PERFORMANCE EVALUATION OF A NEW CROSS LAYER PROTOCOL FOR SINK POWERED UNDERWATER ACOUSTIC SENSOR NETWORKS

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

BY

HASAN TÜRKEN

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

JUNE 2013

Approval of the thesis:

PERFORMANCE EVALUATION OF A NEW CROSS LAYER PROTOCOL FOR SINK POWERED UNDERWATER ACOUSTIC SENSOR NETWORKS

submitted by **HASAN TÜRKEN** 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 Eng. Dept.**

Examining Committee Members:

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

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

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

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

Assoc. Prof. Dr. Murat Erten
Innova Inc.

Date: 13.06.2013

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: HASAN TÜRKEN

Signature :

ABSTRACT

PERFORMANCE EVALUATION OF A NEW CROSS LAYER PROTOCOL FOR SINK POWERED UNDERWATER ACOUSTIC SENSOR NETWORKS

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

June 2013, 68 pages

The term Sink Powered Underwater Acoustic Sensor Networks (SPUASN) refers to a special configuration within the recent Remotely Powered Underwater Acoustic Sensor Networks (RPUASN) paradigm, where the data sink supplies energy to battery-free sensors that constitute the network. This thesis evaluates the performance of a new cross layer protocol, Cross Layer Power Adaptive CSMA/CA (X-PACCA), proposed for SPUASN. Simulation based performance evaluation is conducted both by investigating the effects of design parameters and by comparing with other protocols proposed for underwater sensor networks. Since X-PACCA is a cross layer protocol that integrates MAC, network and transport layer functionalities, comparisons are carried out separately by considering MAC and routing performance of the protocol in terms of end to end delay, packet delivery ratio and network throughput. This study also shows that, with appropriate selection of the protocol parameters, X-PACCA achieves low end-to-end delay, high packet delivery and high throughput performance.

Keywords: underwater acoustic sensor networks, wireless sensor networks, energy harvesting, protocol design, event-based network simulator

ÖZ

ALICI BESLEMELİ SUALTI AKUSTİK SENSÖR AĞLARI İÇİN TASARLANMIŞ YENİ BİR KATMANLAR ARASI PROTOKOLÜN PERFORMANS DEĞERLENDİRMESİ

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

Haziran 2013, 68 sayfa

Alıcı beslemeli sualtı akustik sensor ağları (SPUASN) terimi, yakın geçmişte önerilen uzaktan beslemeli sualtı akustik sensör ağları (RPUASN) paradigmasının özel bir konfigürasyonudur. Bu konfigürasyonda veri alıcısı, aynı zamanda ağı oluşturan pilsiz sensor düğümlerine enerji sağlar. Bu tezde, SPUASN için önerilen yeni bir çapraz katman protokol olan Katmanlar Arası Güç Uyarlamalı CSMA/CA (X-PACCA)'nın performansı değerlendirilmiştir. Protokolün benzetim tabanlı değerlendirmesi, hem tasarım parametrelerinin performans üzerindeki etkisi incelenerek hem de sualtı akustik ağlar için önerilen başka protokollerle performans karşılaştırılması yapılarak gerçekleştirilmiştir. X-PACCA, ortama erişim kontrolü, yönlendirme ve taşıma katmanı fonksiyonlarını entegre eden çapraz katman bir protokol olduğu için; karşılaştırmalar yapılırken ortama erişim kontrolü ve yol atama performansları birbirinden bağımsız olarak, uçtan uca gecikme, paket aktarım oranı ve veri aktarım kapasitesi açısından değerlendirilmiştir. Bu çalışmada ayrıca, protokol parametrelerinin uygun seçimi ile X-PACCA'nın düşük uçtan uca gecikme ve yüksek paket aktarımlarına ulaşabildiği gösterilmiştir.

Anahtar Kelimeler: sualtı akustik sensör ağları, kablosuz sensör ağları, enerji depolama, protokol tasarımı, olay tabanlı ağ simülatörü

To My Family...

ACKNOWLEDGMENTS

I would like to express my special thanks to my supervisor Prof.Dr. Semih Bilgen. My special thanks go to TÜBİTAK. I would like to thank my employer, ASELSAN. Finally, I would like to thank my wife, parents and friends who supported me throughout the whole time.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

As three fourths of the earth is covered with water, exploration and management of those areas are important. Underwater acoustic sensor networks (UASN) are envisioned to be used in many applications such as environmental monitoring, ocean sampling, pollution monitoring, disaster prevention, assisted navigation, military surveillance, etc. These networks have attracted more interest with the recent developments in technology related to both production and deployment of underwater sensor nodes [2] [3]. In the underwater environment, acoustic waves seem to provide the most promising communication medium when compared to radio-frequency and optical communication [3]; and currently it is possible to find commercially available acoustic modems to realize such communication [4].

Even though terrestrial wireless sensor networks constitute a mature technology and lots of work has been done in this area, due to different propagation and noise characteristics of the underwater environment most of these studies are inapplicable to underwater scenarios. The main problems hindering underwater communication are [2]:

- Propagation speed is much lower (1500 m/sec in water for acoustic waves).
- Bandwidth is much more limited as the operating frequencies are much smaller.
- Time varying channel responses and multipath phenomena.

These challenges will be explained in detail in Chapter 2.

Another issue in wireless communications and sensor networks is power limitation. Sensing, communication, data processing and for some applications moving the platform consume energy that is provided from the batteries which have limited energy. Hence it is also critical to optimize the energy consumption of each process to prolong the lifetime of each agent in the network. When compared to terrestrial sensor networks, energy consumption is more problematic in the underwater case due to challenging channel properties. For this reason extra processing and communication power is required for successful communication.

Two main strategies are employed to handle the energy problem in wireless networks: energy conservation and energy harvesting. Energy conservation methods provide techniques to optimize the consumed energy more efficiently. In energy harvesting schemes, the agent in the network tries to gather energy by means of a harvesting unit. For instance, in terrestrial sensor networks, solar power and vibration are two main sources for energy harvesting, as stated in [5]. In the underwater environment, energy harvesting from solar energy is infeasible since the attenuation of light waves in water is quite high. Vibration on the other hand can be considered as a source for energy harvesting.

A new paradigm, Remotely Powered UASN (RPUASN), has recently been proposed [6] to deal with the energy problem in UASN by increasing their lifetime indefinitely. RPUASN nodes harvest their energy from acoustic waves transmitted by an external acoustic source. These nodes include a harvesting unit, which consists of an array of hydrophones that converts energy in acoustic waves into electrical energy. Harvested energy is then stored in a storage capacitor in order to be used for sensing, processing and communicating.

Since energy is one of the main constraints in most of the proposed protocols for underwater sensor networks, removal of this constraint requires the design of new protocols. Cross Layer Power Adaptive CSMA/CA (X-PACCA) [7] protocol is suggested for Sink Powered Underwater Acoustic Sensor Networks (SPUASN), which is a special configuration of RPUASN. In SPUASN, the data sink is also the external supplier of acoustic energy.

X-PACCA is a cross-layer protocol which integrates MAC, network and transport layer functionalities. The basic idea of the protocol is that only the nodes having more harvested power than the previous node relay a packet and the size of the backoff window for CSMA/CA operation also depends on the power difference between nodes such that high powered nodes, that is, nodes closer to the sink, will have smaller backoffs.

In this thesis study, the performance of X-PACCA is evaluated by simulations by considering packet delivery ratio and end to end delay. Evaluations include both investigation of impact of design parameters and comparison of MAC and routing performance with some highly cited protocols.

This thesis is organized as follows:

Chapter 2 reviews previous work on underwater sensor networking and discusses the relevant challenges, mainly concentrating on the energy problem. Later, the idea of RPUASN is briefly described and finally some MAC and routing solutions proposed in the literature are overviewed.

In Chapter 3, the simulation infrastructure is introduced and the X-PACCA protocol is defined in detail.

Chapter 4 starts with the results of fidelity verification simulations, where the simulation infrastructure and metric calculation methods are verified by comparing results obtained in a paper selected from the published literature. The rest of this chapter presents the simulation scenarios and discusses their results.

Chapter 5 concludes the thesis with discussions on findings of the study. Several ideas for related future work are also suggested in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Underwater Acoustic Sensor Networks

Underwater acoustic sensor networks have become more and more attractive recently for scientific exploration, commercial exploration and military needs like coastline protection. Since such networking requires high technology for both production and deployment, technological developments boost this type of networking. Underwater sensor networks are used in applications like environmental monitoring, ocean sampling, pollution monitoring, military surveillance, disaster prevention and assisted navigation. Even topology of UASN depends on the application; basically, there are three types of topologies, namely, Static 2D, static 3D and 3D architecture with autonomous vehicles [2].

Although there is a lot of work done about terrestrial wireless sensor networks, different characteristics of the underwater environment make most of those works inapplicable. Major differences that lead to challenges for underwater communication can be mentioned as: lower propagation speed, much more limited bandwidth, long and varying end-to-end delays and multi-path phenomena [2]. Details of these challenges that are specific to underwater acoustic sensor networks will be considered in section 2.2.

There are three possible signals for wireless networking in underwater world, which are radio, optical and acoustic. Radio waves suffer from attenuation at high frequencies and communication at low frequencies requires large antennas which makes radio waves not applicable. Optical waves on the other hand are strictly related to environmental effects, especially to turbidity and thus applicable only for some special applications. Acoustic waves seem to be the most promising way of underwater communication since they enable communication from long distances and at moderate rates [8].

2.2 Challenges in UASN

Underwater world is much more complicated in terms of wireless networking when compared to terrestrial wireless networking. Some challenges come from terrestrial counterparts and some are specific to underwater. In this section, the challenges, which make underwater networking troublesome, will be detailed.

Complicated Nature of Acoustic Channel: Acoustic waves have typical characteristics when compared to other waves like optical or radio which are common in networking. Two unique characteristics are detailed in [8] as propagation velocity and absorption.Propagation velocity of acoustic waves is much slower when compared to other waves and typically around 1500 m/s. Another point related to propagation velocity which makes acoustic waves problematic is that it depends on factors like temperature, salinity and pressure. Absorption is a result of conversion of wave energy into heat energy. During acoustic wave propagation, wave loses energy as it hits the bottom, surface or objects along the way. It is similar to attenuation in electromagnetic waves and depends on the frequency.

Limited Bandwidth: Low propagation speed of sound wave and the absorption loss which increases with frequency as well as with distance limit the available bandwidth for communication [9]. As an example, the WHOI micro-modem, which is a commercially available modem for underwater acoustic communications, provides a data rate of 130 to 5380 bits per second in a range of 4 to 11 km [10].

Long and Variable Propagation Delays: Propagation delays are long due to low speed of acoustic wave and variable because the speed depends on many factors like temperature, salinity and pressure. In protocols designed for traditional networking, propagation delays are either negligible or too small. So this characteristic of UASN is a challenge that makes underwater networking difficult.

High Bit Error Rates and Shadow Zones: Drastic changes on the factors effecting speed of the sound results in high bit error rates and shadow zones where there is no connectivity [11]. This increases transmission failures, retransmissions and thus energy consumption.

Multipath: Multipath means that from one point to another, there are more than one path which results in different arrivals and inter-symbol interference. Dominant factors of multipath phenomena can be stated as reflections from bottom and surface of the see in shallow waters and variance in speed of sound in deep waters [8].

High Production Cost: Underwater nodes are complicated devices when compared to terrestrial ones. The complexity of acoustic transceivers and necessary protections for harsh underwater environment are factors increasing the production costs. Due to high production costs, dense network topologies are not possible in most cases which is also different from terrestrial sensor networks where deployment of lots of cheap sensor nodes is possible [2].

Mobility and Proneness to Failures: Underwater nodes cannot be simply assumed stationary due to underwater currents whereas this assumption is valid in terrestrial nodes most of time. Underwater nodes are prone to failures due to fouling and corrosion which is a result of harsh underwater environment [2].

Energy problem: Energy is one of the main problems in all wireless sensor networks. But in underwater case it is more serious. Energy problem in UASN is detailed in section 2.3.

2.3 Energy Problem in UASN

As already stated, energy is one of the most critical resources in wireless sensor networks. Energy is used for communication, sensing, processing and moving in some cases. Since the nodes in this type of networks have to work with their limited batteries; lifetime of a wireless sensor network depends on energy consumption.

When compared to terrestrial sensor networks, energy problem is more critical in underwater acoustic sensor networks. Due to the challenging properties of acoustic channel, processing power for successfully receiving and extracting data is higher. Another reason for the high criticality of the energy problem in UASN is that due to sparse deployment of the nodes, which is a result of high cost, communication distances are longer [2]. Deployment cost is also higher than the one with terrestrial networks which makes harder to replace out of energy nodes.

To cope with the energy problem in wireless sensor networks, there are mainly two approaches, energy harvesting and energy conservation. In energy harvesting, a node gathers energy from outside world with a harvesting unit. In terrestrial sensor networks, most common and promising sources are solar power and vibration [5]. Solar power harvesting is not applicable in underwater nodes due to too little or no sun light. In the case of vibration, acoustic waves can be a source of vibration. In [6], harvesting energy in UASN from an external acoustic source is proposed. Energy conservation schemes on the other hand, provide methods for using limited energy as efficiently as possible.

In [12], taxonomy for the methods of energy conservation in wireless sensor networks is given. The three main categories are data driven, mobility based and duty cycling approaches. Data driven approaches try to reduce energy spent for acquiring, receiving and processing data. Energy efficient sampling methods can reduce the energy needed for acquisition and some compression and prediction in data can reduce energy spent for transmitting, receiving and processing data. Mobility based schemes suggest using mobile sinks or relays to collect data from nodes which reduces energy consumed for transmission. Rather than being all nodes mobile, which can result in higher energy consumption, some specific mobile nodes with no energy constraint are used to collect data. Duty cycling approaches suggest dynamically sleeping and awaking nodes to conserve energy.

Duty cycling can be applied in terms of topology control and/or power management. Topology control allows nodes to be active or idle according to their location or the connectivity of the network. Topology control can be applied to dense networks in general. But as stated before, UASN are deployed sparsely due to the cost. Power management on the other hand, uses sleep/wakeup protocols or medium access control (MAC) protocols with low duty cycles. Sleep wakeup protocols can be categorized into three: On-demand, scheduled rendezvous and asynchronous. On demand means, nodes sleep normally and when there is something to send, they are awakened by means of control messages. This method seems promising but requires a second low power radio for control purposes, which increases cost. In scheduled rendezvous, nodes have active and idle periods, but this method requires synchronization. Asynchronous protocols do not need synchronization and have their local idle and active periods. Three categories in low duty cycle MAC protocols are TDMA, contention based and hybrid. TDMA protocols inherently apply duty cycling since nodes turn their radio on in only their own slots. In contention based protocols, duty cycling can be achieved by turning on transceiver of a node when it grabs the channel as a sender or when it is the intended receiver.

2.4 The Idea of RPUASN

RPUASN paradigm is proposed to deal with the energy problem in UASN by increasing their lifetime indefinitely [6]. RPUASN nodes harvest their energy from an external acoustic source. A typical RPUASN node has four main hardware units, namely, sensing and processing (CSP) unit, communication unit, power unit and harvesting unit. CSP and communication units are also available in traditional underwater sensor nodes. On the other hand, harvesting and power units are special units for harvesting and storing power from external source. Harvesting unit consists of several harvesting hydrophones which convert kinetic energy of acoustic vibrations created by external acoustic source into electrical energy. Voltage induced by harvesting unit is stored in power unit with the help of a DC converter and a storage capacitor. In [6] it is shown that, it is practically possible to operate such a node by using commercially available components.

Performance of RPUASN has been investigated [6] in terms of source power and node to source distance by considering sensing coverage and network connectivity. Two different topologies, which differ by placement of external acoustic source, have been presented. First topology is formed by placing an omnidirectional acoustic source in the middle of a spherically shaped topology and deploying nodes around it. Second topology has a spherical cone shape as a result of positioning a circular piston type projector on water surface.

Since feeding underwater sensor nodes with an acoustic power source is a completely new concept, a protocol stack for this network requires further study. Cross Layer Power Adaptive CSMA/CA (X-PACCA) protocol [7] has been suggested for a special configuration of RPUASN and briefly described in section 2.4.1.

2.4.1 Cross Layer Power Adaptive CSMA/CA

Sink Powered Underwater Acoustic Sensor Networks (SPUASN) is a special configuration of RPUASN, where the data sink is also the external energy supplier. X-PACCA is proposed for SPUASN, as a result of the need of new communication protocols of such networks. X-PACCA is a cross-layer protocol which integrates MAC, network and transport layer functionalities. The key idea of the protocol is relaying mechanism which depends on harvested power at node. Only nodes that have more harvested power than the previous node will relay a packet and backoff window size calculation method will give more chance to a node which has more harvested power. Since in SPUASN, nodes that are closer to the sink will harvest more power, the idea also means that, only the nodes that are closer than previous node will forward the packet and closer ones between forwarding nodes will have more priority. This will result in a loop-free path and hence lead to low end-to-end delays. End-to-end reliability is ensured by acknowledgement packets. Congestion is avoided by preventing redundant packet transmission. For this purpose, nodes cancel packet transmission if they hear a successful transmission of that packet by another node or receive ACK for successful reception of the packet by the sink. X-PACCA does not require global network information or clock synchronization at nodes; protocol operation proceeds via local decisions of individual nodes based on harvested power levels.

More details on X-PACCA and its protocol description are given in Section 3.3.3.

2.5 Overview of MAC Protocols for UASN

MAC protocols are responsible to provide effective usage of a single shared medium of multiple nodes. MAC protocols mainly aim to prevent data packet collisions caused by simultaneous access to shared medium and also consider fairness among nodes, low access delays and energy efficiency. Terrestrial wireless MAC protocols cannot directly be applied in UASN due to different channel characteristics of underwater acoustic channel. When designing a MAC protocol for UASN case, peculiar characteristics like narrow channel bandwidth, vulnerability to fading and multipath, dependence on network-wide clock synchronization, handling long propagation delays, optimizing energy consumption, difficulty of power control at each node to avoid the near-far problem, and scalability with number of nodes should be taken into account [13]. A number of solutions have been proposed for the medium access problem in conventional underwater acoustic sensor networks [14].

In general MAC protocols can be divided into two categories: contention free and contention based protocols. Contention free protocols separate node access in time, frequency or code domains. On the other hand, in contention based protocols channel is not separated in any domain, rather nodes access channel randomly or sense carrier to avoid collisions.

Contention free MAC protocols:

Time Division Multiple Access (TDMA) protocols require time synchronization between nodes which is a very challenging issue in UASN due to high and varying propagation delays. Usage of guard times overcomes this requirement to some extent but decreases network throughput. Thus TDMA is not a promising method for UASN if nodes are not close to each other. UWAN-MAC [15] is based on TDMA but does not require a centralized clock rather uses distributed synchronization. Nodes synchronize with neighbor nodes with the help of SYNC packets and adjust their sleep and wake-up schedules accordingly. Duty cycling helps for energy efficiency by only keeping awake nodes when they are either receiver or transmitter. Frequency Division Multiple Access (FDMA) methods are not appropriate to UASN due to its limited bandwidth due to frequency dependent fading and multipath. In Code Division Multiple Access (CDMA) a signal coexists both in time and frequency domain but it is separated at receiver by the help of signal processing techniques. Even CDMA seems one of the most promising solution for underwater case due its robustness to fading of acoustic channel [2], it will suffer from narrow bandwidth of acoustic channel [3]. UW-MAC [16] applies CDMA techniques and does not require slot synchronization.

Contention based MAC protocols:

Nodes contend to capture channel in this type of protocols according to some rules prescribed. ALOHA is the simplest form of this type of protocols where nodes simply transmit packets whenever it is needed. On the other hand, in Carrier Sense Multiple Access (CSMA) nodes listen the channel before transmission and transmit only if it is idle. Collision Avoidance (CA) method also can be applied in order to avoid transmission on an already occupied channel. One design alternative is handshaking or reservation before transmission which can also be applied to underwater case but these protocols generally suffer from large overheads of control packets due to small bit rates. R-MAC [17] is a reservation based MAC protocol for underwater acoustic sensor networks. It is mainly designed for long term aquatic monitoring applications. R-MAC aims to reduce energy consumption by reducing collisions and also to support fairness. In R-MAC nodes estimate propagation delays for scheduling transmissions. T-Lohi [18] is an innovative reservation based MAC protocol that applies CSMA. Nodes have a separate low power transceiver other than data transceiver, to send and receive tones. This tones are used for reserving channel for data transmission instead of conventional RTS/CTS mechanism and usage of tone enabled wakeup receivers allows reducing energy consumption. In [19], an improved RTS/CTS handshaking solution that does not need clock synchronization is proposed. RTS/CTS handshaking brings extra delay to MAC operation. To solve any uncertainty, coordination of medium access is carried out by a centralized controller in [20]; however, deciding on a network-wide collision-free transmission order requires knowledge of relative locations of all nodes. It is also proposed to choose a transmission order at the receiver side [21]. The receiver has to wait until it receives an RTS from all possible contenders, and this decreases channel utilization seriously. Slotted FAMA [22] combines TDMA, carrier sensing with a packet exchange between sender and receiver before transmission. Nodes rely on global time synchronization, and a node is allowed to transmit only at the beginning of a time slot.In [23], receivers periodically start packet transfers, generating continuous packet exchange overhead in the network. Furthermore, for a dynamic topology under channel fluctuations, it is not very practical to determine when to initiate transmission at each node. In [24], Cooperative Underwater Multi-Channel MAC (CUMAC) is proposed to solve the three hidden terminal problems, multichannel, long delay and traditional hidden terminal problems. The protocol depends on RTS/CTS scheme and deploys a cooperative collision detection method with the use of tone sequences.

In this thesis study, Slotted FAMA and UWAN-MAC protocols have been selected and used for comparing the MAC performance of the proposed X-PACCA protocol. The main reason for this selection is that these two protocols are among the most widely referenced underwater MAC protocols and they have already been implemented in Aqua-Sim 1.0 [25] which allows simulating and comparing with different scenarios/topologies. These two protocols are briefly described in the following subsections.

2.5.1 UWAN-MAC

UWAN-MAC [15] is a TDMA based MAC protocol for underwater acoustic sensor networks. Different than pure TDMA protocols, an adaptive method is suggested. It considers underwater ecological sensing and monitoring applications with high node densities. Energy efficiency with low duty cycle is aimed when designing the protocol. The protocol consists of two phases, initialization and data transmission. In the initialization period, nodes randomly select a transmission cycle period and broadcast it to their neighbors in SYNC packets. After receiving a SYNC packet, a node extracts transmission cycle period for that node and stores it. The value of this period is a relative wakeup time. This enables nodes to know exact wake up times of their neighbors without knowing propagation delays. This approach assumes that the propagation delays do not change and clock drift is negligible from one cycle to another. After the initialization, each node wakes up according to the transmission times of its neighbors to receive data. Sender node do not sleep immediately after sending data, it waits for a “listen” time to handle newcomers to the network. Newcomers send “HELLO” packets in these listen times to join the network. To cope with node and synchronization failures, there are reserved fields in data header namely, missing list and SYNC. Missing list enables to inform a node that there is a problem with its transmission. This can be a collision or a synchronization failure. Informed node can try to resynchronize as a newcomer to overcome a synchronization failure. SYNC field on the other hand, keeps the transmission cycle period of the sender and let it change it when it is necessary. UWAN-MAC is an adaptive algorithm for new coming and leaving nodes (out of energy or failed nodes). Channel variations are also considered with the idea of “guard times” by using maximum propagation delay.

2.5.2 Slotted FAMA

Slotted FAMA [22] is an improved version of Floor Acquisition Multiple Access (FAMA) [26] method for underwater sensor networking. FAMA suggests both carrier sensing and RTS/CTS handshaking mechanism. In FAMA, to avoid collisions, RTS length must be greater than maximum propagation delay and CTS length must be greater than RTS length plus two times maximum propagation delay. Due to long propagation delays in underwater acoustic channel, satisfying these conditions will lead to serious loss of power which is not acceptable in UASN. Slotted FAMA introduces slotting in order to overcome this problem. Nodes will only send a packet in the beginning of a slot where slot length must be bigger than maximum propagation delay plus transmission time of CTS packet. With this restriction collisions can be avoided without applying the two restrictions of FAMA. In Slotted FAMA, when a node has something to transmit, it first senses channel in the beginning of next slot, if the channel is idle, then it sends an RTS packet. After sending the RTS, transmitter node waits for two slots, if no CTS is received then this means a collision. Receiver node sends a CTS packet just after. Transmitter sends DATA packet and receiver node sends an ACK packet after successful reception. By using a CRC field, high bit error rate scenario of acoustic channel is handled and a NACK packet is sent if an error occurs. Slotted FAMA also introduces transmission priority and trains of packets techniques to improve fairness and performance of the network respectively. Transmission priority aims to give more priority to nodes that have more packets to send by giving priority of using channel to nodes that has just received a packet. Trains of packets on the other hand, enable sending multiple packets with a single handshake. Even Slotted FAMA is better than FAMA for underwater case; its disadvantage is using time synchronization which is a challenging task in UASN. Slotted FAMA uses time guards to handle varying propagation delays which will lead to a decrease of protocol performance.

2.6 Overview of Routing Protocols for UASN

Sending a packet directly from a source to a destination is not always possible. Especially in large networks, nodes should cooperate to complete transmission successfully. Selection of nodes that will participate in this packet delivery process is the duty of routing protocols. In general, routing protocols can be divided into three categories, namely, proactive, reactive and geometric routing protocols. In [2], proactive protocols are classified as unsuitable for UASN, since they have a large signaling overhead during setup phase, which is more severe in mobile underwater case and there is no need for connecting all nodes to each other. Reactive protocols are also classified as unsuitable since they have a signaling overhead just before sending a packet which causes long delays when transmitting a packet and they rely on symmetrical links which may not be suitable for UASN. Geographical routing protocols seem to be promising but it is not clear yet, how a sensor knows its accurate location, since GPS is not applicable in deep water.

Vector Based Forwarding (VBF) [27], Focused Beam Routing (FBR) [28] and Directional Flooding-Based Routing (DFR) [29] are three examples for location based routing solutions for underwater networks. VBF is considered as the first geographic routing solution for UASN and in VBF, relay nodes are selected according to their distance to a vector between source and destination. Nodes are assumed to know their own location and the location of all source, destination and all relaying nodes are carried in packets. In [30], a hop by hop version of VBF is suggested, in which vectors are calculated at each hop rather than once from source to destination, in order to increase robustness of VBF. On the other hand, FBR is a cross-layer approach that primarily aims power efficiency. Different power levels for transmissions are used in order to route the packet with minimum power. Nodes only participate if they are in the transmitting cone with a predefined cone angle, which is drawn from source to destination. DFR considers dynamic conditions and high error probability in underwater acoustic channel and includes link quality while calculating the routes.

As mentioned earlier, knowing exact location in underwater is a challenging task, but depth can easily be obtained with the help of pressure sensors. Depth based routing (DBR) [31] and Void Aware Pressure Routing (VAPR) [32] exploit this information. In DBR, in case of a contention among multiple candidates for packet relaying, only nodes located closer to the water surface, forward packets. On the surface, one or more sinks gathers these packets. DBR has problems with sparse networks or sparse areas formed later due to node movements, such that it fails to send the packet to sink even there is a possible route by using a deeper relay node [33]. VAPR propose a solution for this problem by using recovery routes. Next node selection depends upon sequence number, hop count, and depth information exchange among nodes and sinks.

A clustering based routing is proposed in Distributed Underwater Clustering Scheme (DUCS) [34] where the whole network is divided into clusters. A distributed algorithm is used to form clusters and select cluster heads. All nodes can access to cluster heads in a single hop and packets transmitted to them first. Cluster heads then forwards the packets to other cluster heads until it reaches to sink.

In this thesis work, VBF and DBR protocols have been selected and used for comparing the routing performance of the proposed X-PACCA protocol. The main reasons for this selection were as follows:

They are highly cited as a routing solution for underwater acoustic sensor networks, can work with a single sink and have a similar underlying principle.

2.6.1 VBF

VBF [27] aims to solve node mobility issue in a scalable way for UASN by considering energy efficiency. It is the first routing protocol where neighbor nodes decide whether to relay a packet or not with the help of location information. In VBF, neighbor nodes decide on relaying a packet according to their distance to vector between source and destination. In other words, only the nodes that fall within the virtual pipe between source and destination participate in packet transmission. In VBF, all packets carry the source location, sink location and relay location. Also a RANGE field is defined to handle mobility and radius of the pipe stored in RADIUS field. There are two types of query packets to initiate routing namely, sink initiated query and source initiated query. In sink initiated queries, sink node sends an interest message either by defining data type or location. On the other hand, in source initiated queries, source node sends a data ready packet into the network. Note that, all interest and data ready packets are relayed by the nodes that fall within the virtual routing pipe. A self adaptation mechanism also defined in the protocol to handle the density changes within the network. This mechanism aims to select the most desirable nodes as relays based on a factor named desirableness factor, which is a measure for relaying capability of a node. Sensitivity about routing pipe radius threshold, poor performance on sparse networks and communication overhead due to its 3-way handshake nature can be mentioned as the main disadvantages of VBF according to [33].

2.6.2 DBR

DBR [31] does not require full localization, rather it requires depth information that can easily be obtained with the help of a depth sensor. In DBR, nodes forward an incoming packet only if it comes from a deeper node. With this idea, the packet can be forwarded to sinks on the water surface. After the packet is received by any of the sinks on the surface, they can easily forward it to each other with via their radio transceivers. When a source node senses an event and wants to transmit a package, it first measures its own depth and puts it in the packet. Neighbor nodes that receive this packet compare this depth value with their own depth and only forward it if previous node is in a deeper location. DBR uses two queues namely, priority queue and packet history buffer to overcome packet retransmissions for energy efficiency. DBR is mentioned as a greedy algorithm which fails in a sparse network and handling this case is left as a future work in the paper which defines the protocol. Even though DBR is expected to perform well in dense networks; there are also problems like collisions and increased energy consumptions due to redundant packet forwardings [33].

CHAPTER 3

SIMULATION WORK

3.1 Simulation Overview

The aim of the simulations conducted within the scope of the thesis is to evaluate the performance of X-PACCA both in terms of its design parameters and by comparing with other protocols that have been proposed for underwater sensor networks. Simulation work consists of two stages: fidelity verification and performance evaluation. The aim of fidelity verification is to verify the simulation and metric calculation infrastructures and methods by comparing with results obtained in a published article. On the other hand, performance evaluation simulations are the main work done in this thesis to evaluate the performance of X-PACCA. Delivery ratio, end-to-end delay and throughput are the three metrics used for evaluations.

The paper in [1] was selected as the published work of reference for the fidelity verification stage of simulation work. This paper simulates three underwater MAC protocols, namely RMAC, UWAN-MAC and Slotted FAMA, in terms of throughput and energy consumption with two different topologies. Since the paper includes the two protocols, Slotted FAMA and UWAN-MAC that were used in the performance evaluation stage, and also compares throughput values of those protocols on a well defined topology, it was selected as reference for the fidelity verification stage. One other reason for this selection was that the simulator used, Aqua-Sim [25], is the same one as that used in the present study.

In the second stage of the simulation work, performance of X-PACCA is evaluated in terms of design parameters and also in comparison to other protocols. As discussed in Chapter 2, UWAN-MAC and Slotted FAMA are used for comparing MAC performance; and DBR and VBF is used for comparing routing performance. Finally overall X-PACCA performance is also evaluated for different node densities. All simulations were performed using Aqua-Sim [25], which is a network simulator for underwater sensor networks. Aqua-Sim is implemented as a patch over NS-2 [35], one of the most commonly used network simulators.

Unless otherwise stated, all simulations were repeated until confident stopping condition was satisfied for a confidence level of 99% in a confidence interval of +/- 10%.

3.1.1 NS-2

NS-2 [35], Network Simulator version 2, is a discrete event simulator that is widely used by research community since it is flexible, powerful and open-source. NS-2 is written in two languages C++ and

Otcl. Since it is open-source, it allows modifying the existing protocols as well as adding completely new ones.

3.1.2 Aqua-Sim

Aqua-Sim [25] has been developed as a result of the need for a specialized network simulator for underwater sensor networks due to their different characteristics from terrestrial networks, like channel characteristics due to acoustic communication, attenuation models and need 3-dimensional deployment. Aqua-Sim is a complete package for underwater network simulations which allows 3D deployment of sensor nodes, effectively simulates signal attenuations and packet collisions.

Aqua-Sim is implemented as an extension of NS-2 such that it can be developed independently. Just like NS-2, it is also open for implementations of new protocols and even new physical channel models. The physical channel model implemented in Aqua-Sim and also used in all simulations is given later in this section.

The latest version of Aqua-Sim accessible over internet was “Aqua-Sim-1.0 Release” at the time when the simulations in the scope of this study were performed. This package can be downloaded from [36].

Physical Layer Model Used Aqua-Sim simulates the physical acoustic underwater channel with the attenuation model given in Equation 3.1. This model is the same as the one used in [6].

$$A(l, f) = l^k * (10^{\frac{\alpha(f)}{10}})^l \tag{3.1}$$

In Equation 3.1, $A(l, f)$ is average signal attenuation, l: distance in meters, $\alpha(f)$ represents absorption coefficient at frequency f in dB per meters and k is the spreading factor.

In Aqua-Sim’s physical layer model, packet reception criterion is as follows:

Packet is sent with power $TxPower$, signal power decreases with attenuation model given in Equation 3.1. A node at distance l receives this packet if the attenuated power of the packet is higher than $RxThresh$ of the receiver node.

3.2 Auxiliary Tools Used in Simulations

3.2.1 Aqua-3D

Aqua-3D [37] is a visualization tool to visualize 3D underwater networks. NAM which is the animator of NS-2 can only show 2 dimensional networks. Since topologies used in these simulations were 3 dimensional, Aqua-3D was installed and used for obtaining topology figures using trace files generated after running simulations.

3.2.2 DeployNodes

DeployNodes is a tool which was developed within the scope of this work, to deploy nodes randomly or in a hexagonal form (this topology has been implemented but has been considered unrealistic and has not been used in simulations) via simulation script quickly.

DeployNodes also calculates harvested power for RPUASN nodes and outputs their locations and harvested powers in a file which is later used by simulations scripts. It only outputs the nodes which are actively participating network, i.e. able to harvest the minimum required power to operate. Step by step calculation of harvested power of an RPUASN node is given below.

Step by Step Calculation of Harvested Power [6]

1. Calculate source level, SL_{dB}, of acoustic power source:

$$SL_{dB} = 170.8 + 10 \times \log_{10} P_{elec} + 10 \times \log_{10} \eta + DI \tag{3.2}$$

where P_{elec} is acoustic source power in Watts, η is electro-acoustic power conversion efficiency and DI is the directivity index which is related to the vertex angle of acoustic transmission θ (in degrees) as in Equation 3.3

$$DI = 20 \times \log_{10} \frac{60\pi}{\theta} \tag{3.3}$$

2. Calculate attenuation depending on the distance R (in meters) between acoustic source and the node (AL_{dB}):

$$AL_{dB} = 20 \times \log_{10} R + \alpha(f) \times R \tag{3.4}$$

where $\alpha(f)$ is the absorbtion coefficient.

3. Calculate received level (RL_{dB}) at the harvesting node with:

$$RL_{dB} = SL_{dB} - AL_{dB} \tag{3.5}$$

4. Finally, harvested power (P_{harv}) in Watts can be obtained with the following formula:

$$P_{harv} = 0.7 \times n \times \frac{10^{(RL_{dB}+RVS)/10}}{4 \times R_p} \tag{3.6}$$

where n is the number of harvesting hydrophones, RVS is the receiving voltage sensitivity in dB and R_p represents the hydrophone impedance in ohms.

Two different usages of DeployNodes are given below:

- $DeployNodes\ random < NofNodes > < Theta > < PowElec > < SinkDepthOffset > < Range > < MinHarvPower >$
- $DeployNodes\ hexagonal < NofNodes > < Theta > < PowElec > < Dz > < R > < SinkDepthOffset > < Range > < MinHarvPower >$

NofNodes: Number of nodes including sink.

Theta: Vertex angle of the acoustic transmission.

PowElec: Electrical power of acoustic source.

SinkDepthOffset: Depth of the sink. (Necessary for Aqua3d viewing of topology)

Range: Communication range of the nodes, used in routing. (This parameter became obsolete, since no routing required.)

MinHarvPower: Minimum harvested power required for a node to operate.

Dz: Depth difference between two consecutive layers. (Only in hexagonal deployment)

R: Distance between two adjacent nodes. (Only in hexagonal deployment)

3.2.3 CheckConfidency

CheckConfidency was also developed for this thesis work and is responsible for checking whether the confidence stopping rule is satisfied. It is called via simulation script with simulation output samples as input and returns whether that simulation is confident.

As previously stated, the simulations are repeated until it is determined with 99% probability that the reported average value is within ±10% of the value being simulated [38].

The usage of CheckConfidency is as follows:

- $CheckConfidency < SampleFile >$

SampleFile: Address for the file which include samples obtained in simulation.

3.3 Changes in Aqua-Sim

Aqua-Sim comes with some protocols implemented for different network layers. “Aqua-Sim-1.0 Release” includes the two MAC protocols used in this work, namely UWAN-MAC and Slotted FAMA. But some changes and corrections needed to be made before starting simulations, since some bugs and missing points were detected in the protocol implementations during fidelity verification stage.

To examine the performance of X-PACCA, it was implemented in Aqua-Sim as a cross-layer protocol. It was implemented at MAC level but by interacting only with application layer and physical layer, it behaves as a cross-layer protocol (Figure 3.1). The details of this implementation are given in section 3.3.3.

3.3.1 Changes for UWAN-MAC

The following bugs and missing points have been corrected to make UWAN-MAC work correctly. Details about the changes can be found in Appendix A.

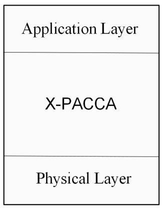


Figure 3.1: Cross-layer Behavior of X-PACCA

- Packet Header for *UWAN_SYNC* is registered.
- Packet size parameter is made configurable via simulation file, since different packet sizes are used at different simulations.
- Next hop is made configurable via simulation file. Since routing protocols are not taken into account; this work only compares the performance of MAC protocols. So in both stages of simulations, next hop for all nodes were set via simulation file, routing protocols do not have any effect.

3.3.2 Changes for Slotted FAMA

The following bugs and missing points are corrected to make Slotted FAMA work correctly. Details about the changes can be found in Appendix A.

- A time variable type, which was previously defined as integer, is corrected.
- Next hop is made configurable via simulation file. Since Routing protocols are not taken into account; this work only compares the performance of MAC protocols. So in both stages of simulations, next hop for all nodes were set via simulation file, routing protocols do not have any effect.
- “Trains of packets technique” is implemented. This technique is crucial for increasing throughput of Slotted FAMA. But in Aqua-Sim 1.0 the implementation of this technique was not implemented.

3.3.3 Implementation of X-PACCA

As previously stated, X-PACCA was implemented at MAC level in Aqua-Sim. But making necessary changes in network layer, it behaves as a cross-layer protocol between application and physical layers. In this section, first protocol definition of X-PACCA will be given and than other necessary changes will be mentioned. X-PACCA class definition is given in Appendix B.

3.3.3.1 X-PACCA Protocol Definition

X-PACCA is responsible for the following MAC, network, and transport layer functions:

- Organizing the access of SPUASN nodes to the shared medium.
- Routing of data packets along a path from the event region to the sink.
- Congestion avoidance through prevention of redundant packet forwarding.
- End-to-end reliability enhancement via ACKs sent by the sink.

Using the relations given in [6], a node knows its power budget harvested from the sink, P_{harv}. The transmission power level of the omnidirectional communication transducer of a node is a portion of the harvested power level, and it is denoted as $MYPL = \beta P_{harv}$ $(0 < \beta < 1)$. Each node has a unique identifier (ID), denoted by $MYID$.

Each node maintains the following queue structures:

- A MAC Transmit Queue (MTQ), holding packets to be transmitted and operating according to CSMA/CA with a unique starting window size for each node.
- A list (IGS) of packets to be ignored (already processed and/or relayed), used to store the IDs of packets ($PIDs$) that must not be re-processed or re-relayed. If PID is already a member of IGS, the list is not modified upon re-insertion.
- A list (WFA) of data packets waiting for ACK together with their detection times, maintained only at event data source nodes.

The format for X-PACCA data packets is shown in Fig. 3.2. A single bit $TYPE$ field is used to identify packets sent by the sink. SID represents the ID of the data source node that has sensed an event, and $EVID$ is the sequence number of the sensed event, which is repeated in a sufficiently long time for sensor network data pipeline capacity. These two fields form the packet ID, $PID = SID + EVID$. The transmission power level (TPL) is the power level with which the packet was transmitted, either from the original sensing node or when relayed. Event data necessary for the application level is carried in the $PAYLOAD$ bits, and CHK contains the checksum bits for error detection and correction.

TYPE	SID	EVID	TPL	PAYLOAD	CHK

Figure 3.2: X-PACCA data packet format.

The operation of X-PACCA upon event sensing is given in Algorithm 1. A node that senses an event constructs a packet with $PID = MYID + EVID$, $TPL = MYPL$, $TYPE = 0$, filling the corresponding $PAYLOAD$ and CHK fields. It inserts the packet into MTQ, and starts CSMA/CA transmission, applying the backoff procedure as given in Algorithm 3, with a fixed initial backoff window size W_s. Hence, regardless of where an event is sensed, the data source gets a fair chance for medium access. After successful transmission, the packet is deleted from MTQ, the node enters PID and local timestamp value into WFA, and PID is inserted into IGS, meaning that the node has processed the packet. Thus, if the node hears the same packet again from a neighbor, it simply discards the newly arrived copy.

Algorithm 1 X-PACCA pseudo-code for event sensing

```
procedure EventSensed($EVID$)
    $p$ = CreatePkt(0, $MYID$, $EVID$, $MYPL$, $payload$, $chk$)
    $MTQ$.insert($p$)
    Backoff($W_s$, $p$)                                ▷ CSMA/CA backoff, initial window size $W_s$
    $MTQ$.delete($p$)
    $WFA$.insert($p.PID$, $timestamp(EVID)$)           ▷ Wait for ACK
    $IGS$.insert($PID$)                                ▷ If you hear this event again, do not relay packets
    while $WFA$ is not empty do                        ▷ $WFA$ is checked for ACK from the Sink
        for all $PID \in WFA$ do
            if $myClock \geq WFA.timestamp(EVID) + WFA_{thresh}$ then
                                                       ▷ No ACK has arrived yet
                if SenseEvent($EVID$) then
                                                       ▷ The event is still active; re-transmit
                    $p$ = CreatePkt(0, $MYID$, $EVID$, $MYPL$, $payload$, $chk$)
                    $MTQ$.insert($p$)
                    Backoff($W_s$, $p$)
                    $MTQ$.delete($p$)
                    $WFA$.update($p.PID$, $timestamp(EVID)$)
                else                                   ▷ The event is not active anymore
                    $WFA$.delete($p.PID$, $timestamp(EVID)$)
                                                       ▷ Do not re-transmit
                end if
            end if
        end for
    end while
end procedure
```

Algorithm 2 X-PACCA pseudo-code for packet reception

```
procedure PacketReceived($p$)
    if $p.Type == 1$ then                              ▷ Sink packet
        if $WFA$.find($p.PID$) == $TRUE$ then          ▷ Check $WFA$
            $WFA$.delete($p.PID$, $time(p.EVID)$)      ▷ ACKed; do not wait
        end if
        if $MTQ$.find($p.PID$) == $TRUE$ then          ▷ Check $MTQ$
            $MTQ$.delete($p.PID$)                      ▷ Cancel retransmission
        end if
        if $IGS$.find($p.PID$) == $TRUE$ then          ▷ Check $IGS$
            $IGS$.scheduleDelete($p.PID$, $D$)
                                                       ▷ Delete $PID$ from $IGS$ at $currentTime + D$
        end if
    else                                               ▷ Packet from a relay node, $p.Type = 0$
        if $IGS$.find($p.PID$) == $FALSE$ then         ▷ I have not processed this pkt
            $IGS$.insert($p.PID$)
            if $p.TPL < MYPL$ then                     ▷ Closer to the sink; may relay $p$
                $p.TPL = MYPL$                         ▷ Update $p.TPL$ field
                $MTQ$.insert($p$)                      ▷ Packet waiting for transmission
                $W_{min} = k/(MYPL - p.TPL)$           ▷ Initial backoff window size
                Backoff($W_{min}$, $p$)
                $MTQ$.delete($p$)
            else                                       ▷ Packet relayed by a node closer to the sink
                if $MTQ$.find($p.PID$) == $TRUE$ then
                    $MTQ$.delete($p.PID$)              ▷ Cancel retransmission
                end if
            end if
        end if
        if $MYID == SinkID$ then                       ▷ I am the sink
            $p.Type = 1$                               ▷ ACK broadcast from the sink
            BroadcastPkt($p$)
        end if
    end if
end procedure
```

The node periodically checks the $(PID, timestamp)$ pairs in WFA, and those still unacknowledged after WFA_{thresh} are re-sent in case the corresponding event $EVID$ is still active.

The behavior of a node upon packet reception is presented in Algorithm 2. When a node receives a packet with $TYPE = 1$, that is an ACK coming from the sink, PID is searched and removed, if found, from WFA. The node also deletes PID from MTQ if found, and cancels pending CSMA/CA if not yet completed, because there is no need to relay a packet that has already been received successfully by the sink. Outdated packets are removed from the queues eventually to avoid indefinite growth of queue size; however, PID is not erased from IGS immediately. Instead, the node keeps PID in IGS for another duration of D, which is the propagation delay of the longest hop in the network. Hence, duplicate forwarding is avoided in case delayed packet arrivals occur from other neighbors.

When an SPUASN node receives a packet from any neighbor ($TYPE = 0$), the node compares its own transmission power level ($MYPL$) with that (TPL) of the sender of the packet. If $TPL < MYPL$, the packet is coming from a neighbor that is further from the sink, and the node may relay the packet. It ignores the packet if $PID \in IGS$, showing that it has processed and/or relayed the packet before, and this is a duplicated arrival. Otherwise, the node must relay the packet with $TPL = MYPL$ update, and it enters PID into IGS and MTQ. To minimize end-to-end delay, nodes closer to the sink have a higher priority to access the medium for relaying packets. Hence, at relay nodes, the initial backoff window size is adapted according to the power differential between the previous sender and this relay node, and determined as $W_{min} = k/(MYPL - TPL)$. Effects of the choice of the backoff constant k on performance are investigated in Chapter 4. Then, the node starts exponential backoff, as detailed in Algorithm 3. When CSMA/CA does succeed in transmission (i.e., no collision) after backoff, the nodes deletes PID from MTQ.

If $TPL \geq MYPL$ for the received packet, this means the node is hearing a packet relayed ahead of itself, from a node closer to the sink. The node inserts PID into IGS if $PID \notin IGS$. The packet has already been relayed by a node closer to the sink, so there is no need to relay a duplicate. The node cancels the pending backoff or transmission procedure for that packet and removes it from MTQ.

Whenever a packet is received at the sink, it is acknowledged by simply setting the $TYPE$ bit and broadcasting it, using W_s as the starting window size. That is, we are assuming that, as the sink can supply power to all sensors in a single hop, the sink can also be heard by all sensors in a single hop. Hence $TYPE = 1$ packets are not relayed, they simply cause sensors to drop acknowledged packets from their transmission lists.

Algorithm 3 X-PACCA backoff algorithm

```
procedure BACKOFF($W, p$)
    $i = 0$                                                        ▷ Backoff stage
    $txFlag = 0$                                                   ▷ Transmission flag
    $txAttempts = 0$                                               ▷ Number of transmission attempts
                                                                   ▷ $maxAttempts$ is the maximum number of attempts
    while $txFlag == 0$ && $txAttempts \leq maxAttempts$ do
        $txAttempts = txAttempts + 1$
        $W_i = 2^i W$                                              ▷ Window size, stage $i$
        $cnt = unif(0, W_i - 1)$                                   ▷ Counter
        while $cnt \neq 0$ do
            repeat
                if $MTQ$.find($p.PID$) == $FALSE$ then
                    EXIT
                                                                   ▷ Someone else has already transmitted the packet, so cancel transmission
                end if
            until channel idle for a slot time
            $cnt = cnt - 1$                                        ▷ Decrement counter
        end while
        TransmitPkt($p$)                                           ▷ Forward packet
        if $collision$ then
            if $i \neq m$ then                                     ▷ Maximum backoff stage check
                $i = i + 1$
            end if
        else
            $txFlag = 1$                                           ▷ Transmission successful
        end if
    end while
end procedure
```

3.3.3.2 Other Necessary Changes in Aqua-SIM for X-PACCA

Some other changes were required to add a new packet type, define RPUASN as a new node type, make X-PACCA cross-layer, use already implemented sink agent and use harvested power for packet transmission. Details about the changes can be found in Appendix A.

- X-PACCA packet header is registered.
- RPUASN node parameters defined in underwater sensor class. RPUASN node is a special type of underwater acoustic sensor nodes. So some variables were added to underwater sensor node class to handle this specialty.
- Skipped network layer for X-PACCA packets. Changes that were implemented simply means passing routing layer without any processing. This was required to make X-PACCA a cross-layer protocol even though it is implemented at the MAC level.
- Sink agent in Aqua-Sim is modified in order to handle X-PACCA packets.
- Harvested power is used in packet transmission. In Aqua-Sim, all nodes use the same power level for packet transmission, but RPUASN nodes use different power levels proportional to harvested power.

CHAPTER 4

SIMULATION BASED PERFORMANCE EVALUATION

In this chapter, simulation results are presented for both verification of the fidelity of the simulation infrastructure, and for performance evaluation of the X-PACCA protocol. Section 4.1 verifies simulation and metric calculation infrastructures and methods by comparing with the results published in the literature [1]. Section 4.2 includes subsections which evaluate the performance of X-PACCA in terms of design parameters (4.2.1), by comparing MAC performance with UWAN-MAC [15] and Slotted FAMA [22] (4.2.2), by comparing routing performance with DBR [31] and VBF [27] (4.2.3) and finally by investigating cross layer performance of the protocol (4.2.4).

4.1 Verification of the Simulation Infrastructure

Before starting performance evaluation and comparison simulations for X-PACCA, simulation infrastructure and metric calculation methods have been verified by selecting the paper in [1] as the published work and obtaining similar values with the ones given in the paper. UWAN-MAC and Slotted FAMA are simulated according to the scenario given in the paper. Simulation scenario can be described as follows:

- There are 2 sinks and 6 sources as shown in Figure 4.1, where the two sinks were in the middle and distance between any two adjacent nodes is 100 meters.
- Traffic is generated according to a Poisson process where the mean packet generation rate λ varies between 0.02 pkts/s and 0.2 pkts/s. Packet size used is 600 Bytes.
- Simulations were performed with the same settings as given in the paper and it is shown in Table 4.1.
- Values of parameters for UWAN-MAC and Slotted FAMA protocols were optimized to give maximum throughput values for the topology simulated since they are not given in the paper. The values used for these parameters are given in Table 4.2.
- As mentioned in the referenced paper, all simulations run for 3600 seconds, and each data point in the graph was obtained as an average of 10 runs. Confidence intervals for 95% confidence level also shown in the figure 4.2 for the results obtained within this work.

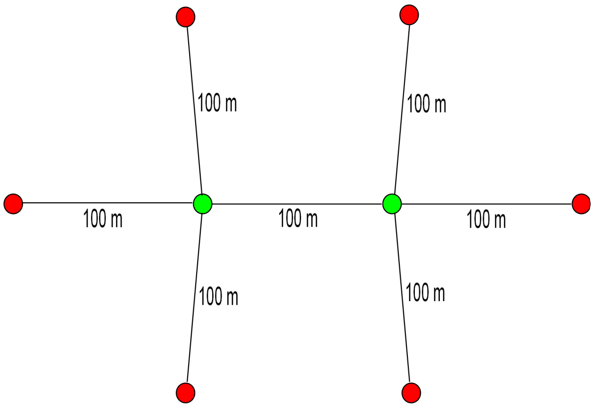


Figure 4.1: The topology used for the second scenario in the paper [1]

Table 4.1: Simulation Parameters given in Paper [1]

Topology area	500m*500m
Topology depth	500m
Transmission power	0.6 watt
Receive power	0.3 watt
Idle power	0.01 watt
Maximum transmission range	100 m
Routing protocol	VBF(vector based forwarding)
Bandwidth	10 Kbps
Frequency range	25 KHz

Table 4.2: Algorithm parameters used in simulations

Average cycle period (UWAN-MAC)	15
Standard deviation of cycle period (UWAN-MAC)	1
Guard time (Slotted FAMA)	0.0001
Maximum burst (Slotted FAMA)	15
Maximum number of backoff slots (Slotted FAMA)	35

The results are given in Figure 4.2. Results obtained in the paper are plotted with dashed lines. As can be seen from the figure, the results obtained in this work acceptably fit the results given in the paper. This is taken to demonstrate that the simulation environment and metric extraction methods used in this thesis work allow proceeding to the next stage of simulation work, where X-PACCA is simulated and evaluated by also comparing with these two protocols.

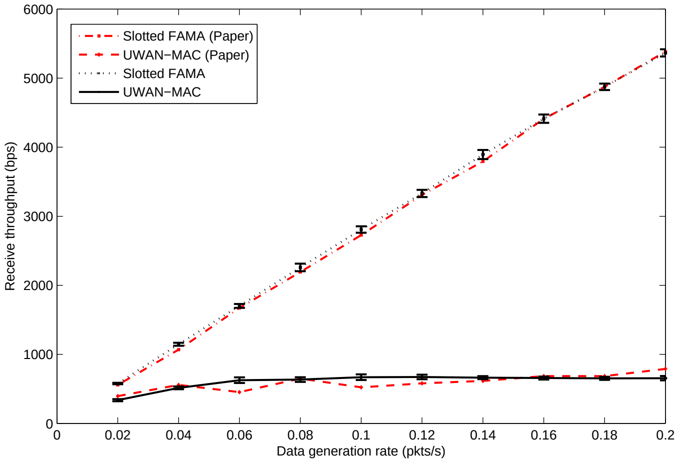


Figure 4.2: Result of fidelity verification simulations and comparison with the paper [1]

4.2 X-PACCA Simulations

In this section, the performance of X-PACCA is evaluated in terms of design parameters and compared with other protocols that have been proposed for underwater sensor networks. Namely, MAC performance of X-PACCA is compared with UWAN-MAC [15] and Slotted FAMA [22], and the routing performance is compared with DBR [31] and VBF [27].

Simulation Settings:

The following conditions hold for all simulations:

- The sink is the external acoustic power source placed at the water surface with the input power of $P_elec = 10$ kW, operating at the frequency $f = 10$ kHz with an electro-acoustic power conversion efficiency of $\eta = 0.5$.
- Communication frequency among nodes is set to 25 kHz at a bit rate of 10 kbps. Spherical spreading is assumed for acoustic signals.
- Unless otherwise stated, default values for X-PACCA and SPUASN node parameters are given in Table 4.3 and Table 4.4, respectively.
- All simulations except routing comparison simulations are repeated until confident stopping rule [38] is satisfied for a confidence level of 99% with a confidence interval of ±10%. In routing comparison simulations, each data point is a result of 20 simulations, which means an average of 1000 packets.

Table 4.3: X-PACCA Protocol Parameters

Parameter	**Default Value**
Data sink/source starting window size (W_s)	3
Maximum backoff stage (m)	5
Maximum number of retransmissions ($maxAttempts$)	5
Ack waiting time (WFA_{thresh})	20 s
Ignore timeout (D)	50 s
Aqua-Sim receive threshold	8.7×10^{-8} W

Table 4.4: SPUASN Node Parameters

Parameter	**Default Value**
Required power for a node to operate (P_{req})	0.5 W
Number of harvesting hydrophones (n)	5
Receiving voltage sensitivity (RVS)	-150 dB re V/uPa
Hydrophone impedance (R_p)	125 Ω
Harvesting efficiency	0.7

All parameters used in each simulation, including acoustic power source, RPUASN node, simulation environment and X-PACCA algorithm parameters, can be found in Appendix C. The receive threshold value 8.7x10-8 W is defined as follows: a packet sent with a transmit power of 1mW can be received from a node which is at most 100 m far away.

Figure 4.3 shows the topology for the default simulation parameters. As can be seen from the figure, a spherical cone shaped network is formed and nodes, which are randomly distributed in the medium and satisfying minimum harvested power requirement, are actively participating in network operation.

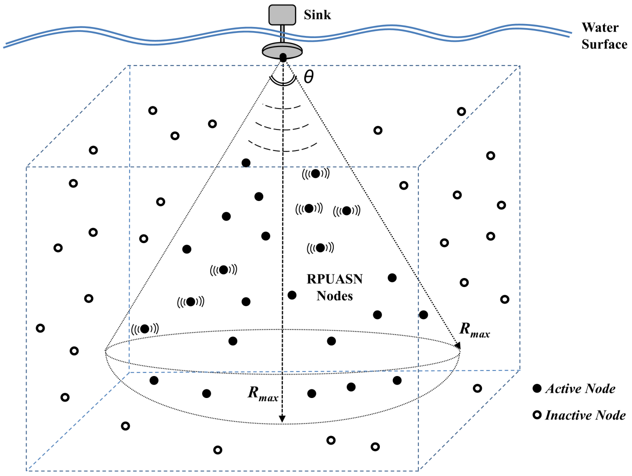


Figure 4.3: Spherical cone deployment for SPUASN

4.2.1 Impact of Design Parameters

In this set of simulations, the impacts of network and protocol parameters over the performance of X-PACCA are examined.

4.2.1.1 Impact of Backoff Constant (k)

Backoff constant (k), is the parameter that relates the initial transmission backoff window size to the harvested power at a relay node with the following equation:

$$W_{min} = \frac{k}{\beta * (MYPL - TPL)} \tag{4.1}$$

here β is the ratio of transmission power to the total harvested power, which is a characteristic of the node electronics, W_{min} is initial backoff window size, $MYPL$ is the transmission power used by the relaying node and TPL is the transmission power of previous sender of the packet.

Selection of k requires special attention because the determination of the initial window size with the help of harvested power is the basic idea of X-PACCA. Big k values will result in unduly elongated delays by increasing the initial backoff window size. On the other hand, small k values will hurt the performance by increasing collisions. There is no optimum W_{min} value for all network deployments, and the best selection depends on parameters like packet length/slot length, node density, power and vertex angle of transmission, average number of hops etc. While seeking for an optimum k parameter for a network, one should keep in mind that the optimum value of k depends on β. In other words, if β value changes somehow, that is, for nodes constructed with different electronic characteristics, k should be adjusted accordingly to preserve the optimum initial window size.

In the simulations carried out in this thesis work, β is changed to adjust the transmission range of nodes according to the simulation aim. For example, to analyze single hop behavior, a high β value is used to increase transmission ranges. Alternatively, to set an average transmission range, β is set accordingly. This is the reason for using different k values in the simulations.

Backoff constant, k, effects latency by changing the initial backoff window size. To minimize latency, window size, thus k, should be kept at a minimum such that the number of collisions will not exceed an acceptable amount. Effect of k on end to end delay can be divided into two components, namely, time lost due to collisions and delay while waiting to be transmitted due to backoff. It is clear that, as k decreases collision related latency dominates and as k increases time passed before transmission will be the main component of the packet delay.

There is no unique simulation to analyze the impact of k over performance of the protocol, rather, different k values are simulated in the impact of slot length, density and vertex angle simulations.

4.2.1.2 Impact of Slot Length

For a slotted MAC protocol, slot length is one of the most crucial parameters in terms of delay and throughput. Generally, handshaking based MAC protocols, like Slotted FAMA, have some restrictions on these parameters, which depend on control packet sizes and maximum propagation delays. Whereas, in X-PACCA, there is no such restriction; thus this parameter should be tuned according to network parameters, especially according to packet size. To see the effect of slot length on delay and packet delivery probability, the following scenario is simulated (all parameters can be found in Appendix C with simulation id = 1):

- The sink is at the water surface with the input power of P_{elec} = 10 kW and a vertex angle of 20°.
- A single source is placed at a farthermost point to the sink, which is 777 m away.
- There is an average of 60 randomly placed active nodes that participate in network operation.
- A single packet of 600 Bytes is transmitted by the unique source in each run of simulations.
- Nodes give up retransmission after the maximum backoff stage, $m = 5$.
- Effect of four different slot lengths (0.01 s, 0.1 s, 0.5 s and 1 s) on end to end delay and packet delivery probability are investigated for different backoff constant (k) values varying between 0 and 0.08.

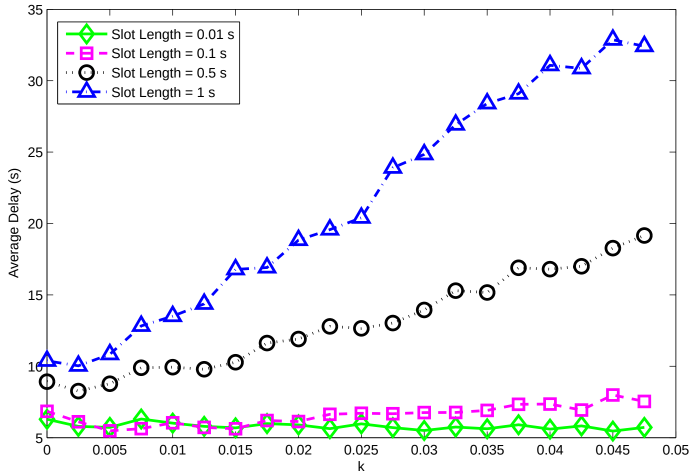
Results obtained with this simulation are shown in Figures 4.4a and 4.4b for average delay and success probability respectively. By also considering these results, the following observations can be made about the impact of slot length over network performance:

- Shorter slot lengths reduce the performance by increasing the probability of collision and longer slot lengths cause longer delays.
- Decreasing the slot length decreases observed delays until a point where collisions begin to dominate. It must be noted that, in this configuration since a single packet is simulated and nodes go on retransmissions until successful transmission, effect of collisions (only among relayed copies of the same packet) does not have a high impact on success probability.
- The effect of k is less significant when shorter slot lengths are used. To decrease the dependence on k, slot length should be kept at a minimum while setting up a network.
- There is a trade-off between end to end delay and success probability, which is also closely related to throughput, when selecting the slot length. So depending on the application, the best value for this parameter may change. For example, for this scenario, if the application aims at minimum delays, then this parameter can be set as 0.1 s with $k = 0.0075$.

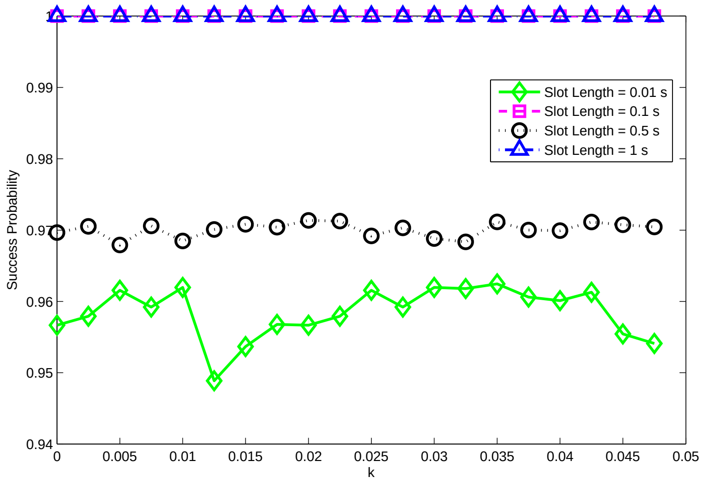
4.2.1.3 Impact of Node Density

In this set of simulations, X-PACCA protocol performance is investigated for different node densities. Effect of node density, again on end-to-end delay and delivery ratio, is simulated for different values of the backoff constant, k. Details of the network constructed are as follows (all parameters can be found in Appendix C with simulation ids = 2, 3 and 4):

- The sink is at the water surface with the input power of P_{elec} = 10 kW and a vertex angle of 20°, 30° and 45°.
- Data source node is placed at the farthermost point to the sink, which is 777 m away for 20°, 535 m for 30° and 365 m for 45°.



(a) Average delay



(b) Packet delivery ratio

Figure 4.4: Variation of X-PACCA performance with slot length and backoff constant

- Number of randomly placed active nodes varies between 31 and 297.
- A single packet of 600 Bytes is transmitted by the unique source in each run of the simulations.
- Nodes give up retransmission after the maximum backoff stage, $m = 5$.
- Slot length is set to 0.1.

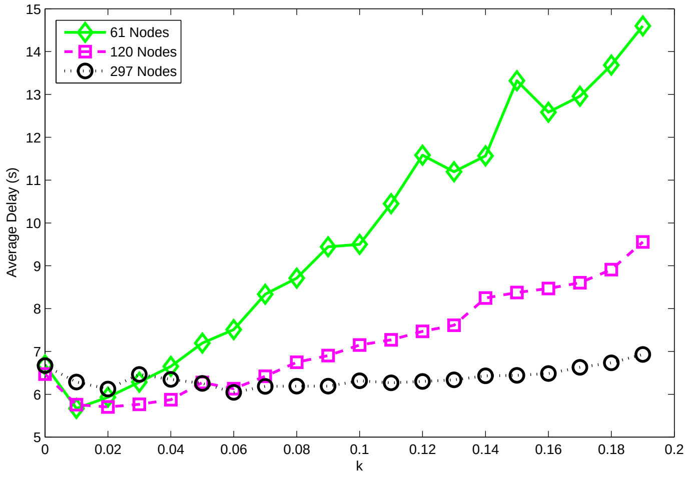
Results of this simulation set are plotted in Figures 4.5, 4.6 and 4.7 for vertex angles of 20°, 30° and 45° respectively. According to the results obtained:

- Higher sensor densities also result in higher packet delivery ratio as expected. However, packet delivery ratio is very close to 100% for all investigated node densities and angles.
- The selection of k is more significant for low node densities, but proper selection of this constant leads to lower latencies.
- k effects latency in two ways: (i) delay while retransmitting collided packets (ii) time spent during backoff. For sparse deployments, effect of collisions dominates before minimum delay point, and after that point, time spent due to increasing backoff windows is responsible for increasing delays.
- In denser networks, the selection of k is less critical since there are always some relay nodes that will send the packet within a random slot between 0 to W_{min}. It is expected that a higher backoff window is required in such topologies indicating a higher optimum value for k. This effect can be observed if delay performance of all three simulations above is considered.
- For smaller vertex angles ($\theta = 20°$), it is possible to find some smaller k values to minimize delays. Since there are not so many relay nodes available as with wider angles, smaller backoff windows are better thus smaller k values can be used. On the other hand, with bigger vertex angles ($\theta = 45°$), there will be more relay nodes at similar power levels for a packet sent from a far node which will lead to collisions. In Figure 4.7a, effect of collisions become more severe when k =0.02 to 0.04 especially with higher densities. Thus, some higher k values are required to increase backoff window lengths.

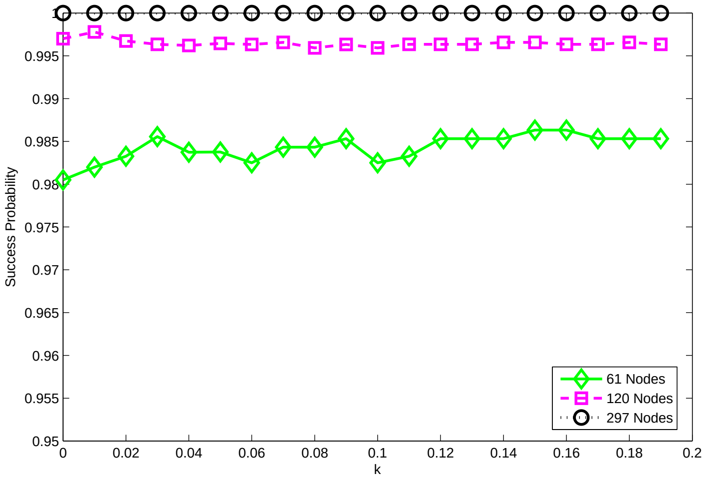
4.2.1.4 Impact of Vertex Angle

In this step, X-PACCA protocol performance is investigated for different vertex angles, the unique source is placed at a fixed distance for all angles. Delay and success probability are again the two metrics investigated. Constructed networks can be described as follows: (all parameters can be found in Appendix C with simulation id = 5):

- The sink is at the water surface with the input power of P_{elec} = 10 kW and a vertex angle of 20°, 30° and 45°.
- The single data source node is placed 550 m away from the sink for all angles.
- Number of randomly placed active nodes are 233, 177, and 127 for θ = 20°, 30° and 45° for the same deployment, respectively.

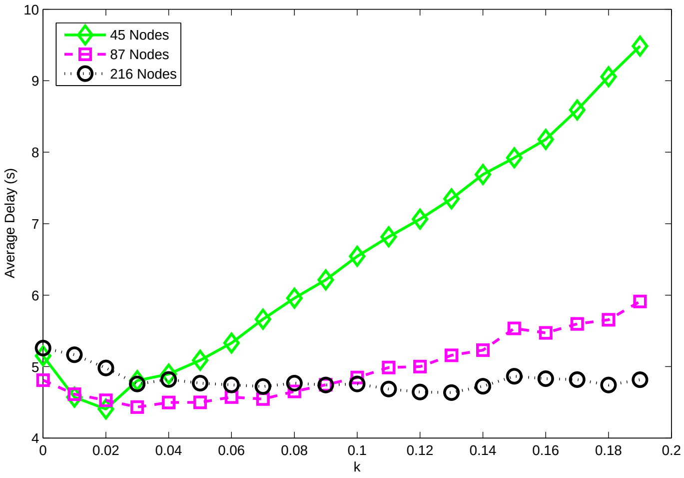


(a) Average delay

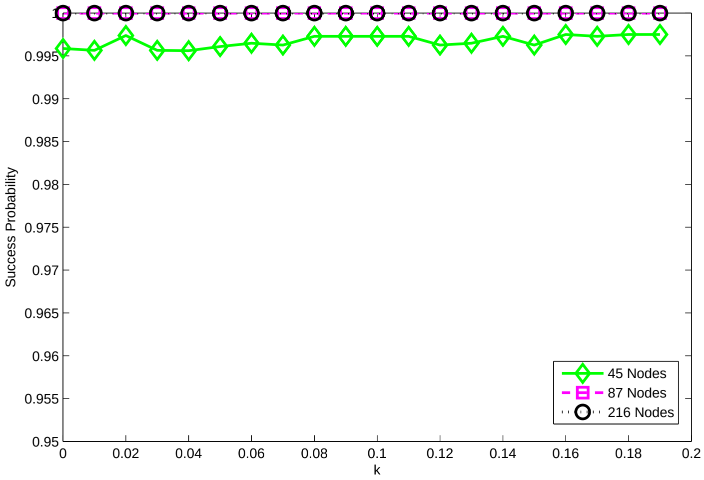


(b) Packet delivery ratio

Figure 4.5: Variation of X-PACCA performance with backoff constant and node population for vertex angle 20°

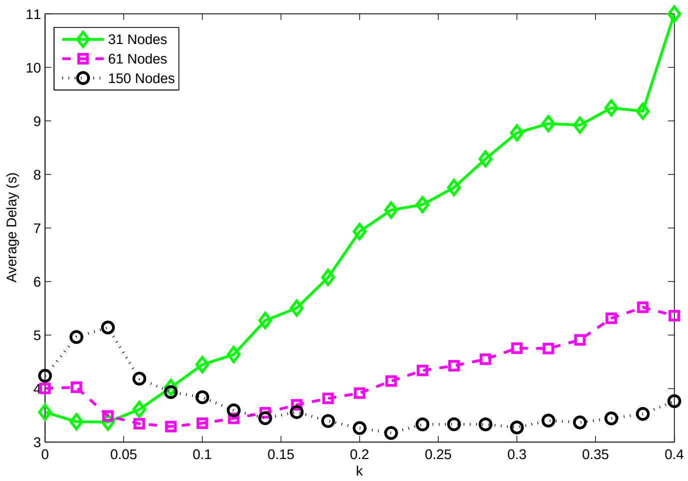


(a) Average delay

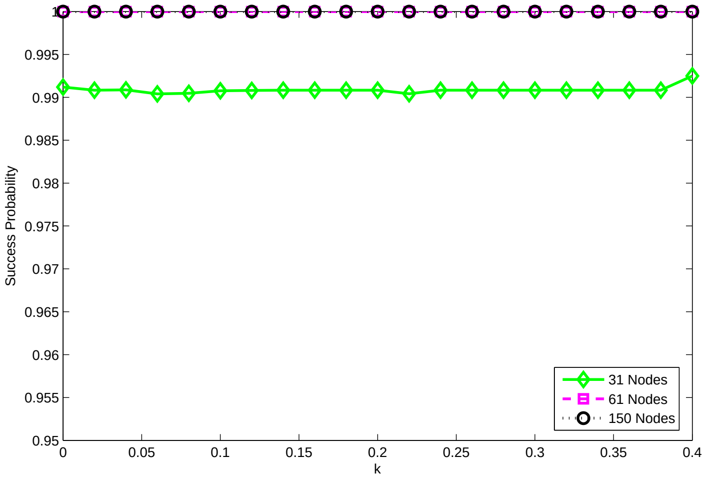


(b) Packet delivery ratio

Figure 4.6: Variation of X-PACCA performance with backoff constant and node population for vertex angle 30°



(a) Average delay



(b) Packet delivery ratio

Figure 4.7: Variation of X-PACCA performance with backoff constant and node population for vertex angle 45°

- A single packet of 600 B is transmitted by the source in each run of the simulations.
- Nodes give up retransmission upon reaching the maximum backoff stage, $m = 5$.

Assuming that the vertex angle of acoustic transmission can be adjusted in practice, when data from a specific node, or area of interest, performance can be optimized by adjusting this angle. As can be seen from Figures 4.8a and 4.8b:

- Success probability highly depends on vertex angle of transmission for the same deployment. With wider angles, number of relay candidates at same level is higher than narrower angles, probability of collision is higher. This results in lower success probabilities.
- For this deployment, if accessing more nodes at the same time is more important for the application, vertex angle of 450 can be used since more than 80% of all packets can be collected. With proper selection of k, around 0.002, similar delays as smaller angles can be obtained.
- Smaller angles will be required for accessing nodes at deeper locations if there are any.

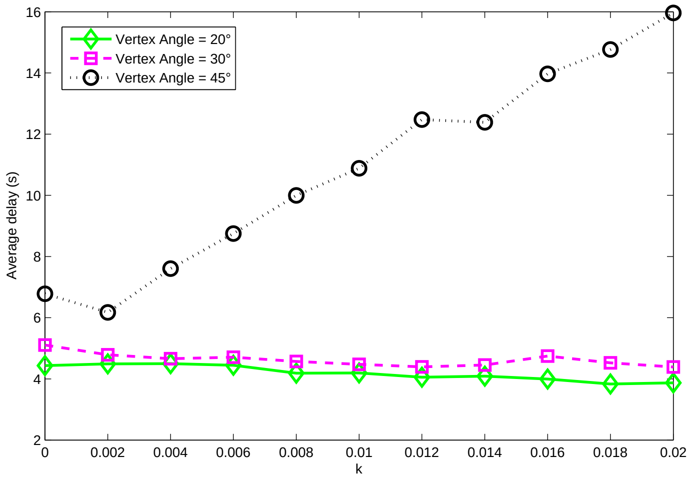
4.2.1.5 Effect of Source Window Size (W_s)

When an event is sensed by a source node, Ws is used as the starting window size for backoff. That is, classical CSMA/CA is applied at the source. The same starting backoff window size (W_s) is also used by the sink to send an acknowledgment for a received packet. To see the effects of W_s on end to end delay and also packet delivery ratio, the following scenario is constructed (all parameters can be found in Appendix C with simulation id = 6):

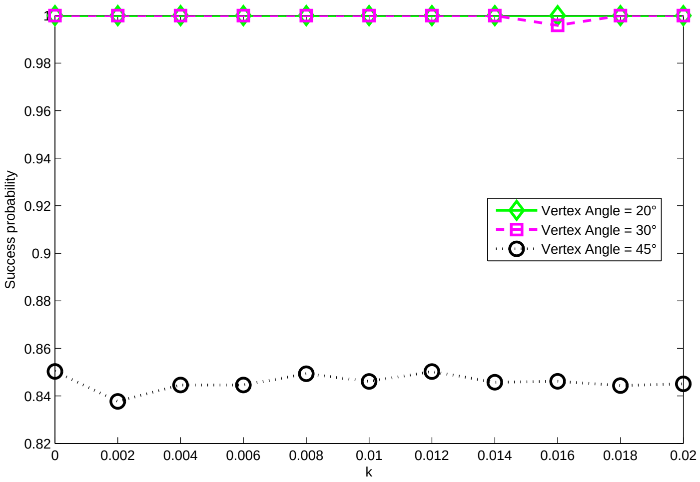
- The sink is at the water surface with the input power of $P_{elec} = 10$ kW and a vertex angle of 20°.
- All randomly placed sensor nodes are data sources as well as carrying out relay functionality and the number of active nodes are 5 and 10.
- All sources can access the sink in a single hop to ignore effects of routing. While relaying packets, nodes use a different window size related to the harvested power according to the expression 4.1. But in this simulation there is no relaying, but only the starting window size used for transmission at the data sources is being considered. For this purpose, k is set to 1 and β is set to 0.025.
- Each node generates 50 Byte packets according to a Poisson process with 0.02 to 0.22 pkts/s.
- Nodes give up retransmission upon reaching the maximum backoff stage, m = 5.
- Two different window sizes, 3 and 30, are simulated.
- Slot length is 0.005.

Figures 4.9a, 4.9b and 4.9c show latency and delivery ratio values for different source window sizes and different average number of source nodes. As can be seen from these figures:

- Small window sizes result in better performance at low traffic rates, whereas larger window sizes are needed with higher traffic rates.

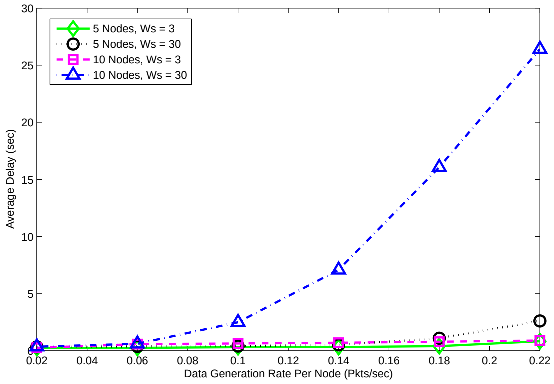


(a) Average delay

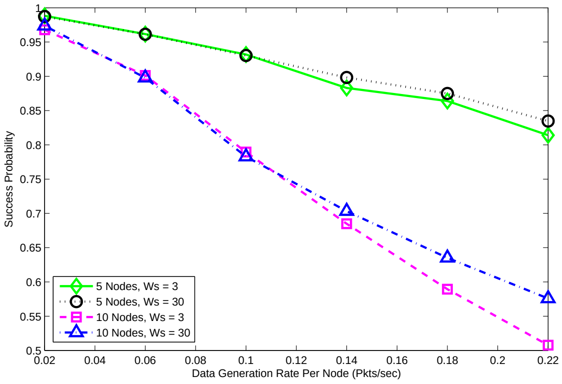


(b) Packet delivery ratio

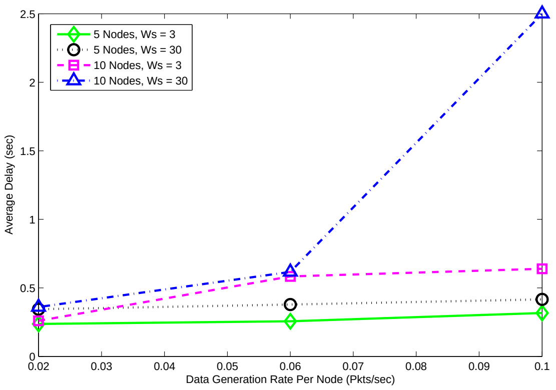
Figure 4.8: Variation of X-PACCA performance with backoff constant and node population for a fixed distance source



(a) Average delay



(b) Packet delivery ratio



(c) Average delay (only 0.02 to 0.1 pkts/s)

Figure 4.9: X-PACCA performance for different W_s values

- With 10 node topology, at highest traffic rate (0.22 pkts/s per node), changing window size from 3 to 30 improves delivery by about 7% but increases delay by about 1300% (from 2 to 26). So, even though it depends on the application, keeping source window size as small as possible is obviously preferable.

- Also note that, in this configuration, sources access the sink in a single hop. This was intentionally enforced in the simulation to investigate the effect of the data transmission source window size. However, in general, this is not the case and in multi-hop scenarios, a packet can be relayed by more than one node. Thus, selection of window size for relays, which depends on k, is more critical than selection of the source window size, especially in dense topologies.

4.2.2 Comparison of MAC Performance

The goal of this part is to simulate and compare the MAC performance of X-PACCA with UWAN-MAC and Slotted FAMA in terms of throughput and delay. Along with the publication [7] on which these comparisons have been made, two different topologies have been used for this purpose, namely, the line topology and a random topology.

Line Topology:

A line topology as shown in figure 4.10 is used to evaluate the end to end delays of X-PACCA, UWAN-MAC and Slotted FAMA. Simulation scenario can be described as follows (all parameter set can be found in Appendix C with simulation id = 7):

- The sink is at the water surface with the input power of $P_{elec} = 10$ kW and a vertex angle of 30°.

- There is only one source, which is node n shown in Figure 4.10.

- Distance between two adjacent nodes is 80 m and transmission range for UWAN-MAC and Slotted FAMA is 90 m. For X-PACCA, the transmission range of the source is 90 m when there are 10 hops.

- The source node generates 60 Byte packets according to a Poisson process where average packet generation rate is $\lambda = 0.1$ pkts/s.

- The distance between the sink and source changes between 80 m and 800 m with the number of hops.

- Nodes give up retransmission upon reaching the maximum backoff stage, $m = 5$. Slot length is set as 0.005 s.

- For Slotted FAMA [22], maximum backoff stage is set as 15, and maximum burst size is taken as 30 packets.

- For UWAN-MAC [15], Average and deviation of cycle period parameters used are 1 and 0.1, respectively.

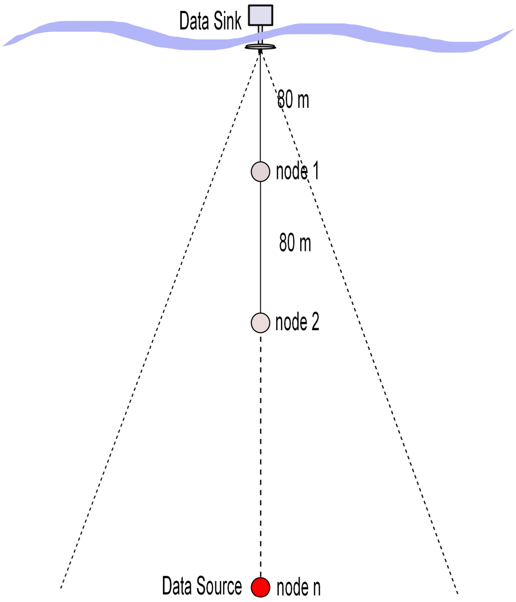


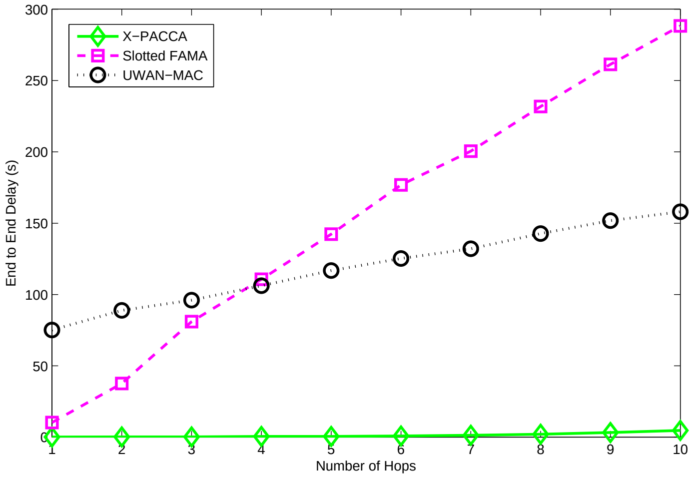
Figure 4.10: Line topology for MAC performance comparison

Random Single Hop Topology:

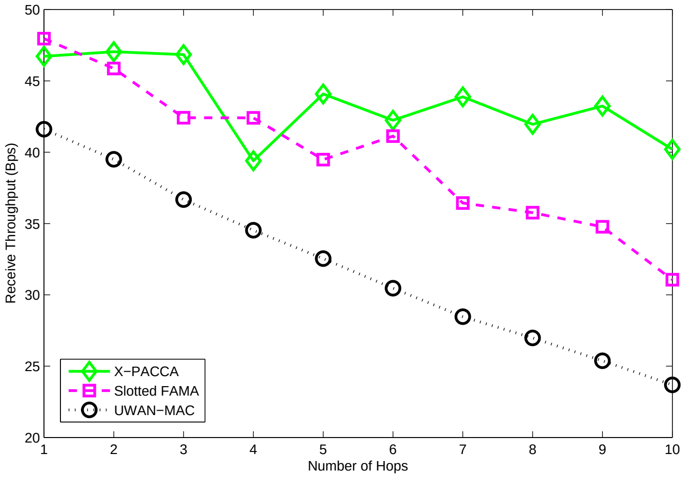
MAC performance of X-PACCA is compared with UWAN-MAC [15] and Slotted FAMA [22] in terms of throughput and delay. Protocol parameters for both UWAN-MAC and Slotted FAMA are also tried to be tuned to give the best performance for the scenario. Details of the simulation scenario are as follows(all parametersset can be found in Appendix C with simulation id = 8):

- The sink is at the water surface with the input power of P_{elec} = 10 kW and a vertex angle of 30°.
- All nodes are also data sources, and the numbers of active node simulated are 2, 4, 8 and 16.
- To get rid of routing effects, network configuration is set such that transmitted packets can reach the sink in a single hop. For this purpose, k is set to 1 and β is set to 0.025.
- Distance between the sink and the source is random with a maximum of 380 m.
- Nodes give up retransmission upon reaching the maximum backoff stage, $m = 5$. Slot length is set as 0.005 s.
- For Slotted FAMA, maximum backoff stage is set as 15, and maximum burst size is taken as 30 packets.
- For UWAN-MAC, Average and deviation of cycle period parameters used are 1 and 0.1, respectively.

End to end delay for line topology is given in Figures 4.11a and throughput is given 4.11b . In Figures 4.12-4.15 (a), throughput performances and in Figures 4.12-4.15 (b), delay performances of all three protocols for random topology for different numbers of sources are plotted. These results lead to the following observations:

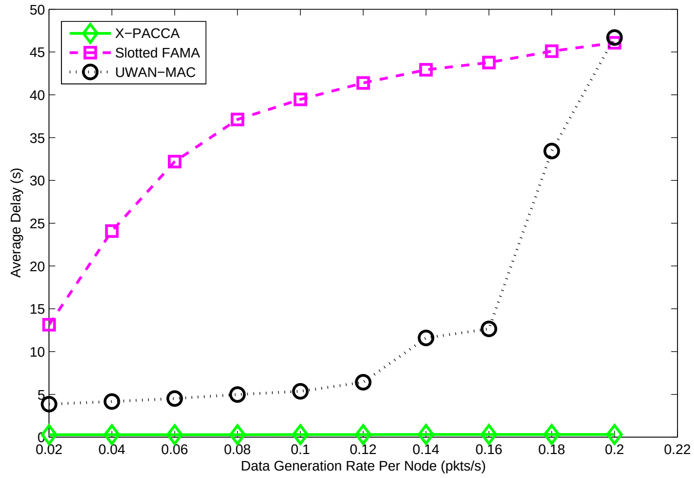


(a) End to end delay

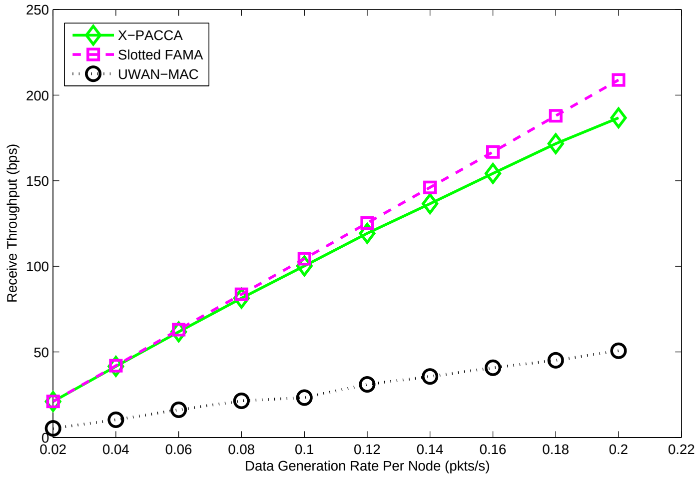


(b) Throughput

Figure 4.11: Comparison of performance of X-PACCA with UWAN-MAC and Slotted FAMA with line topology

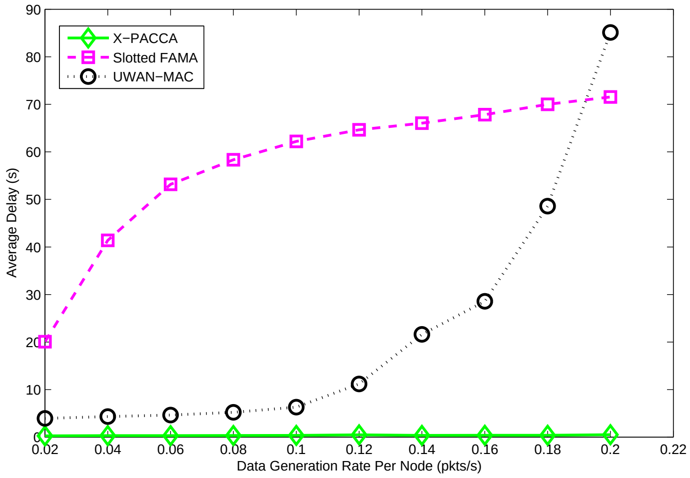


(a) Average delay

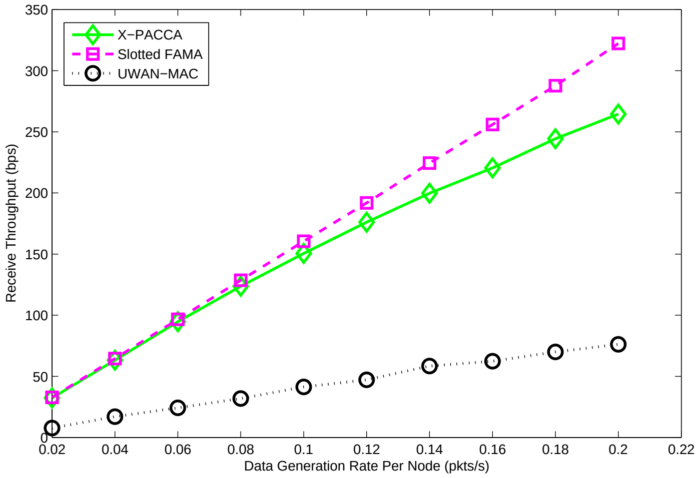


(b) Receive throughput

Figure 4.12: Comparison of MAC performance of X-PACCA with UWAN-MAC and Slotted FAMA with 2 sources

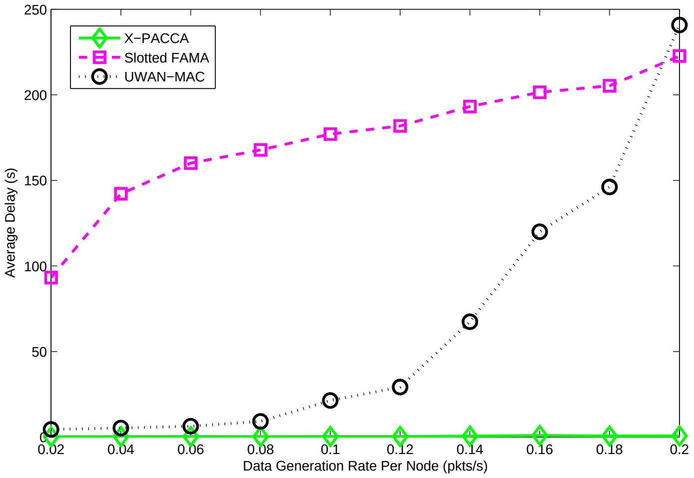


(a) Average delay

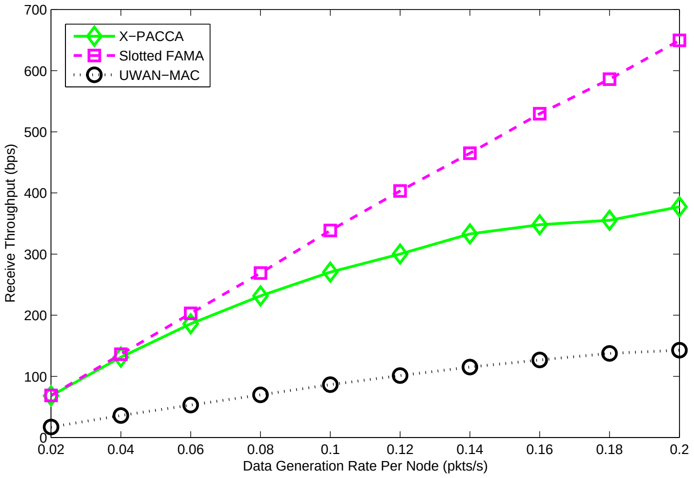


(b) Receive throughput

Figure 4.13: Comparison of MAC performance of X-PACCA with UWAN-MAC and Slotted FAMA with 4 sources

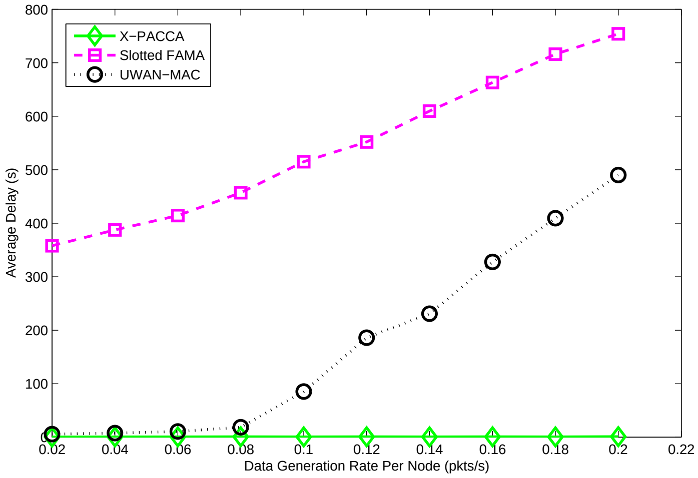


(a) Average delay

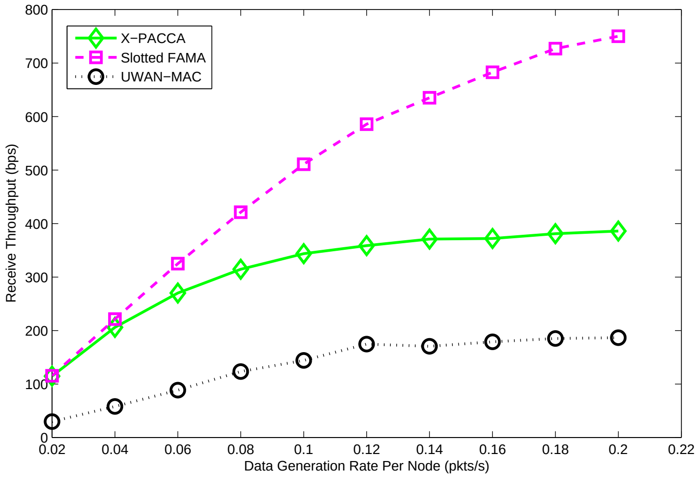


(b) Receive throughput

Figure 4.14: Comparison of MAC performance of X-PACCA with UWAN-MAC and Slotted FAMA with 8 sources



(a) Average delay



(b) Receive throughput

Figure 4.15: Comparison of MAC performance of X-PACCA with UWAN-MAC and Slotted FAMA with 16 sources

- In line topology, X-PACCA outperforms both other protocols both in terms of end-to-end delay and throughput performance. Here, X-PACCA uses the advantage of SPAUSN networks, which is harvested power based transmission range, and packets reaches the sink in fewer hops. Besides, X-PACCA does not have overheads due to TDMA or handshaking.

- UWAN-MAC achieves lower throughputs and higher delays at high packet generation rates compared to the other two protocols. This is an expected result of duty cycling applied in UWAN-MAC, which primarily aims to enhance energy efficiency. In UWAN-MAC, two different nodes might have the same transmitting period resulting in a collision, so when traffic rate increases, collisions dominate, and throughput will no longer increase.

- Slotted FAMA and X-PACCA have similar throughputs at low packet generation rates, but when this rate increases Slotted FAMA achieves better throughput as a result of its RTS/CTS handshaking mechanism and trains of packets technique (burst of packets sent with a single handshake).

- RTS/CTS mechanism in Slotted FAMA decreases collision probability hence results in better throughput values, but significant delays in packet transmissions can be observed when compared with X-PACCA.

- UWAN-MAC, which is recommended mainly for delay-tolerant applications, favors energy efficiency and has lower throughput and high delay at high traffic rates.

- X-PACCA and Slotted FAMA display similar throughputs at low packet generation rates, but when this rate increases, the effect of collisions lowers throughput values of X-PACCA.

4.2.3 Comparison of Routing Performance

In this section, routing performance of X-PACCA is compared with DBR [31] and VBF [27] in terms of packet delivery ratio and average end-to-end delay. The scenario used for simulations is the same as that in [31] except for the topology, operating X-PACCA in a spherical cone shaped geometry, as shown in Figure 4.3. It must be noted that, in this part of the work, only X-PACCA is simulated and results reported in the paper [31] are used directly by extracting from the plots given in the publication. The details of the simulated scenario are (all parameters can be found in Appendix C with simulation id = 9):

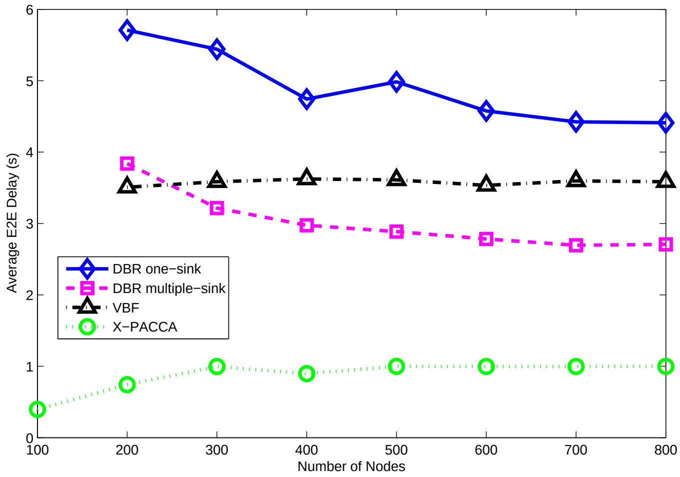
- The sink is at the water surface with the input power of $P_{elec} = 10$ kW and a vertex angle of 30°.

- Single source node is placed at the bottom layer which is about 500 m far from the sink.

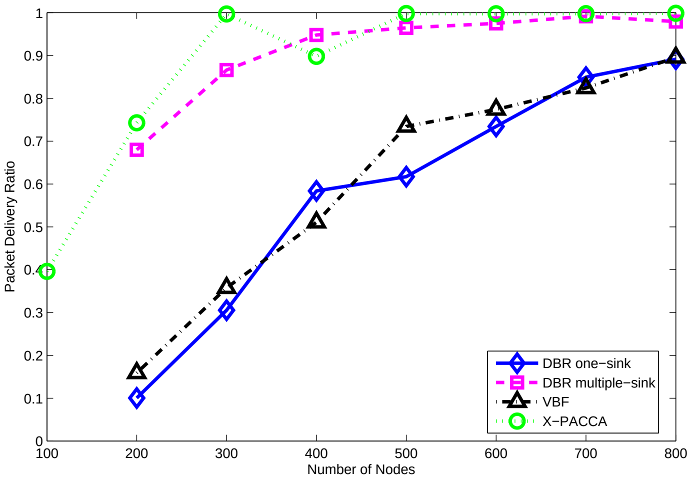
- Transmission ranges in DBR and VBF simulations are fixed for all nodes and set as 100 m. In X-PACCA, transmission ranges are proportional to the power harvested by each node [6]. Using $\beta = 0.0007$ at the nodes, the average transmission range of nodes in X-PACCA is 95 m, which is slightly less than the fixed transmission range used in DBR and VBF, i.e., favoring DBR and VBF at the expense of X-PACCA.

- All simulations last for 5000 s with 50 packets of 50 Bytes, and each data point is a result of 20 simulations (average of 1000 packets).

- Nodes give up retransmission upon reaching the maximum backoff stage, $m = 5$.



(a) Average end-to-end delay



(b) Packet delivery ratio

Figure 4.16: Comparison of routing performance of X-PACCA with DBR and VBF

- Slot length is set as 0.005 s, and k is set to 0.001.

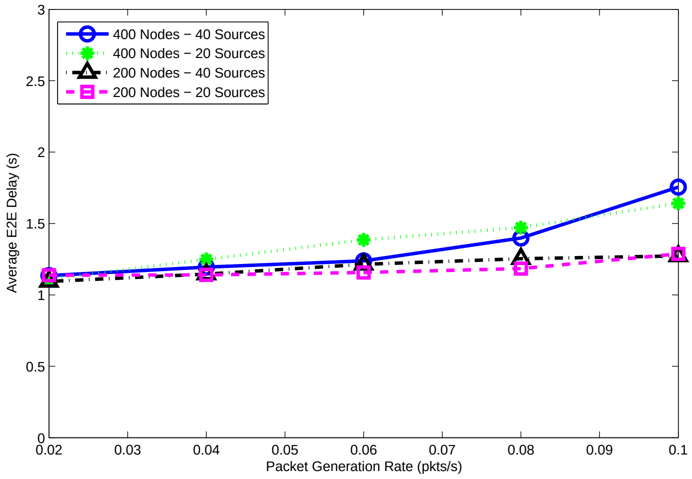
Average end-to-end delay and packet delivery ratio performances are plotted in Figure 4.16a and 4.16b, respectively. Since SPUASN nodes have different transmission ranges increasing with power harvested by node, X-PACCA can send via a smaller number of hops which means smaller delays. X-PACCA also performs best in terms of packet delivery ratio when compared with single sink simulations. Only multi-sink DBR performs better at low node population.

4.2.4 Cross Layer Performance of X-PACCA

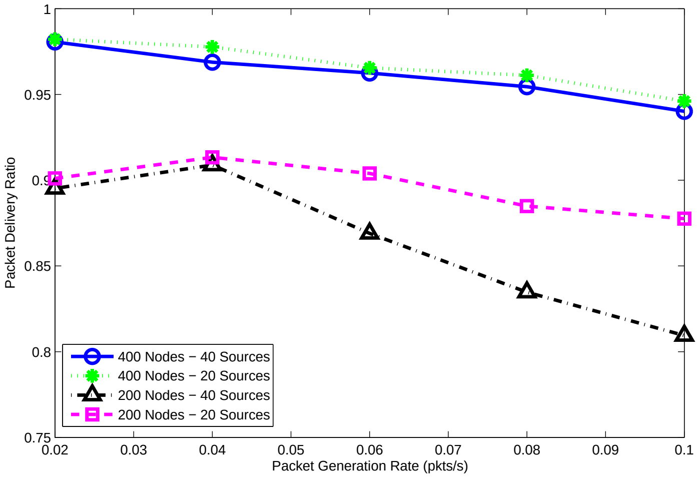
Finally, the cross layer performance of X-PACCA will be presented in terms of the latency and delivery success performance achieved when MAC, relay and transport layer functionalities are collectively effective. For this purpose, the following scenario is investigated with simulations (all parameters can be found in Appendix C with simulation id = 10):

- The sink is at the water surface with the input power of P_{elec} = 10 kW and a vertex angle of 30°.
- There are a total of 200 and 400 randomly placed active nodes in the network and 20 or 40 of them are randomly selected as sources with a maximum possible distance of 535 m to the sink.
- Randomly selected source nodes generate Poisson traffic of 50 B packets, contributing to the total average packet generation rates depicted in the figures.
- Nodes give up retransmission upon reaching the maximum backoff stage, m = 5.
- X-PACCA slot length was taken as 0.005 s, W_s was set to 30, and the backoff constant was k= 0.001.
- β was taken as 0.0007, leading to an average transmission range of 95 m.

With the investigated configuration, packets generated by 40 sources are delivered over a maximum distance of 535 m within at most 1.5-2 s, at a delivery ratio of at least 80%. Populations of 200 and 400, which are above the minimum number of sensors to achieve 1-coverage in the given volume, as analyzed in [6], and hence, even better performance may be obtained with networks of sparser populations.



(a) Average end-to-end delay



(b) Packet delivery ratio

Figure 4.17: Cross layer performance of X-PACCA in terms of latency and delivery ratio.

CHAPTER 5

CONCLUSION

In this thesis, the performance of Cross Layer Power Adaptive CSMA/CA (X-PACCA) has been studied with different simulation scenarios. All simulations were performed in Aqua-Sim, which is an NS-2 based simulation tool for underwater acoustic sensor networks. Before starting performance evaluation simulations of X-PACCA, simulation infrastructures and metric calculation methods were verified by comparing with the results reported in [1]. X-PACCA performance evaluations were conducted through investigating the impact of design parameters on the performance and by comparing with other protocols that have been proposed in the literature for UASN. The performance metrics taken into consideration were end-to-end delay, packet delivery ratio and network throughput.

Since X-PACCA does not have a restriction on the selection of slot length, determination of an appropriate slot length requires some other criteria. For this purpose, different slot lengths were simulated with the same scenario, and it was shown that there is a trade-off between end-to-end delay and success probability when selecting this parameter. Shorter slot lengths decreases the dependence of performance over k and with proper selection of both k and slot length, a minimum delay point can be obtained.

Next, the impact of node density was investigated with three different vertex angles of the conical geometry of energy supply, and different node numbers. Higher sensor densities result in higher packet delivery ratios as expected, packet delivery ratios very close to 100% were obtained for all scenarios. Since in sparse networks, the number of possible relay nodes of a packet is lower when compared to dense ones, X-PACCA backoff constant, k has a higher effect on network performance, especially on latency. Simulations that investigated the effect of vertex angle showed that performance can be optimized by adjusting this angle, if this is possible. In other words, a higher number of nodes that are close to the surface can be accessed by increasing the energy supply vertex angle and data from deeper nodes can be gathered by narrowing this angle. Adjusting vertex angle, if and whenever practically possible, is shown to be a flexibility of SPUASN and X-PACCA enables optimization of performance through this adjustment. Finally, simulations that investigated the effect of the starting window size, W_s, showed that optimal selection of W_s depends on source density and traffic rates. Even though increasing W_s improves delivery ratio, it might lead to high delays which is not preferable in most cases.

To study MAC performance of the protocol, two different topologies were simulated with X-PACCA, UWAN-MAC [15] and Slotted FAMA [22]. To examine the end-to-end delay performance of the protocols, a line topology was used. A single hop random topology with different number of sources was used to study both throughput and average delay. Simulation results showed that X-PACCA performs better than the other protocols in terms of latency in all scenarios. Even though X-PACCA and Slotted FAMA have similar throughput values at low traffic levels; at high traffic, Slotted FAMA achieves better throughput with the advantage of its RTS/CTS handshaking mechanism and trains of packets technique which effect fairness and end-to-end delays negatively. Since TDMA based UWAN-MAC mainly aims at energy efficiency, it has lower throughputs and higher delays when compared to energy constraint free X-PACCA.

Routing performance of the protocols is examined by setting a similar topology to that implemented in [31] and comparing the obtained results with the ones given for DBR [31] and VBF [27] protocols in the same article. Simulation results indicated that X-PACCA exploits the advantage of increasing transmission ranges with harvested power and sources can access the sink over a smaller number of hops which means smaller delays. Packet delivery ratio of X-PACCA also outperforms DBR and VBF when compared to the single sink results given in the paper.

Finally, the cross layer performance of X-PACCA is investigated with different number of nodes and promising performances have been predicted in terms of end-to-end delay and packet delivery ratio.

Simulations that were carried out in the scope of this study have shown that X-PACCA performs quite well for SPUASN. While providing the advantage of energy constraint removal, X-PACCA performs considerably better than other protocols used for comparison in this study.

As a future work, combinations of different high performance MAC and routing protocols can be simulated and compared with the cross layer performance of X-PACCA. Effects of topology and effects of packet size on performance of the protocol are also other points that require further analysis.

The obtained results also imply that, the performance of X-PACCA depends highly on the backoff constant, k. In fact, this is an expected behavior since the basic idea of the protocol depends on this parameter as shown in equation 4.1. Even though an optimum k parameter can be obtained by simulating the scenario to be implemented, changes in network topology due to node failure or mobility, which is known to be very common for UASN, will result in poor performances with initially obtained k parameter. For this reason, some adaptive methods for updating k according to these changes or enabling control of this parameter by an outside operator require further study.

In sparse random deployment of sensor nodes, there might arise some void areas, where there are no eligible relays that have more power than the transmitter. Since network connectivity can still be achieved over some other lower power nodes, some recovery mechanisms have to be devised to handle such cases. Also deserving further study are issues such as operation with multiple data sinks and possible configurations of RPUASN with multiple energy sources.

REFERENCES

[1] K. Manijeh, J. Reza, and M. Reza. Comparative Performance Evaluation of MAC Layer Protocols for Underwater Wireless Sensor Networks. *In Modern Applied Science*, 6(3):65 – 72, March 2012.

[2] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: Research challenges. *AD HOC NETWORKS (ELSEVIER)*, 3:257–279, 2005.

[3] John Heidemann, Milica Stojanovic, and Michele Zorzi. Underwater sensor networks: Applications, advances, and challenges. *Philosophical Transactions of the Royal Society–A*, 370(1958):158–175, 2011.

[4] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS Conference*, 2005.

[5] S. Roundy, P.K. Wright, and J.M. Rabaey. *Energy scavenging for wireless sensor networks: with special focus on vibrations*. Kluwer Academic Publishers, 2004.

[6] A. Bereketli and S. Bilgen. Remotely powered underwater acoustic sensor networks. *Sensors Journal, IEEE*, 12(12):3467–3472, 2012.

[7] A. Bereketli, H. Turken, and S. Bilgen. Cross Layer Power Adaptive CSMA/CA for Sink Powered Underwater Acoustic Sensor Networks. *12th Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, To appear.

[8] Liu Lanbo, Zhou Shengli, and Cui Jun-Hong. Prospects and problems of wireless communication for underwater sensor networks. *Wirel. Commun. Mob. Comput.*, 8(8):977–994, October 2008.

[9] Milica Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. In *Proceedings of the 1st ACM international workshop on Underwater networks*, WUWNet '06, pages 41–47, New York, NY, USA, 2006. ACM.

[10] S. Singh, S.E. Webster, L. Freitag, L.L. Whitcomb, K. Ball, J. Bailey, and C. Taylor. Acoustic communication performance of the whoi micro-modem in sea trials of the nereus vehicle to 11,000 m depth. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1–6, 2009.

[11] D. Makhija, P. Kumaraswamy, and R. Roy. Challenges and design of mac protocol for underwater acoustic sensor networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–6, 2006.

[12] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.*, 7(3):537–568, May 2009.

[13] Paolo Casari and Michele Zorzi. Protocol design issues in underwater acoustic networks. *Computer Communications*, 34(17):2013 – 2025, 2011.

[14] C. Petrioli, R. Petroccia, and M. Stojanovic. A comparative performance evaluation of MAC protocols for underwater sensor networks. In *OCEANS 2008*, pages 1–10, 2008.

[15] V. Rodoplu and M.K. Park. An energy-efficient mac protocol for underwater wireless acoustic networks. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1198–1203 Vol. 2, 2005.

[16] D. Pompili, T. Melodia, and I.F. Akyildiz. A CDMA-based Medium Access Control for UnderWater Acoustic Sensor Networks. *Wireless Communications, IEEE Transactions on*, 8(4):1899–1909, 2009.

[17] Peng Xie and Jun-Hong Cui. R-MAC: An Energy-Efficient MAC Protocol for Underwater Sensor Networks. In *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, pages 187–198, 2007.

[18] A.A. Syed, Wei Ye, and J. Heidemann. T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 231–235, 2008.

[19] B. Peleato and M. Stojanovic. Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks. *Communications Letters, IEEE*, 11(12):1025–1027, 2007.

[20] Yin-Jun Chen and Hao-Li Wang. Ordered CSMA: a collision-free MAC protocol for underwater acoustic networks. In *OCEANS 2007*, pages 1–6, 2007.

[21] Wen-Hwa Liao and Chih-Chien Huang. SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks. *Sensors Journal, IEEE*, 12(6):1686–1694, 2012.

[22] M. Molins and M. Stojanovic. Slotted FAMA: a MAC protocol for underwater acoustic networks. In *OCEANS 2006 - Asia Pacific*, pages 1–7, 2006.

[23] N. Chirdchoo, Wee-Seng Soh, and Kee Chua. RIPT: A Receiver-Initiated Reservation-Based Protocol for Underwater Acoustic Networks. *Selected Areas in Communications, IEEE Journal on*, 26(9):1744–1753, 2008.

[24] Zhong Zhou, Zheng Peng, Jun-Hong Cui, and Zaihan Jiang. Handling Triple Hidden Terminal Problems for Multichannel MAC in Long-Delay Underwater Sensor Networks. *Mobile Computing, IEEE Transactions on*, 11(1):139–154, 2012.

[25] Peng Xie, Zhong Zhou, Zheng Peng, Hai Yan, Tiansi Hu, Jun-Hong Cui, Zhijie Shi, Yunsi Fei, and Shengli Zhou. Aqua-Sim: An NS-2 based simulator for underwater sensor networks. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1–7, 2009.

[26] J. J. Garcia-Luna-Aceves and Chane L. Fullmer. Floor acquisition multiple access (FAMA) in single-channel wireless networks. *Mob. Netw. Appl.*, 4(3):157–174, October 1999.

[27] Peng Xie, Jun-Hong Cui, and Li Lao. VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks. In *NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, volume 3976 of *Lecture Notes in Computer Science*, pages 1216–1221. Springer Berlin Heidelberg, 2006.

[28] Josep Miquel Jornet, Milica Stojanovic, and Michele Zorzi. Focused beam routing protocol for underwater acoustic networks. In *Proceedings of the third ACM international workshop on Underwater Networks*, WuWNeT '08, pages 75–82, New York, NY, USA, 2008. ACM.

[29] Daeyoup Hwang and Dongkyun Kim. DFR: Directional flooding-based routing protocol for underwater sensor networks. In *OCEANS 2008*, pages 1–7, 2008.

[30] N. Nicolaou, A. See, Peng Xie, Jun-Hong Cui, and D. Maggiorini. Improving the robustness of location-based routing for underwater sensor networks. In *OCEANS 2007 - Europe*, pages 1–6, 2007.

[31] Hai Yan, ZhijieJerry Shi, and Jun-Hong Cui. DBR: Depth-Based Routing for Underwater Sensor Networks. In *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, volume 4982 of *Lecture Notes in Computer Science*, pages 72–86. Springer Berlin Heidelberg, 2008.

[32] Youngtae Noh, Uichin Lee, Paul Wang, Brian Sung Chul Choi, and Mario Gerla. Vapr: Void-aware pressure routing for underwater sensor networks. *Mobile Computing, IEEE Transactions on*, 12(5):895–908, 2013.

[33] Muhammad Ayaz, Imran Baig, Azween Abdullah, and Ibrahima Faye. A survey on routing techniques in underwater wireless sensor networks. *Journal of Network and Computer Applications*, 34(6):1908 – 1927, 2011.

[34] M.C. Domingo and R. Prior. A distributed clustering scheme for underwater wireless sensor networks. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5, 2007.

[35] The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

[36] Aqua-Sim-1.0 Release package. `http://ubinet.engr.uconn.edu/aqua_packages/aqua-sim-1.0.tgz`.

[37] Matthew T. Tran. Visualizing the Simulation of 3-D Underwater Sensor Networks. `http://digitalcommons.uconn.edu/srhonors_theses/72`, 2009.

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

APPENDIX A

NECESSARY CHANGES APPLIED ON AQUA-SIM 1.0

A.1 Changes for UWAN-MAC

A.1.1 Registering packet header for UWAN_SYNC

Packet Header for UWAN_SYNC is required to be registered. For this purpose, the header name UWAN_SYNC was inserted at line 180 of the file: `/AquaSim/ns-2.30/tcl/lib/ns-packet.tcl`

```
  ...
  OTMAN
  UW_ALOHA
+ UWAN_SYNC
# Other:
  Encap   # common/encap.cc
  IPinIP  # IP encapsulation
  ...
```

A.1.2 Making packet size configurable via simulation file

Since different packet sizes are used at different simulations, this parameter had to be made configurable via simulation file (.tcl file).

For this purpose, first the variable “packet_size_” was added to the definition of UWAN_MAC class in the header file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/uwan-mac/uwan-mac.h`

```
  ...
  static Time  hello_tx_len;
  static Time  WakePeriod_;

+ int packet_size_;
  ScheduleQueue WakeSchQueue_;
  ...
```

Then this variable was bound to a parameter with the same name to make it accessible via simulation file.

The following line was added to the constructor of UWAN_MAC class in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/uwan-mac/uwan-mac.cc`

```
  ...
  bind("StdCyclePeriod", &StdCyclePeriod_);
+ bind("packet_size_", &packet_size_);
  start_timer_.resched(0.001);
  ...
```

Finally this variable was written over previous constant value, which is 1600 bits.

The following changes were implemented to the TxProcess() and start() functions of UWAN_MAC class in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/uwan-mac/uwan-mac.cc`

```
  void UWAN MAC::TxProcess(Packet  p)
  {

  - HDR_CMN(p)->size() = 1600;
  + HDR_CMN(p)->size() = packet_size_;

  ...
```

```
  void UWAN MAC::start()
  {
    Random::seed_heuristically();
    SYNCSchedule(true);
  - MaxTxTime_ = 1610 + encoding_efficiency_/bit_rate_;
  + MaxTxTime_ = (packet_size_+10)*encoding efficiency_/bit_rate_;
    hello_tx_len = (hdr SYNC::size())*8*encoding_efficiency_/bit rate_;
    ...
```

A.1.3 Use next hop set via simulation file

Routing protocols are not taken into account; this work only compares the performance of MAC protocols. So in both stages of simulations, next hop for all nodes were set via simulation file, routing protocols do not have any effect.

The following changes were implemented in TxProcess() and sendoutPkt() functions of UWAN_MAC class in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/uwan-mac/uwan-mac.cc`

```
void UWAN MAC::TxProcess(Packet  p)
{
  ...
+ hdr mac* mach =HDR_MAC(p);
+ hdr cmn* cmh=HDR_CMN(p);
+ UnderwaterSensorNode* n=(UnderwaterSensorNode*) node_;
+ if (n->setHopStatus){
+    cmh->next hop()=n->next_hop;
+    mach- macDA() = n->next_hop;
+ }

  PacketQueue_.push(p);
  ...
```

```
void UWAN_MAC::sendoutPkt(Time NextCyclePeriod)
{
  ...
```

```
  nsaddr_t next_hop = MAC_BROADCAST;

- if( neighbors_.size() != 0 )
-   set nsaddr_t :: iterator pos = neighbors_.begin();
-   for(uint i=0; i<next_hop_num; i++, pos++);
-   cmh->next_hop() =  pos;
-   next hop_num = (next_hop_num+1)%neighbors_.size();
-   vbsh->target_id.addr_ = cmh  next_hop();
-
  hdr_mac* mh=hdr_mac::access(pkt);
  ...
```

A.2 Changes for Slotted FAMA

A.2.1 Correct a time variable type

The variable “data_sending_interval_” stores a value of type “Time”. So the type of this variable was corrected in the declaration of SFAMA class at line 181 in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.h`

```
void UWAN MAC::sendoutPkt(Time NextCyclePeriod)
{
  ...
  int max_burst_; /*maximum number of packets in the train*/
- int data_sending_interval_;
+ Time data_sending_interval_;
  SFAMA_CallBackHand
  ...
```

A.2.2 Use next hop set via simulation file

Routing protocols are not taken into account; this work only compares the performance of MAC protocols. So in both stages of simulations, next hop for all nodes were set via simulation file, routing protocols do not have any effect.

The following changes were implemented in fillDATA() function and a header was added to SFAMA class in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.cc`

```
void UWAN MAC::sendoutPkt(Time NextCyclePeriod)
{
   include "sfama.h"
+  include "underwatersensor/uw_common/underwatersensornode.h"
  ...
```

```
  Packet  SFAMA::fillDATA(Packet  data_pkt)
{
  hdr_cmn* cmh = HDR_CMN(data_pkt);
  hdr_mac* mach = HDR_MAC(data_pkt);
  hdr SFAMA* SFAMAh = hdr_SFAMA::access(data_pkt);
+ UnderwaterSensorNode  n=(UnderwaterSensorNode ) node_;

  cmh->size() += hdr SFAMA::getSize(hdr_SFAMA::SFAMA_DATA);
  cmh->txtime() = getTxTime(cmh->size());
```

```
  cmh->error() = 0;
  cmh->direction() = hdr_cmn::DOWN;

  mach->macSA() = index_;
- mach->macDA() = cmh->next_hop();
+ mach->macDA() = n->next_hop;

  SFAMAh  packet_type = hdr_SFAMA::SFAMA_DATA;
  ...
```

A.2.3 Implementing Trains of Packets Technique

Trains of packets technique is crucial for increasing throughput of Slotted FAMA. But in Aqua-Sim 1.0 the implementation of this technique is not complete. For this purpose, the following changes were applied to SFAMA class:

- A new field was added to the SFAMA packet header to inform receiver whether the burst goes on. The following changes were implemented in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama-pkt.h`

```
struct hdr_SFAMA{
  //nsaddr_t SA;
  //nsaddr_t DA;
  uint16_t SlotNum;  //the number of slots required for transmitting the ←↩
      DATA packet
+ uint16_t burstGoesOn;

  enum PacketType {
  ...
```

```
static int getSize(enum PacketType p_type)
  int pkt_size = 2*sizeof(nsaddr_t); //source and destination addr in hdr mac
  if( p_type == SFAMA_RTS || ptype == SFAMA_CTS )
- pkt_size += sizeof(u_int16_t)+1; //size of packet_type and slotnum
+ pkt_size += 2*sizeof(u_int16_t)+1; //size of packet_type changed with the ←↩
    effect of new added field   burstGoesOn
  }
  return pkt size;
}
```

- A new private variable was declared and initialized to keep whether burst goes on or not. An integer variable “does_burst_goes_on_” was declared in the header file of SFAMA: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.h`

```
...
private:
  //index_ is the mac address of this node
  Time guard_time_; //need to be binded
  Time slot_len_;

  bool is_in_round;
  bool is_in_backoff;

  int max_backoff_slots_;
+ int does_burst_goes_on_;
  ...
```

Initialize this variable in file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.cc`

```
SFAMA::SFAMA():UnderwaterMac(), status_(IDLE WAIT), guard_time_(0.00001), slot↩
     _len_(0.0),is_in_round(false), is_in_backoff(false), max_backoff_slots_(4)↩
     , max_burst_(1),data_sending_interval_(0.0000001), callback_handler(this),↩
      status_handler(this),slotinit_handler(this), wait_send_timer(this),wait_↩
     reply_timer(this), backoff_timer(this), datasend_timer(this),
+ does_burst_goes_on_(0)
{
  bind("guard_time_",  guard_time_);
  ...
```

- Receiver side: Process data received as a burst

  The function processDATA() of SFAMA class was modified to handle packets coming as a burst. Changes were applied to the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.cc`

```
void SFAMA::processDATA(Packet* data_pkt)
{
- //hdr_SFAMA* SFAMAh = hdr_SFAMA::access(data_pkt);
+ hdr_SFAMA* SFAMAh = hdr_SFAMA::access(data_pkt);
  hdr_mac* mach = HDR_MAC(data_pkt);

  if( mach->macDA() == index_    getStatus() == WAIT_RECV_DATA ) {
+   if(SFAMAh->burstGoesOn == 0)
+   {
      //send ACK
      stopTimers();
      setStatus(WAIT_SEND_ACK);

      wait_send_timer.pkt_ = makeACK(mach->macSA());
      wait_send_timer.resched(getTime2ComingSlot(NOW));
+   }

    /*send packet to upper layer*/
    hdr_cmn::access(data_pkt)->size()  = hdr_SFAMA::getSize(hdr_SFAMA::SFAMA_↩
        DATA);
    sendUp(data_pkt->copy()); /*the original one will be released*/
  }
  else {
  ...
```

- Sender side: Prepare data to send as a burst The newly added variable keeping the state of burst (*does_burst_goes_on_*), was set in the function according to the status of sending queue. Changes were applied in the file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.cc`

```
void SFAMA::DataSendTimerProcess()
{

#ifdef SFAMA_DEBUG
  printf("DataSendTimerProcess(before)");
  printAllQ();
#endif
  if( !SendingPktQ_.empty() ) {
    Packet* pkt = SendingPktQ_.front();
    Time txtime = hdr_cmn::access(pkt)->txtime();
    Backup_SendingPktQ_.push(pkt);

```

```
    SendingPktQ_.pop();
+   if(SendingPktQ_.empty())
+   {
+     does_burst_goes_on_ = 0;
+   }
+   else
+   {
+     does_burst_goes_on_ = 1;
+   }
    sendDataPkt(pkt->copy());

    datasend_timer.resched(data_sending_interval_+txtime);

  }
  else {
  ...
```

sendDataPkt() function is the function which puts the packets to lower layer, this function also needed to be modified to be capable of sending data as a burst. Changes include, adding the state of burst to SFAMA packet header, rescheduling next packet in the queue and increasing wait time for ACK packet from 1 slot to 2 slots. The file modified was: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.cc`

```
void SFAMA::sendDataPkt(Packet* pkt)
{
  hdr_cmn* cmh=HDR_CMN(pkt);
  //hdr_FAMA* FAMAh = hdr_FAMA::access(pkt);
+ hdr_SFAMA* SFAMAh = hdr_SFAMA::access(pkt);

+ SFAMAh->burstGoesOn = does_burst_goes_on_;

  cmh->direction() = hdr_cmn::DOWN;
+ double txtime=cmh->txtime();

  UnderwaterSensorNode* n=(UnderwaterSensorNode*) node_;

  switch( n->TransmissionStatus() ) {
    case SLEEP:
      Poweron();
    case IDLE:
      n->SetTransmissionStatus(SEND);
      cmh->timestamp() = NOW;
      #ifdef SFAMA_DEBUG
      printf("%f: node %d send to node %d",
           NOW, HDR_MAC(pkt)->macSA(), HDR_MAC(pkt)->macDA());
      #endif
      sendDown(pkt);

+     status_handler.slotnum() = 2;
+     Scheduler::instance().schedule( status_handler, status_event, +txtime);
      break;
    case RECV:
    ...
```

StatusProcess() function is modified to preserve status as WAIT_SEND_DATA until all packets in the burst finish. After the burst is finished, status changes to WAIT_RECV_ACK. Changes were applied on file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/slotted-fama/sfama.cc`

```
void SFAMA::StatusProcess(int slotnum)
{
  UnderwaterSensorNode* n=(UnderwaterSensorNode*) node_;
  n->SetTransmissionStatus(IDLE);
```

```
  switch(getStatus()) {
    case WAIT_SEND_RTS:
    slotnum = 1;
    setStatus(WAIT_RECV_CTS);
    break;
    case WAIT_SEND_CTS:
    //slotnum += 1;
    setStatus(WAIT_RECV_DATA);
    break;
    case WAIT_SEND_DATA:
    //cannot reach here
+   if(does_burst_goes_on_ == 1)
+   {
+     return;
+   }
    slotnum = 1;
    setStatus(WAIT_RECV_ACK);
    //wait_reply time has been scheduled.
    return;
    ...
```

A.3 Changes for X-PACCA

A.3.1 Registering packet header for X-PACCA

Packet header for X-PACCA needed to be registered. For this purpose, the header name “X-PACCA” was inserted at line 181 of the file: `/AquaSim/ns-2.30/tcl/lib/ns-packet.tcl`

```
  ...
  OTMAN
  UW_ALOHA
  UWAN_SYNC
+ X_PACCA
# Other:
  Encap   # common/encap.cc
  IPinIP  # IP encapsulation
```

Also the following lines should be added to file: `/AquaSim/ns-2.30/common/packet.h`

```
#define HDR_DBR(p)     (hdr_dbr::access(p)) /* hai s dbr */
  #define HDR_LMS(p)   (hdr_lms::access(p))
+ #define HDR_X_PACCA(p) (hdr_x_pacca::access(p)) // ht

  enum packet_t {
  PT_TCP,
  PT_UDP,
  PT_CBR,
  ...
  PT_UW_MESSAGE, /*the packet generated by uw_sink agent*/
  PT_UWALOHA,
  PT_UW_DROUTING,
+ PT_X_PACCA,


  PT_NTYPE // This MUST be the LAST one
};
...
class p_info {
public:
```

```
  p_info() {
    name_[PT_TCP]= "tcp";
    name_[PT_UDP]= "udp";
    name_[PT_CBR]= "cbr";
    ...
    name_[PT_UWALOHA] = "UW Aloha ACK";
    name_[PT_UW_DROUTING]= "uw_drouting";
+   name_[PT_X_PACCA]= "x_pacca";

    name_[PT_NTYPE]= "undefined";
    }
```

A.3.2 Adding RPUASN parameters to underwater sensor node

RPUASN nodes are a special type of underwater sensor nodes. So some variables added to underwater sensor node class to handle this specialty. The following changes were implemented in file: `/AquaSim/underwatersensor/uw_common/underwatersensornode.h`

```
class UnderwaterSensorNode : public MobileNode
{
  friend class UnderwaterPositionHandler;
  ...
  Event uw_pos_intr_;

+ // Added for X_PACCA
+ int rpuasnNode_; // is this node an RPUASN node
+ int rpuasnSink_; // is this node an RPUASN sink
+ double rpuasnPowerHarvested_; // Pharv: power budget harvested from the external ↩
    acoustic source
+ double rpuasnPowerToTxConstant_; // c: MPL = c*Pharv

protected:
  ...
```

A.3.3 Skip network layer

To skip network layer, main entry point of a packet coming from up or down to “VectorbasedforwardAgent” class is changed as following and in all simulations vector based routing class is selected as routing class. Changes that were implemented simply means passing routing layer without any processing. This was required to make X-PACCA a cross-layer protocol even it is implemented at MAC level.

File for changes: `/AquaSim/ns-2.30/underwatersensor/uw_routing/vectorbasedforward.cc`

```
void VectorbasedforwardAgent::recv(Packet* packet, Handler*)
{
  hdr_uwvb* vbh = HDR_UWVB(packet);
  ...
+ if(HDR_CMN(packet)->ptype() == PT_X_PACCA)
+ { // if it is a X_PACCA packet, don t touch it, just forward up or down
+   if(HDR_CMN(packet)->direction_ == hdr_cmn::UP)
+   {
+     DataForSink(packet);
+   }
+   else
```

```
+   {
+     MACsend(packet, 0);
+   }
+   return;
+ }
  ...
```

A.3.4 Use already implemented sink agent

The following changes were implemented to use already implemented sink agent with X-PACCA. The following variable was added in file: `/AquaSim/ns-2.30/underwatersensor/uw_common/uw_sink.h`

```
// Class SinkAgent as source and sink

class UWSinkAgent : public Agent {
 public:
  UWSinkAgent();
  ...
  int explore_status;
+ int isXPACCANetwork_;
  UWSink_Timer sink_timer_;
  ...
}
```

A.3.5 Use harvested power for packet transmission

In Aqua-Sim, all nodes use same power level for packet transmission, but RPUASN nodes use different power levels proportional to harvested power. To reflect this property, the following changes were implemented in file: `/AquaSim/ns-2.30/underwatersensor/uw_mac/underwaterphy.cc`

```
void UnderwaterPhy::sendDown(Packet *p)
{
  UnderwaterSensorNode* n1;

+ UnderwaterSensorNode *n = (UnderwaterSensorNode*) node_;
+ if(n->rpuasnNode_) // is this an RPUASN node
+ {
+   Pt_ = n->rpuasnPowerHarvested_*n->rpuasnPowerToTxConstant_;
+ }

  assert(initialized());
  ...
}
```

APPENDIX B

X-PACCA CLASS DEFINITION

```
Class X_PACCA
{
  Time slot_len_; // slot length used in protocol
  int currentBackoffRound_; // current backoff round (i)
  int maxBackoffRound_; // maximum backoff round (m)
  int minBackoffWindow_; // minimum backoff window (Wmin)
  int currentBackoffWindow_; // current backoff window (Wi)
  double backoffWindowConstant_; // backoff window constant (k)
  int sourceBackoffWindow_; // source backoff window (Ws)
  int remainingSlotsToTx_; // remaining slots to transmit (cnt)
  double wfaThresh_; // threshold value for waiting ACK from Sink
  Time ignoreTimeout_; // ignore timeout (D)
  int payloadSizeInBytes_; // payload size in bytes

  queue EventInfo*  WFAQ_; // waiting for ACK queue (WFA)
  queue Packet*  MTQ_; // MAC transmit queue (MTQ)
  queue PacketID*  IGSQ_; // ignore list queue (IGS)

  enum X_PACCA_Status status_; // status of the protocol,
  // one  of the followings: Idle waiting, Sending data, Receiving data,
  // Sending ACK, Receiving ACK
  void StatusProcess(int slotnum); // handles the status of the   protocol by also
  //considering collisions.
  void RecvProcess(Packet*); // the process which is called just  after a packet
  // received
  void TxProcess(Packet*); // the process which is called just  after a packet
  // created to transmit
  void WaitSendTimerProcess(Packet* pkt); // handles sending packet  at correct
  // slot by considering backoff and sensing channel.
  void WaitAckTimerProcess(); // handles checking reception of ACK  for an already
  // transmitted packet.
  void PacketIdDeleteTimerProcess(PacketID* pktId); // manages IGS  queue. It,
  // deletes packet IDs from list if they are timed out.
  Packet* ConstructPkt(Packet* data_pkt); // the function used to  create an X PACCA
  // packet by using the payload coming from  application layer.
  void processRelayPacket(Packet* data_pkt); // the function that  is called for
  // processing a packet coming from a relay node (paket  type = 1).
  void processSinkPacket(Packet* ack_pkt); // the function that is  called for
  // processing a packet coming from sink(paket type = 0).
  void sendPkt(Packet* pkt); // send packet to physical layer.
  int randBackoffSlots(); // generates a random number by using   current
  // backoff window.
  bool SenseEvent(int evid); // if packet transmission fails, this  function is
  // called to check whether the event is still active.
  // void prepareSendingDATA() calculates backoff Windows if there are packets
  // to send in MTQ, schedules packet send timer accordingly.
  void prepareSendingACK(); // schedules ACK send timer with source window size.
  void postProcessSendingDATA(); // After packet transmission completed,
  // deletes packet from MTQ, pushes its info to WFA and IGQ.
  void postProcessSendingACK(); // After sending ACK, deletes it from MTQ.
};
```

APPENDIX C

SIMULATION PARAMETERS

Table C.1: Simulation Ids vs Simulation Descriptions

Simulation Id	**Description of the Simulation**
1	Impact of Slot Length
2	Impact of Node Density when vertex angle = 20°
3	Impact of Node Density when vertex angle = 30°
4	Impact of Node Density when vertex angle = 45°
5	Impact of Vertex Angle for Fixed Distance Source
6	Effect of Source Window Size
7	Comparison of MAC Performance with Line Topology
8	Comparison of MAC Performance with Random Single Hop Topology
9	Comparison of Routing Performance
10	Cross Layer Performance of X-PACCA

Table C.2: Acoustic power source parameters

Sim Id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Acoustic Source Power (kW)	10	10	10	10	10	10	10	10	10	10
Acoustic Source Frequency (kHz)	10	10	10	10	10	10	10	10	10	10
Electro-acoustic power conversion efficiency (η)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vertex angle of acoustic transmission (Degrees)	20	20	30	45	20, 30, 45	20	30	30	30	30

Table C.3: RPUASN node parameters

Sim Id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Required power for a node to operate (P_{req} in Watts)	0.5	0.5	0.5	0.5	0.2	2	0.2	1	0.5	0.5
Number of harvesting hydrophones (n)	5	5	5	5	5	5	5	5	5	5
Receiving voltage sensitivity (RVS in dB)	-150	-150	-150	-150	-150	-150	-150	-150	-150	-150
Hydrophone impedance (R_p in Ω)	125	125	125	125	125	125	125	125	125	125
Harvesting efficiency	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Power to transmit power constant (β)	0.005	0.005	0.005	0.005	0.005	0.025	0.004	0.025	0.0007	0.0007

Table C.4: Simulation environment parameters

Sim Id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Number of Active Nodes	60	61, 120, 297	45, 87, 216	31, 61, 150	233, 177, 127	5, 10	1 to 10	2,4, 8,16	100 to 800	200, 400
Number of Sources	1	1	1	1	1	5, 10	1	2,4, 8,16	1	20, 40
Source to Sink Distance (m)	777	777	535	365	550	Random up to 410	80, 160, ..., 800	Random up to 380	500	Random up to 535
Payload Size (bits)	4800	4800	4800	4800	4800	400	480	400	400	400
Data Rate (pkts/sec)	Single Packet	Single Packet	Single Packet	Single Packet	Single Packet	0.02 to 0.22 per node	0.1	0.02 to 0.2 per node	0.001	0.02 to 0.1
Bandwidth (kbps)	10	10	10	10	10	10	10	10	10	10
Communication frequency (kHz)	25	25	25	25	25	25	25	25	25	25
Spherical Spreading Constant	2	2	2	2	2	2	2	2	2	2
Simulation Time (s)	1000	1000	1000	1000	1000	1000	1000	3600	5000	1000

Table C.5: X-PACCA algorithm parameters

Sim Id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Slot length (s)	0.01, 0.1, 0.5, 1	0.1	0.1	0.1	0.1	0.005	0.005	0.005	0.005	0.005
Sink/Source Starting Window Size (W_s)	3	3	3	3	3	3, 30	3	3	3	30
Maximum backoff stage (m)	5	5	5	5	5	5	5	5	5	5
Maximum number of retransmissions (maxAttempts)	5	5	5	5	5	5	5	5	5	5
Ack waiting time (Wfa_{thresh} in s)	20	20	20	20	20	20	20	20	20	20
Ignore timeout (D in s)	50	50	50	50	50	50	50	50	50	50
Aqua-Sim Receive Threshold (W)	8.7e-8	8.7e-8	8.7e-8	8.7e-8	8.7e-8	8.7e-8	8.7e-8	8.7e-8	8.7e-8	8.7e-8
Backoff Constant (k)	0 to 0.08	0 to 0.08	0 to 0.08	0 to 0.4	0 to 0.05	1	1	0.1	0.001	0.001

Table C.6: Simulation confidence parameters

Confidence level (g)	99%
Confidence interval (Δ)	0.1

It should be noted that all simulations except routing comparison simulations (simulation id = 9) are repeated until confident stopping rule is satisfied with parameters given in table C.6. In routing comparison simulations, each data point is a result of 20 simulations, which means an average of 1000 packets.