion efficiency	0.5
Transmission power constant (β)	0.025
Number of harvesting hydrophones	5
Receiving voltage sensitivity (RVS)	-150 dB re V/μPa
Hydrophone impedance	125Ω
Harvesting efficiency	0.7
Transmission bit rate	10kbps
ACK waiting time (WFA_{thresh})	20s
Ignore timeout (D)	50s
Receive threshold	8.7×10^{-8}W
Packet size	50Bytes
ACK size	24bits
Window size	3
Slot length (s)	0.005s
Retransmission attempts	5
MAC queue size	50
Spreading factor	2 (spherical spreading)

4.2 Comparison Studies of the MATLAB Model of X-PACCA

To compare the performance of MATLAB model of X-PACCA protocol with the model in [4], some of the simulation scenarios in [4] are implemented. The first scenario is the one related to the effect of source window size in [4]. The specific simulation parameters for this scenario as in Table 4.2.

Table 4.2: Parameter values for the comparison study

Parameter	**Value**
Vertex angle	20°
P_{req} (required power for a node)	2W
Backoff constant (k)	1
R_{max}	405m
Data Generation Rate Per Node	0.02 - 0.22 packets per seconds
Simulation Time (s)	3000 s
PER_{req}	0.98

The obtained average delay and success probability graphs are as in Figure 4.2 and Figure 4.3 respectively. Figure 4.4 and Figure 4.5 show the graphs obtained from the same simulation scenario in [4]. As can be seen from the figures, the simulation results display similar characteristics; however there are some differences. The reason for the differences can be the result of channel model or variable differences between the MATLAB simulator and AquaSim in [4]. In section 4.3, two possible reasons, which are PER_{req}(required packet error rate) and MAC queue size are investigated.

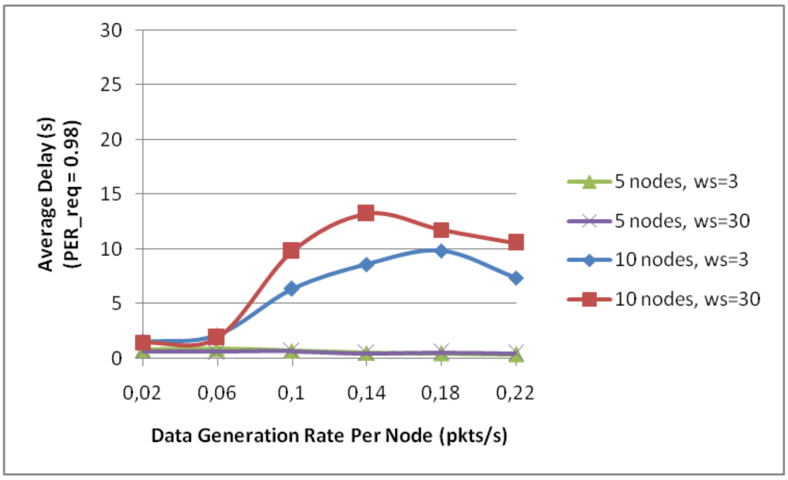

Figure 4.2: X-PACCA Average Delay Using the MATLAB Simulator

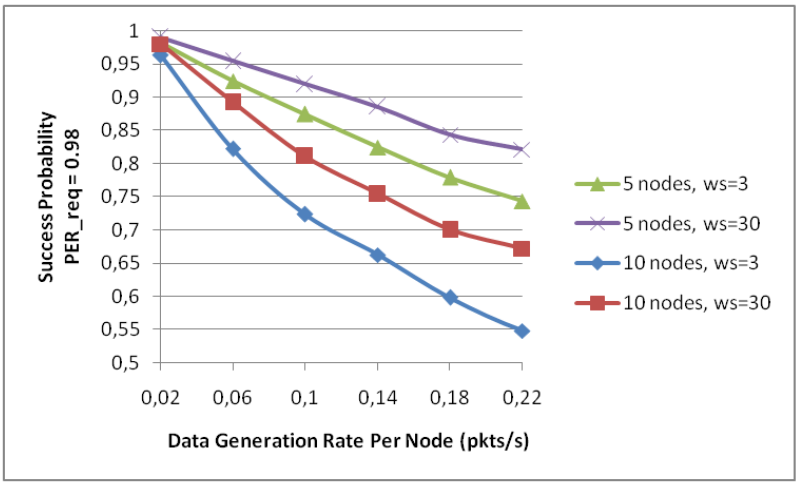

Figure 4.3: X-PACCA Success Probability Using the MATLAB Simulator

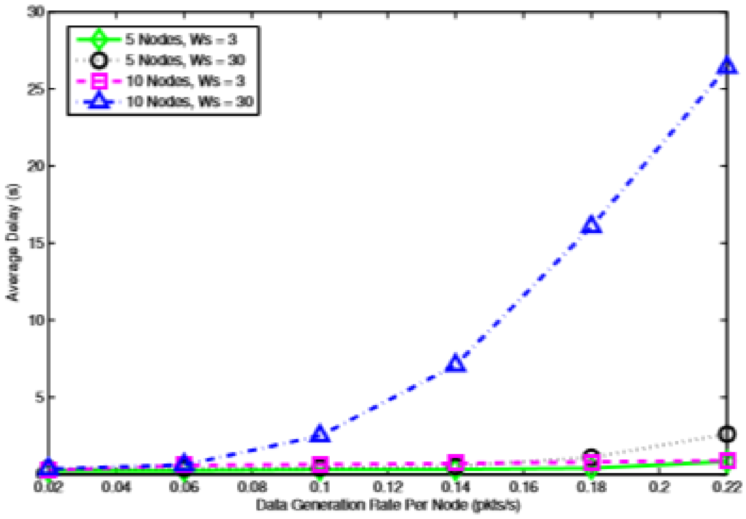

Figure 4.4: X-PACCA Average Delay in [4]

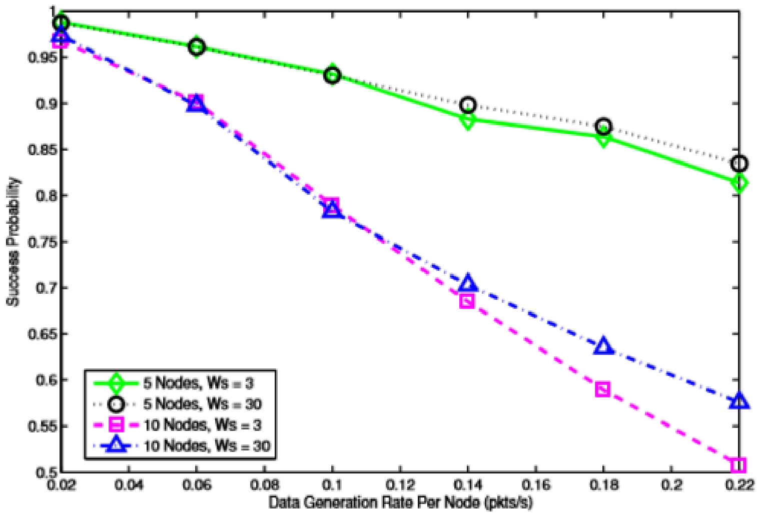

Figure 4.5: X-PACCA Success Probability in [4]

The second scenario is the one related to MAC performance evaluation. Simulation parameters for this comparison study are as in Table 4.3.

Table 4.3: Parameter values for MAC performance evaluation

Parameter	Value
Vertex angle	30°
Backoff constant (k)	0.1
P_{req}(required power for a node)	1W
R_{max}	380m
PER_{req}	$4*10^{-10}$
Data Generation Rate Per Node	0.02 - 0.2 packets per seconds
Active nodes	9
Simulation Time (s)	3000 s

The obtained receive throughput and average delay graphs for X-PACCA protocol are as in Figure 4.6 and Figure 4.7 respectively for the present study and the study in [4]. As can be seen from the figures, the simulation results are almost the same.

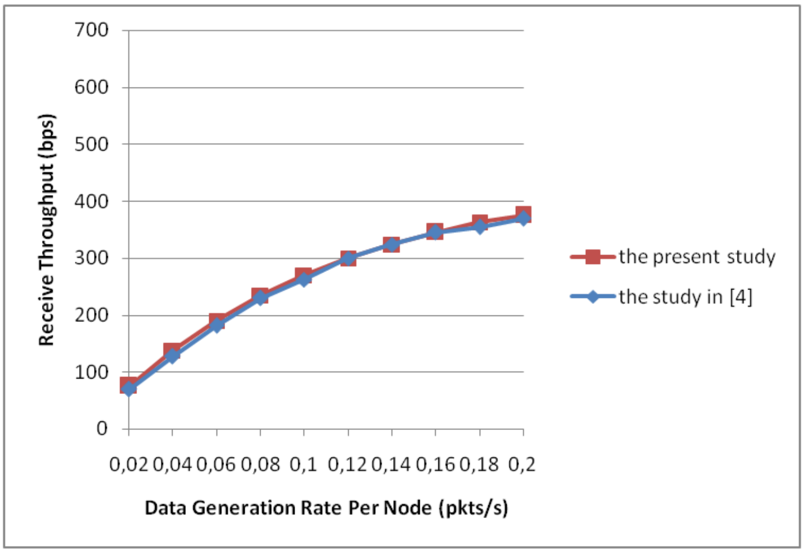

Figure 4.6: X-PACCA Receive Throughput

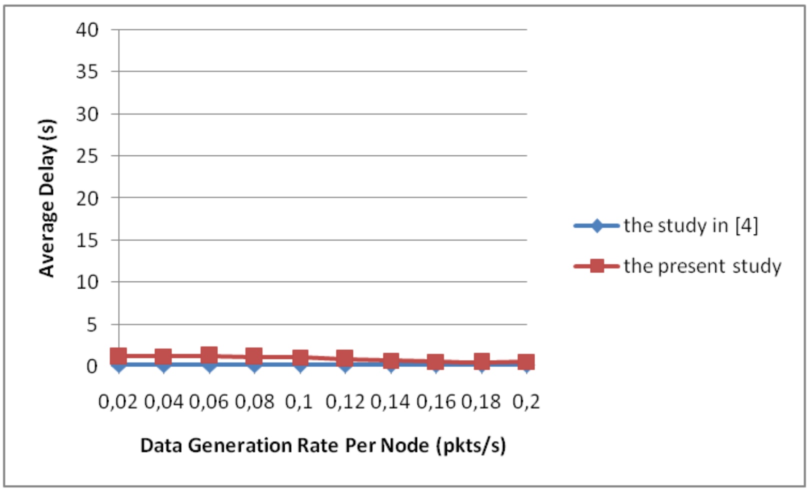

Figure 4.7: X-PACCA Packet Latency

4.3 Investigation of the Effect of PER Value

In this section, the effect of chosen PER_{req} (required packet error rate) value to the performance of X-PACCA protocol in a sample scenario has been investigated.

Choosing smaller PER_{req} values in the simulations means that network becomes more sensitive to the effect of packet collisions caused by interferences. On the other hand, choosing higher PER_{req} values means that the effect of interferences are almost neglected.

The chosen scenario is the one related to the examination of the effect of window size in [4]. Simulation parameters are as in Table 4.2. The three simulation studies have been realized for the cases that BER_{req} value is chosen as 10^{-6}, 10^{-9} and 10^{-12}, which corresponds to PER_{req} values of $4*10^{-4}$, $4*10^{-7}$ and $4*10^{-10}$ respectively. The MAC buffer size is chosen as 50 for these simulations. The results for this study are as follows:

- Figure 4.8, Figure 4.10 and Figure 4.12 show the average delay graphs for the cases when PER_{req} is $4*10^{-4}$, $4*10^{-7}$ and $4*10^{-10}$ respectively and MAC buffer size is 50. As can be seen from the figures, average delay values become smaller when PER_{req} gets smaller. The reason for this relies on the fact that pending packets are ignored when a new packet is generated. When PER_{req} value gets smaller, the status of more packets become pending because interferences become more significant. Thus, the average delay drops when the pending packets are discarded due to newly generated packets. The related pseudo-code leading this result can be found in event-sensing algorithm of X-PACCA protocol in [4]. In the algorithm, events in WFA queue are checked after timeout duration if they are still active or not. If the pending packet is still active, it is retransmitted. However, if a new event packet exists to be transmitted, the pending packet is deleted. Also, relay traffic decreases with decreasing PER_{req} value because packets can not be easily received by the nodes with increasing sensitivity to interferences. Therefore, decreasing overall load leads to decrease in average delay.

- However, success probability values do not display a significant change as can be seen from the figures which are Figure 4.9, Figure 4.11 and Figure 4.13. The reason for this is that MAC buffer size is chosen as 50 and it limits the number of packets sent at higher loads.

Another experiment is realized for the case when PER_{req} is $4*10^{-10}$ and MAC buffer size is infinite. The results for this study are as follows:

- Average delay values increases for the high data rate values as in Figure 4.14 when compared to the results of the previous studies. This is resulted from the fact that MAC buffer fills excessively when the data rate increases and if the size of buffer is finite old packets are discarded when the buffer is full. When the packets are not discarded due to infinite buffer size average delay increases due to the old packets waiting in the queue.

- However, success probability does not change so significantly, only it decreases slightly. The reason for this is that when the buffer is infinite, the number of packets in the network is high and the collision rate is high so the number of dropped packets is high. When the number of dropped packets increases, success probability decreases. However, the decrease in Figure 4.15 is not so significant because data rate is not high enough.

- When MAC buffer is chosen as infinite, the average delay graph (Figure 4.14) becomes more similar to the graph obtained in [4] (Figure 4.4). MAC buffer size in [4] is not stated in the paper, it may have been chosen as infinite or a higher size.

To see the effect of PER_{req} value on success probability more clearly, a simulation study is realized by choosing the value of window size as 3 and employing 10 active nodes. Also, MAC queue size is chosen as infinite to prevent the limitation on the number of sent packets. For PER_{req} values of 0,98 (BER_{req} 10^{-2}) and $4*10^{-10}$ (BER_{req} 10^{-12}) success probability and throughput efficiency characteristics have been studied. As can be seen from Figure 4.16 and Figure 4.17, the results differ from each other at higher packet rates while they are more similar for lower rates. Success probability and throughput efficiency values for PER_{req} value of $4*10^{-10}$ are lower than that values for threshold value of 0,98 because interferences gain importance for small PER_{req} values while they are almost neglected for higher PER_{req} values.

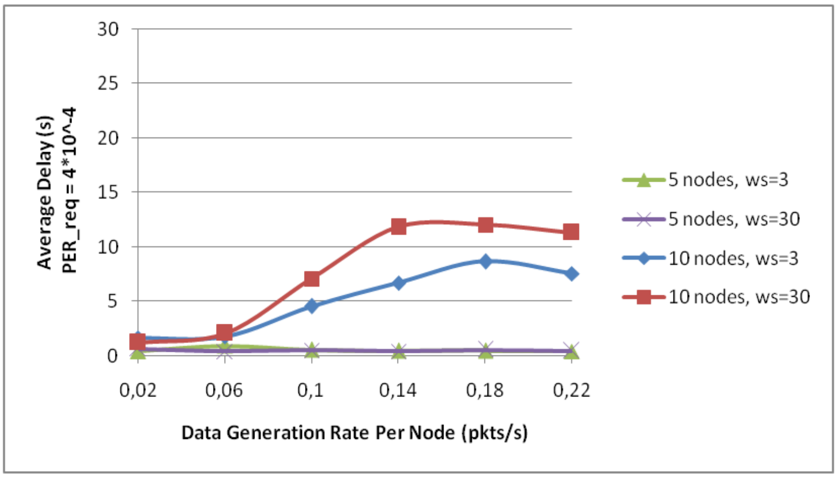

Figure 4.8: X-PACCA Average Delay when BER_{req} is 10^{-6}

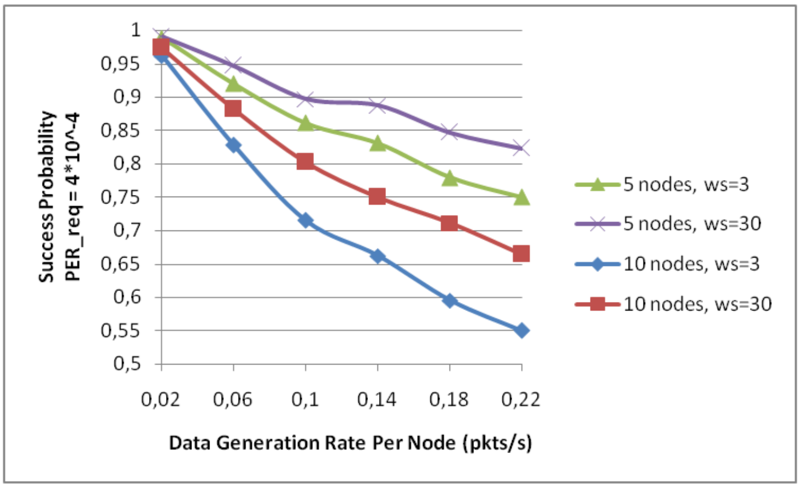

Figure 4.9: X-PACCA Success Probability when BER_{req} is 10^{-6}

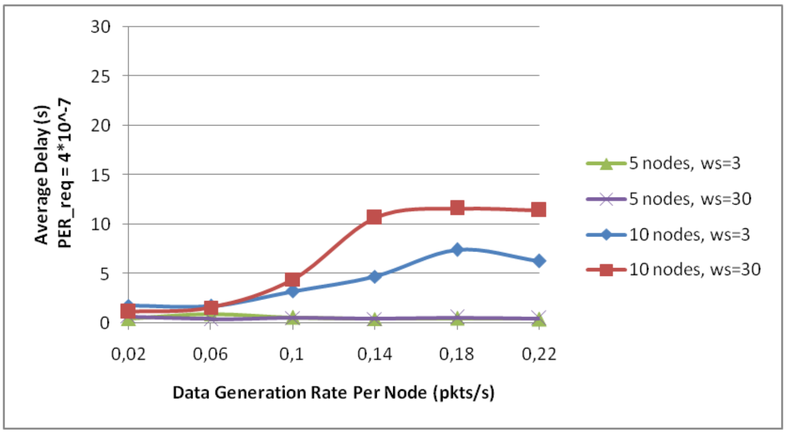

Figure 4.10: X-PACCA Average Delay when BER_{req} is 10^{-9}

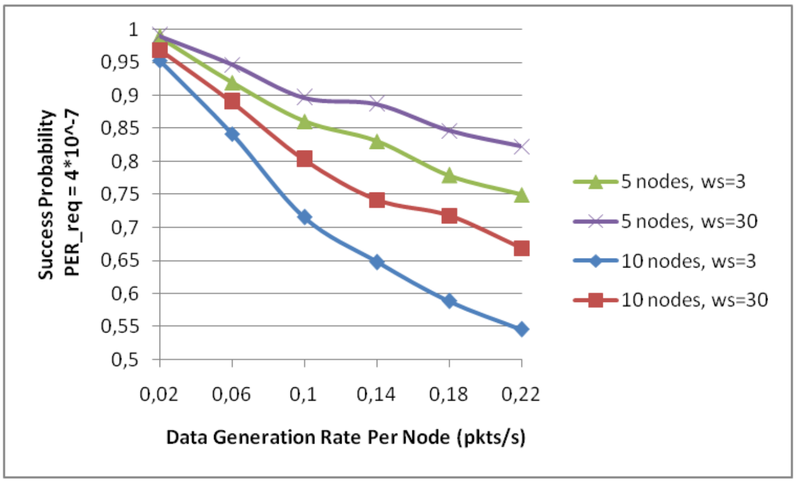

Figure 4.11: X-PACCA Success Probability when BER_{req} is 10^{-9}

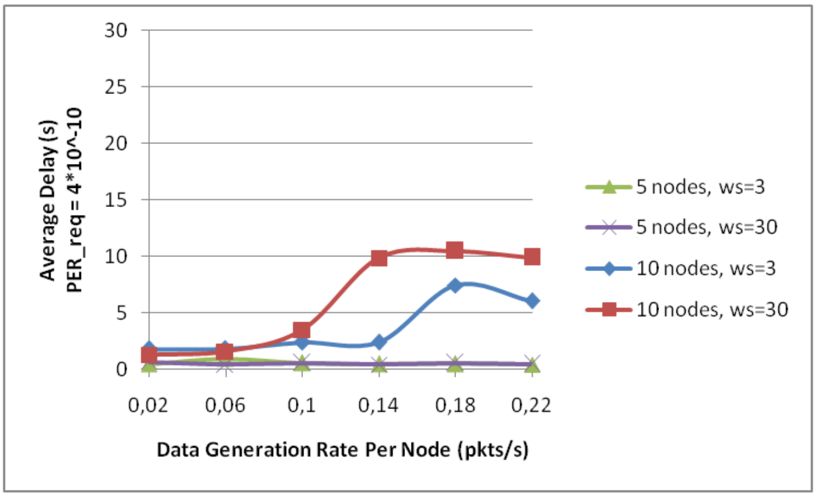

Figure 4.12: X-PACCA Average Delay when BER_{req} is 10^{-12}

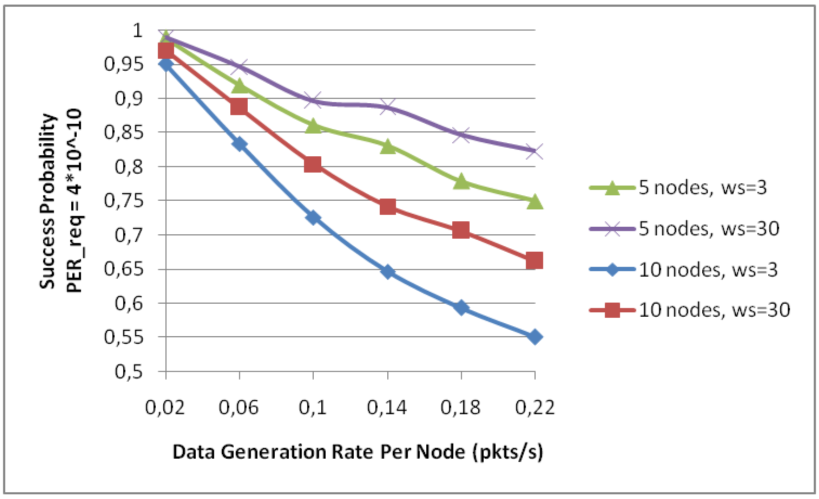

Figure 4.13: X-PACCA Success Probability when BER_{req} is 10^{-12}

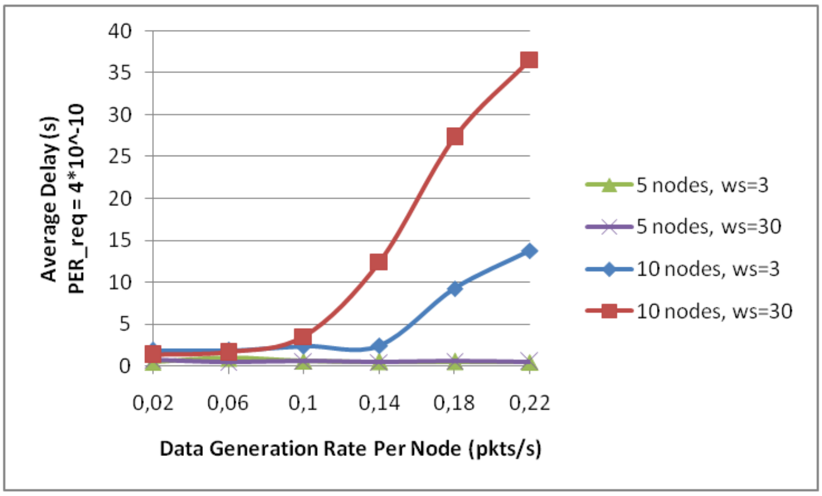

Figure 4.14: X-PACCA Average Delay when BER_{req} is 10^{-12} and MAC queue size is infinite

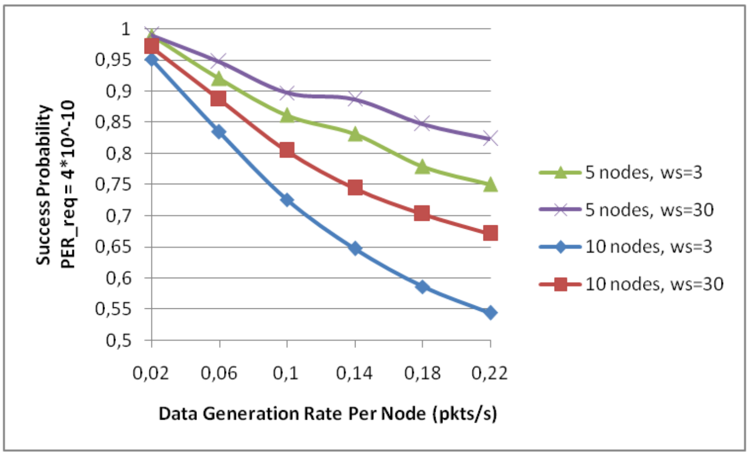

Figure 4.15: X-PACCA Success Probability when BER_{req} is 10^{-12} and MAC queue size is infinite

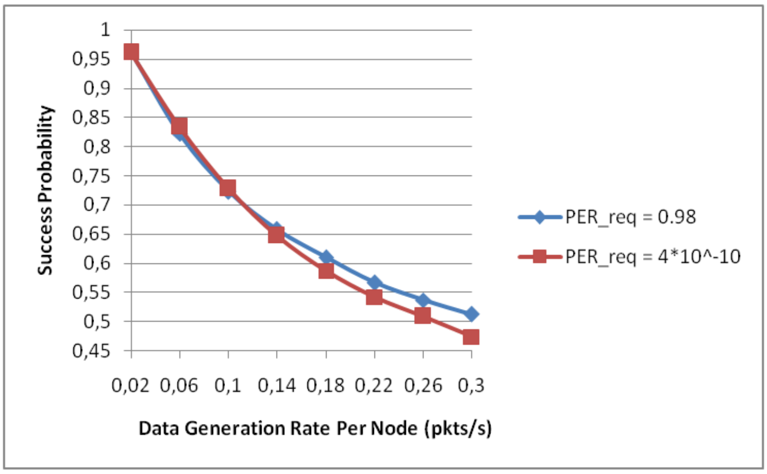

Figure 4.16: X-PACCA Success Probability for comparison of PER values

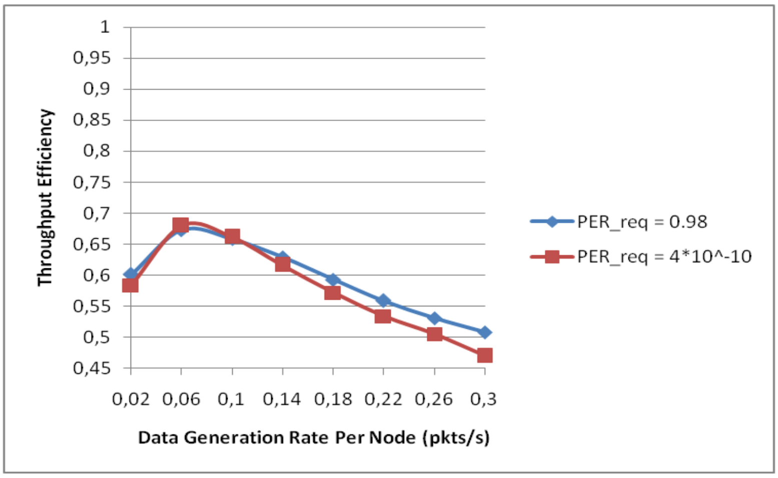

Figure 4.17: X-PACCA Throughput Efficiency for comparison of PER values

4.4 Evaluation of the Effect of WFA Threshold Value

In this section, the effect of WFA threshold variation on X-PACCA protocol performance is evaluated. The investigated performance metrics are average delay, success probability and receive throughput. 4 or 9 active nodes randomly distributed in a spherical cone with the vertex angle of 30^o produce packets according to a Poisson arrival process. The nodes are at such a distance away from the sink that the transmitted packets can reach the sink at a single hop. Simulation parameters for this study are as in Table 4.4.

Table 4.4: Parameter values for WFA threshold variation study

Parameter	**Value**
Vertex angle	30°
P_{req}(required power for a node)	0.5W
Backoff constant (k)	0.1
R_{max}	395m
PER_{req}	0.98
Data Generation Rate Per Node	0.02 - 0.2 packets per seconds
Active nodes	4 or 9 nodes
WFA Threshold	2s, 20s, 40s
Simulation Time (s)	3000 s

Figure 4.18, Figure 4.19 and Figure 4.20 displays the average delay, success probability and receive throughput performance metrics respectively. Inferences about the results are as follows:

- Average delay does not change significantly. It remains below 2 seconds for all the simulations. A small increase in average delay occurs when the WFA threshold value decreases with respect to increasing data generation rate per node. The reason for this increase is the increasing data traffic due to increasing retransmission attempts with shortening timeout duration.

- Success probability increases dramatically with decreasing WFA threshold value. The reason for this is that the number of received packets increases when the timeout duration decreases. Otherwise, newly generated packets will lead to the cancellation of retransmission attempts of the timed out packets if the WFA threshold value increases, decreasing the number of received packets.

- Receive throughput also increases dramatically with decreasing WFA threshold value due to increased number of received packets.

- There is no considerable difference between the simulation results of the cases when the WFA threshold value is 20s and 40s in terms of average delay, success probability and receive throughput. The reason for this is that the probability of the cancellation of retransmission attempts due to newly generated packets increases if the WFA threshold value increases and increasing further that value beyond a large value has almost no further effect on the cancellation of retransmission attempts. In other words, newly generated packets prevents most of the retransmission attempts limiting the number of received packets in the cases of large WFA threshold value.

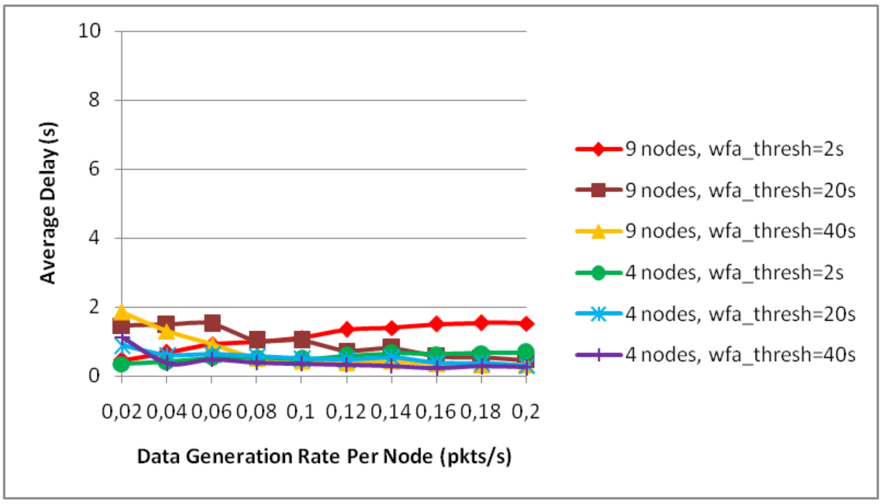

Figure 4.18: X-PACCA Average Delay

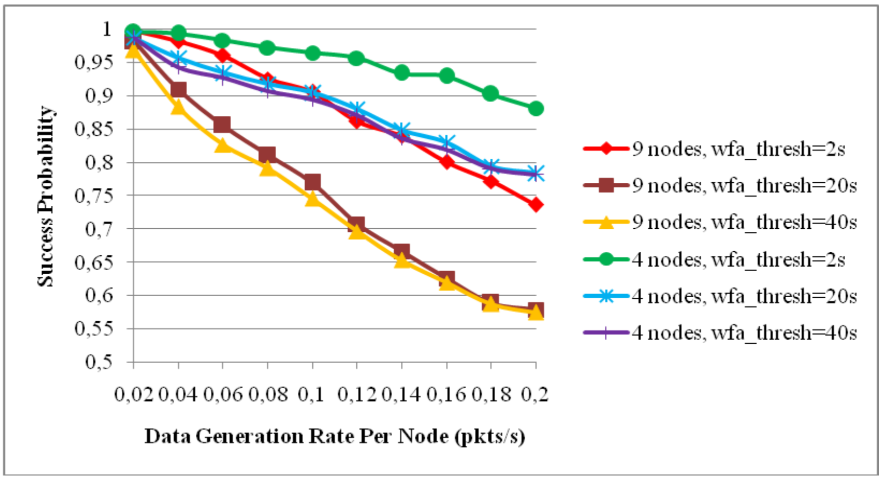

Figure 4.19: X-PACCA Success Probability

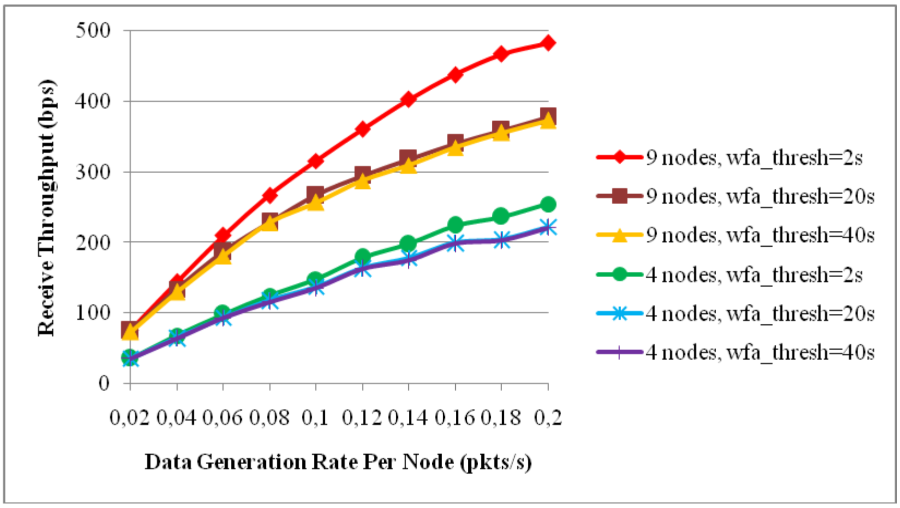

Figure 4.20: X-PACCA Receive Throughput

4.5 Evaluation of X-PACCA Performance in Spherical Deployment

The performance of X-PACCA protocol in spherical deployment is investigated as well as spherical cone deployment. In this study, the sink node is assumed to be in the center of a sphere submerged in water. The other nodes are randomly deployed in a sphere with R_{max} value. Figure 4.21 shows an example deployment obtained from the MATLAB simulator.

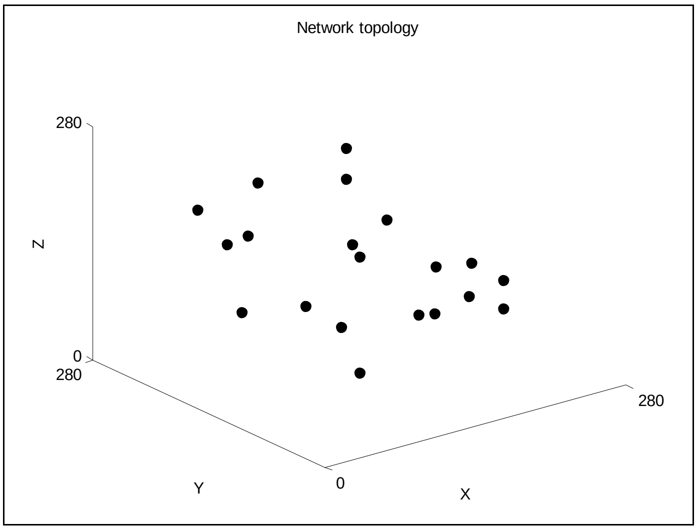

Figure 4.21: Randomly deployed nodes in a sphere (19 source nodes, 1 sink node)

The chosen parameters for this study are as in Table 4.5.

Table 4.5: Parameter values for spherical deployment

Parameter	Value
P_{req}(required power for a node)	0.2W
Backoff constant (k)	0.1
PER_{req}	$4*10^{-10}$
R_{max}	140m
Data Generation Rate Per Node	0.02 - 0.22 packets per seconds
Simulation time	3000s

Three scenarios are implemented, in which 4, 9 or 19 active nodes are randomly deployed in the same amount of volume and 1 sink node is deployed in the center of the sphere. The obtained results are as follows:

- Average delay for the scenario with 19 active nodes is slightly higher than the ones with 4 or 9 active nodes as can be seen from Figure 4.22.
- Success probability for 4 active nodes is significantly higher as in Figure 4.23.

- Receive throughput increases with increasing number of active nodes as in Figure 4.24. However, the structure of graph returns to the logarithmic type as the number of nodes increases while it is linear for smaller node numbers because the number of collisions at the sink node increases under heavy traffic.
- There is a significant decrease in success probability but slight increase in delay. The reason can be explained as follows. When the number of nodes increases in the same volume, the number of packets coming to the sink node at the same time increases so that the number of collisions at the sink node increases and the number of successfully received packets decreases. Delay is computed for successfully received packets so it does not change significantly. Furthermore, congestion avoidance mechanism of X-PACCA protocol prevents exponential increase in average delay.

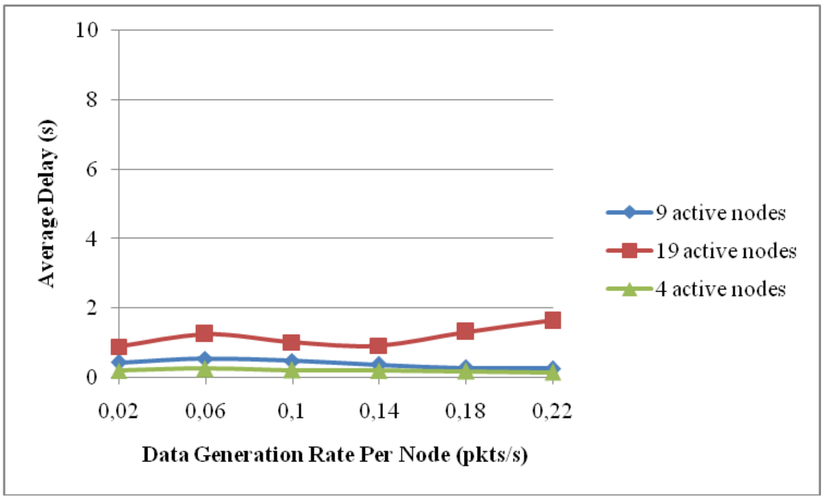

Figure 4.22: X-PACCA Average Delay in Spherical Deployment

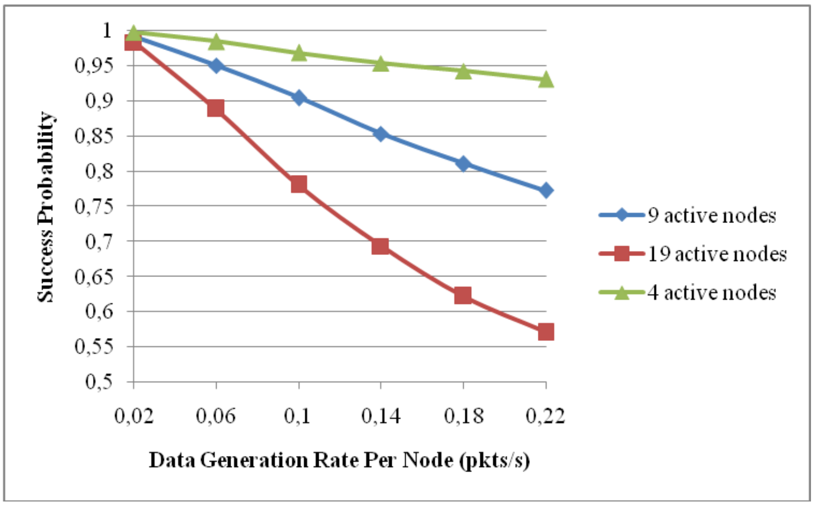

Figure 4.23: X-PACCA Success Probability in Spherical Deployment

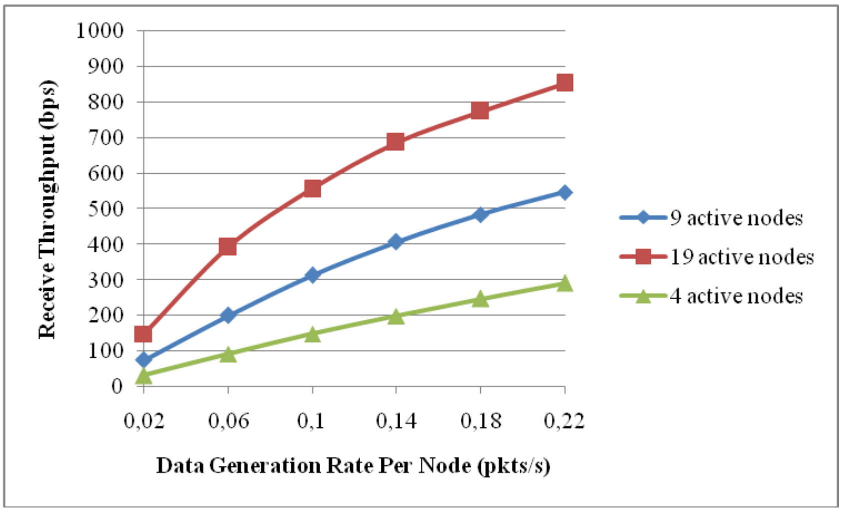

Figure 4.24: X-PACCA Receive Throughput in Spherical Deployment

4.6 Comparison of MAC Layer Performance

The final study regarding performance evaluation of X-PACCA protocol is the comparison of MAC layer performance with CSMA and DACAP protocols, which are known UASN MAC protocols. In this scope, 9 active nodes are randomly deployed in a spherical cone with the vertex angle of 30°. Single-hop transmission from source nodes to the sink node is guaranteed by adjusting R_{max} value to monitor only the performance of MAC layer. The chosen parameter values are as in Table 4.6.

Table 4.6: Parameter values for MAC performance comparison study

Parameter	Value
P_{req}(required power for a node)	1W
R_{max}	380m
Vertex angle	30°
Packet size	50Bytes
ACK size	24bits
RTS, CTS and WARNING packet size (DACAP)	24bits
Slot length (X-PACCA)	0.005s
Backoff constant (k) (X-PACCA)	0.1
Window size (X-PACCA)	3
MAC queue size	50
Retransmission attempts	5
PER_{req}	$4*10^{-10}$
Data Generation Rate Per Node	0.02 - 0.22 packets per seconds
Simulation time	3000s

The obtained results are as follows:

- As can be seen from Figure 4.25, X-PACCA provides low packet latency at all data rates while CSMA and DACAP performs increasing latency with increasing data rate. The basic reason for this is that X-PACCA cancels retransmission attempts of old packets when a new event is detected so it decreases the data traffic and so decreases the average delay. DACAP displays an excessive increase in average delay because it comprises an RTS/CTS handshaking mechanism and a warning mechanism by introducing additional delays.

- As can be seen from Figure 4.26, CSMA provides the highest receive throughput while DACAP degrades the performance at high data rates because packets are accumulated in the MAC buffer due to long delays and the ones exceeding MAC buffer limit are discarded so the number of received packets decreases. On the other hand, receive throughput of X-PACCA is lower due to cancellation of retransmission attempts when a new event is detected, which limits the offered load to the network and leads to a decrease in the number of received packets.

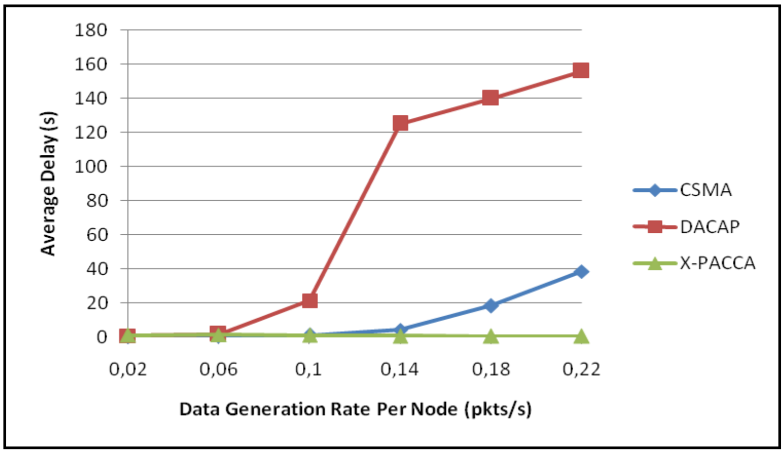

Figure 4.25: Comparison of MAC performance in terms of packet latency

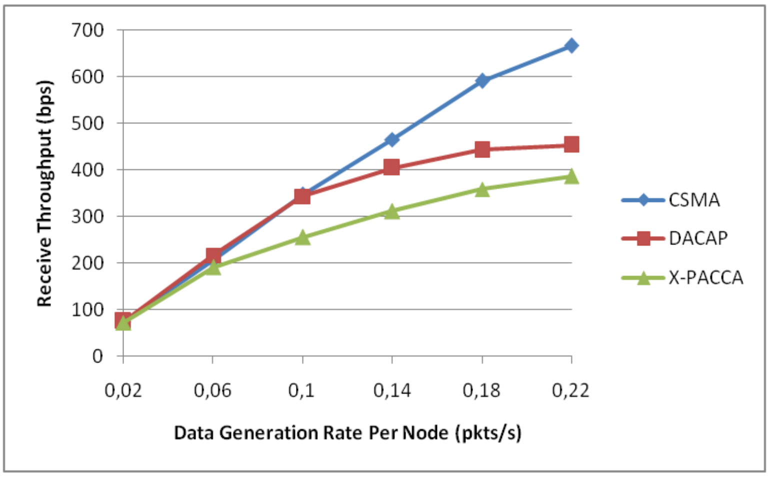

Figure 4.26: Comparison of MAC performance in terms of receive throughput

CHAPTER 5

CONCLUSION

In this thesis study, an event-based simulator for underwater channel environment has been developed using MATLAB. The main purpose was to be able to evaluate the performance characteristics of X-PACCA protocol. Such a simulator would also enable further evaluation studies of X-PACCA protocol.

Secondly, CSMA and DACAP which are popular MAC protocols for underwater sensor networks are implemented using this simulator. As a third step, some simulation studies have been realized to validate the simulator in terms of performance characteristics of these two protocols. Single hop scenario in [1] has been implemented to compare the latency and throughput values. The results are promising, however, they are not exactly the same. There can be several reasons for these differences. For example, channel models used in the present study and [8] are different so it is expected to see different results. Also, in [8] all parameters are not given so some assumptions had to be made for some parameters. Therefore, it is expected to see some differences.

Comparison studies with [4] have been carried out to observe possible differences on the implementation of X-PACCA protocol in a different simulator, namely Aqua-Sim. One of them was the one related to MAC performance evaluation in [4]. The receive throughput and latency results obtained in the present study and in [4] are almost the same. The other comparison study was the one related to the effect of source window size on protocol performance in [4]. The latency and success probability results of the same simulation scenario with [4] display some differences while they exhibit similar tendencies. As a later study, the reasons of these differences have been investigated by appling different PER_{req} values and choosing the size of MAC buffer as infinite as they are suspected issues. It is seen that when PER_{req} value is chosen as a smaller value and MAC buffer size is infinite, the results become more similar. Also, it is observed that choosing PER_{req} small increases the sensitivity of the network to interferences and decreases the relay traffic and the number of received packets. It is thought that the reason for further differences may be the channel model or interference model differences between the MATLAB simulator and Aqua-Sim.

As another study, the effect of WFA threshold parameter on the performance of X-PACCA protocol has been investigated. It is shown that choosing WFAthreshold value smaller leads to higher success probability while it causes very little increase in end-to-end delay. Thus, choosing WFA threshold value smaller is preferable.

Also, a simulation study for spherical deployment has been realized by employing different numbers of sensor nodes in the same volume. As expected, success probability decreases significantly with increasing number of nodes. However, average end-to-end delay increases slightly. It is important to show the usefulness of X-PACCA protocol in time-critical applications.

Finally, a comparison study of X-PACCA protocol with CSMA and DACAP protocols in terms of MAC layer performance has been achieved. The obtained results shows that X-PACCA outperforms CSMA and DACAP in terms of average end-to-end delay. On the other hand, receive throughput of X-PACCA is lower due to the cancellation of retransmission attempts when a newly generated packet comes. However, it is important to limit the number of packets in the underwater network because of the challenging structure of underwater channel.

As a limitation of the present study, the interference, propagation etc. models and some variable values used in ns2 Miracle [8] and AquaSim [4] simulators are not known, so the results of comparison studies can not be guaranteed. Also, all parameters used for protocol implementations are not given in the papers, so some assumptions had to be made.

As a future work, it would be beneficial to compare the developed simulator with a commonly used underwater network simulator such as AquaSim in terms of used models and variables. In the context of X-PACCA protocol performance, parameter effects in spherical deployment can be investigated. Furthermore, comparison studies with other MAC and routing layer protocols which claim to achieve high performance for UASN can be carried out.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257-279, May 2005.

[2] J. Heidemann, W. Ye, J. Wills, A. Syed, Y. Li, "Research Challenges and Applications for Underwater Sensor Networking," *WCNC Proceedings*, 2006.

[3] A. Bereketli and S. Bilgen, "Remotely Powered Underwater Acoustic Sensor Networks," *IEEE Sensors Journal*, vol. 12, no. 12, pp. 3467-3472, Dec. 2012.

[4] A. Bereketli, H. Türken, and S. Bilgen, "Cross Layer Power Adaptive CSMA/CA for Sink Powered Underwater Acoustic Sensor Networks," in *Proc. 12th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 26-33, Ajaccio, Corsica, France, June 2013.

[5] X. Wu, G. Chen, J. Chen, "Energy-efficient and Topology-aware Routing for Underwater Sensor Networks," in *Proc. 19th International Conf. on Computer Communications and Networks (ICCCN)*, pp. 1-6, 2010.

[6] M. Stojanovic, "Underwater Acoustic Communications: Design Considerations on the Physical Layer," in *Proc. 5th Annual Conf on WONS 2008*, pp. 1-10, January 2008.

[7] M. K. Park, V. Rodoplu, "UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 3, pp. 710-720, July 2007.

[8] C. Petrioli, R. Petroccia and J. Potter, "Performance evaluation of underwater MAC protocols: From simulation to at-sea testing," in *Proc. OCEANS 2011*, pp. 1-10, June 2011.

[9] K. Kredo II, P. Djukic and P. Mohapatra, "STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol," *IEEE INFOCOM 2009 Proceedings*, pp. 2961-2965, 2009.

[10] A. Cho, C. Yun, J. Park and Y. Lim, "Design of a block time bounded TDMA (BTB-TDMA) MAC protocol for UANets," *IEEE Journal of Oceanic Engineering*, vol. 38, pp. 2215-2226, October 2011.

[11] Z. Peng, Y. Zhu, Z. Zhou, Z. Guo and J.H. Cui, "COPE-MAC: A Contention-based Medium Access Control Protocol with Parallel Reservation for Underwater Acoustic Networks," in *Proc. OCEANS 2010*, pp. 1-10, May 2010.

[12] A. F. Harris III, M. Stojanovic and M. Zorzi, "Idle-time energy savings through wake-up modes in underwater acoustic networks," *Ad Hoc Networks*, vol. 7, pp. 770-777, 2009.

[13] B. Peleato, M. Stojanovic, "Distance Aware Collision Avoidance Protocol for Ad-Hoc Underwater Acoustic Sensor Networks," *IEEE Communications Letters*, vol. 11, no. 12, pp. 1025-1027, December 2007.

[14] F. Zorzi, M. Stojanovic and M. Zorzi, "On the Effects of Node Density and Duty Cycle on Energy Efficiency in Underwater Networks," in *Proc. OCEANS 2010*, pp. 1-6, May 2010.

[15] P. Casari, M. Zorzi, "Protocol design issues in underwater acoustic networks," *Computer Communications*, vol. 34, pp. 2013-2025, November 2011.

[16] S. Bilgen. Private note on simulation confidence. http://www.eee.metu.edu.tr/~bilgen.

[17] H. Türken, "Performance Evaluation of a New Cross Layer Protocol for Sink Powered Underwater Acoustic Sensor Networks," MS Thesis, Middle East Technical University, Electrical and Electronics Engineering Department, June 2013.

[18] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC protocol for underwater acoustic networks," in *Proc. OCEANS 2006 - Asia Pacific*, pp. 1-7, May 2006.

[19] W.-H. Liao and C.-C. Huang, "SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1686-1694, June 2012.

[20] Y. Noh, P. Wang, U. Lee, D. Torres, and M. Gerla, "DOTS: A propagation Delay-aware Opportunistic MAC protocol for underwater sensor networks," in *Proc. 18th IEEE International Conference on Network Protocols (ICNP)*, pp. 183-192, October 2010.

[21] L. Pescosolido, C. Petrioli, and L. Picari, "A Multi-band Noise-aware MAC Protocol for Underwater Acoustic Sensor Networks," in *Proc. IEEE 9th International Conference on WiMOB*, pp. 513-520, October 2013.

[22] G. Fan, H. Chen, L. Xie, and K. Wang, "A hybrid reservation-based MAC protocol for underwater acoustic sensor networks," *Ad Hoc Networks*, vol. 11, pp. 1178-1192, 2013.

[23] Z. Zhang and L. Blanco, "A Cluster Based Smart MAC Layer Protocol for Delay-Tolerant Underwater Wireless Sensor Networks," in *Proc. 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 308-315, October 2013.

[24] A. A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks," in *Proc. IEEE INFOCOM 2008*, pp. 231-235, April 2008.

[25] H.-H. Ng, W.-S. Soh, and M. Motani, "BiC-MAC: Bidirectional-Concurrent MAC protocol with packet bursting for underwater acoustic networks," in *Proc. IEEE OCEANS 2010*, pp. 1-7, Sep. 2010.

[26] Y. Chen, C. Lien, Y. Fang, and K. Shih, "TLPC: A Two Level Power Control MAC Protocol for Collision Avoidance in Underwater Acoustic Networks," in Proc. OCEANS 2013 MTS/IEEE, pp. 1-6, June 2013.

[27] T. Hu and Y. Fei, "QELAR: A Machine-learning-based Adaptive Routing Protocol for Energy-efficient and Lifetime-extended Underwater Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796-809, June 2010.

[28] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "VAPR: Void Aware Pressure Routing for Underwater Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 895-908, May 2013.

[29] S. Lee, and D. Kim, "Underwater Hybrid Routing Protocol for UWSNs," in *Proc. 5th International Conference on Ubiquitous and Future Networks*, pp. 472-475, July 2013.

[30] R. Coutinho, L. Vieira, and A. Loureiro, "DCR: Depth-Controlled Routing Protocol for Underwater Sensor Networks," ISCC, 2013.

[31] H. Yan, J. Shi, and J.-H. Cui, "DBR: Depth-Based Routing for Underwater Sensor Networks," in *Proc. Networking 2008*, vol. 4982, pp. 72-86, May 2008.

[32] P. Xie, J.-H. Cui, and L. Lao, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks," in *Proc. Networking 2006*, vol. 3976, pp. 3467-3472, May 2006.

[33] S. Yoon, A. Azad, H. Oh, and S. Kim, "AURP: An AUV-Aided Underwater Routing Protocol for Underwater Acoustic Sensor Networks,"*Sensors*, vol. 12, pp. 1827-1845, February 2012.

[34] M. Hill, P. Wang, Y. Noh, L. Vieira, M. Gerla, and J.-H. Cui, "Pressure Routing for Underwater Sensor Networks," in *Proc. IEEE INFOCOM 2010*, pp. 1-9, Mar. 2010.

[35] J. Llor, M. P. Malumbres, "Underwater Wireless Sensor Networks: How Do Acoustic Propagation Models Impact the Performance of Higher-Level Protocols?," *Sensors*, vol. 12, pp. 1312-1335, January 2012.

[36] A. Bereketli, "Remotely Powered Underwater Acoustic Sensor Networks," Ph.D. Thesis, Middle East Technical University, Electrical and Electronics Engineering Department, May 2013.

APPENDIX A

DETAILED DESCRIPTION OF THE MATLAB SIMULATOR

The simulator developed for underwater acoustic sensor networks using MATLAB is an event-based simulator. An event-based simulator is a kind of network simulator such that network events call the next events whose generation time is determined and by knowing the initiation time of each event the closest event is determined and run. The simulator consists of some .m files whose functions are described as follows:

- simulation.m: It is the main function of the simulator. In this file, some initializations are made firstly and parameter assignments are realized by calling parameter.m file. Network topology is generated within the call of topology.m file. Then, poisson traffic is generated and assigned to the sensor nodes. By these assignments a list of events is formed. Events are simulated until the simulation end time by calling run.m function. Afterwards, performance metrics are calculated according to the simulation results and related graphics are obtained for visualization of the results in simulation.m file.
- parameter.m: Network variable values and structures used in simulations such as queues and noise model are initialized in this file.
- topology.m: Localization of sensor nodes is realized in this file. Also, the 3D graph of deployment is drawn after the nodes are positioned.

- run.m: Events are queued and the event possesing the nearest initiation time is run. This function calls action.m function to run each event and determine the next events.
- action.m: In this file, a switch-case structure exists and each case represents a state of a packet in the network. In other words, a state machine composed of network states exists in this file. The network layer implementations are realized in this file. The main states indicating the status of a packet in network layers are recv_phy, send_phy, recv_mac, send_mac, recv_app, send_app, backoff etc. This file calls some other functions such as carrier_sense.m, prop_delay.m, new_id.m, tx_time.m and recv_phy.m.
- carrier_sense.m: The channel control which is in the scope of MAC protocols is realized with this function. It returns the information of the channel as it is idle or busy.
- prop_delay.m: The propagation delay between sensor nodes are calculated in this file.
- new_id.m: New identity numbers are generated in this file in order to be assigned to the newly generated packets.
- tx_time.m: Transmission time of packets are calculated in this file.
- recv_phy.m: The received power level and PER (packet error rate) are calculated in this file, and they are returned to the action.m file to determine the successful reception of a packet. This function calls recv_power.m function for determining received power levels.
- recv_power.m: The power level received by a node from other nodes is calculated in this file regarding trasmitter power level and the attenuation in underwater.

Function call graph for the simulator in terms of these .m files is as in Figure A. 1.

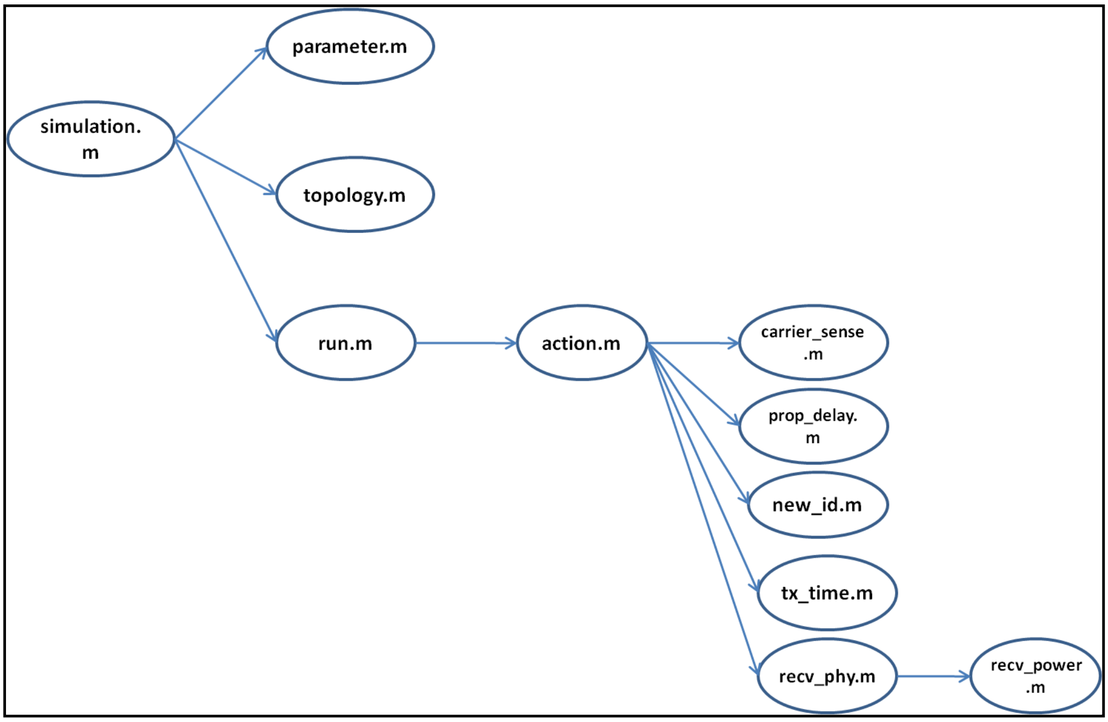

Figure A. 1: function call graph of the MATLAB simulator