Année: 2021 **Thèse N° : 218/ST2I**

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

École Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Centre d'Études Doctorales en Sciences des Technologies de l'Information et de l'Ingénieur

THÈSE DE DOCTORAT

PERFORMANCE AND COMMUNICATION QUALITY IMPROVEMENT OF UNDERWATER INTERNET OF THINGS

Présentée par

Hajar BENNOURI

Le 15/06/2021

Formation doctorale : Informatique
Structure de recherche : Smart Systems Laboratory (SSL)

JURY

Professeur Mohamed Dafir ECH-CHERIF EL KETTANI	**Président**
PES,ENSIAS, Université Mohammed Vde Rabat	
Professeur Amine BERQIA	**Directeur de thèse**
PES,ENSIAS, Université Mohammed Vde Rabat	
Professeur Hamadou SALIAH-HASSANE	**Rapporteur**
Professeur titulaire, Université TELUQ, Montréal, Canada	
Professeur Khalid ZINE-DINE	**Rapporteur**
PES, FSR, Université Mohammed V de Rabat	
Professeur Mohamed HANINI	**Rapporteur**
PH, FST, Université Hassan 1er de Settat	
Professeur Hassan BERBIA	**Examinateur**
PES, ENSIAS, Université Mohammed V de Rabat	

DEDICATION

To my dear parents,

No act or expression can express my feelings towards you.

To my brothers and sisters,

As a witness to the love and affection that bind us together.

To those who have never stopped encouraging and advising me.

To those who have never been stingy, neither with their time nor with their knowledge, to satisfy my questions.

To those benevolent educators,

I dedicate the fruit of my research career.

To my friends,

For the unforgettable time we spent together.

To all who love me,

I dedicate this work ...

Acknowledgements

Undertaking this PhD has been a truly life experience for me and it would not have been possible without the help and support of many people.

An adventure, a beautiful adventure that comes to an end and by this I would like to thank the jury members for agreeing to evaluate my work.

I would like to express my warm thanks to my supervisor Professor *Amine BERQIA* for giving me the choice and the freedom to try things since the first day of this thesis, his availability, his commitment and his advice.

I would like to thank all the jury members for accepting to judge this work.

My thanks go for Professor *Anis Yazidi* from OsloMet – Oslo Metropolitan University in Oslo for the collaboration which has enriched my work, I thank him for his patience, his availability and above all his sound advice, which contributed to my reflection. I also thanks all his team, this work would not have proceeded without their welcomes, collaboration, support and advice.

I also address my most distinguished thanks to our dear Professor *Mohamed Essaaidi* as well as the entire administrative and teaching staff of ENSIAS, with whom I have spent moments of work but also moments of life.

I address my sincere thanks to all the professors, speakers and all the people who by their words, their writings, their advice and their criticisms guided my reflections and agreed to meet me and answer my questions during my research.

I would like to thank in particular *Frank Domoney*, my confidences *Sofia* and *Amina* for their numerous proofreading and corrections of my disastrous English and their personal and professional supports. They have always responded present, even in the most desperate cases. Thank you.

I will not forget my parents, brothers and sisters, no thanks can match their endless encouragement, as well as their unconditional help, understanding and support.

Finally, May all those who crossed my way and contributed directly or indirectly to the success of that part of my life find the expression of my warmest thanks..

" When wireless is perfectly applied the whole earth will be converted
into a huge brain, which in fact it is, all things being particles of
a real and rhythmic whole. We shall be able to communicate
with one another instantly, irrespective of distance."

Sir Nikola Tesla
(1856-1943)

Abstract

Underwater Wireless Sensor Networks (UWSNs) consist of a number of sensor nodes and intelligent vehicles interconnected to form what is known as Internet of Underwater Things (IoUT). Research in this area has led to the deployment of several autonomous underwater object applications that are used to explore, monitor or collect data from large, unexplored areas of water, for example, the effects of climate change on coral reefs. It would also help detecting seismic activity and monitor pipelines. Despite the importance and utility of these applications, submarine sensor nodes are more expensive and less deployed unlike terrestrial sensor networks. Electromagnetic waves are unusable for this purpose (except at very low frequencies and at low flow) and the optical wave only carries a few tens of meters, even in the blue-green range whose propagation characteristics are especially favorable. Underwater wireless communications are established by transmission of acoustic waves which is the only physical medium for the transmission of viable wireless information in the marine environment. Underwater acoustic communication is known for its limited bandwidth, long propagation time and signal fading. In addition to its poor performance in terms of packet loss during transmission. In this aquatic environment, the traditional transport protocols TCP (Transmission Control Protocol) are very sensitive to packet loss, which makes their integration into submarine networks too difficult because the default parameters used in terrestrial communication are not suitable for this type of environment. However, there is a lack of research studies on the behavior of these TCP protocols in the underwater environment. Therefore, it is necessary to study the behavior of these protocols in order to adjust their parameters to achieve more efficient operations in this aquatic environment. The submarine sensor nodes are fitted with a limited battery which cannot be replaced or recharged. High transmission power and lengthy of data packet transmission consumes a large amount of energy due to the difficult type of communication in this environment. The challenge of energy conservation for underwater sensor networks is to develop efficient routing methods and communication techniques to reduce the rate of energy consumption. In this thesis, we aim to study and evaluate the performance of traditional TCP protocols in UWSNs in order to improve their performance in this environment. We are therefore particularly interested in studying the effect of controlling the maximum window and adjusting the value of round-trip time (RTT) of TCP NewReno under different scenarios in order to improve the communication of our underwater network. The results of these studies allowed us to present a new transport protocol well adapted to this underwater environment called U-NewReno. To present the results of our work, simulation results show the efficiency of our proposed protocol in terms of packet delivery gain and packet delivery retransmission rate. On the other hand, underwater environments are subject to varying conditions which might degrade the quality of communications. The submarine sensor nodes must be able to self-configure and adapt to the harshest conditions of this environment. In this fact, we propose an adaptive control mechanism to control the mobility of thermocline sensors to improve the link stability in underwater networks using the theory of Learning Automata (LA). The reported results show the ability of our algorithm to find the optimal sensor positions for better network monitoring. In addition, we also propose an optimization approach to improve the quality of monitoring of UWSNs by resorting to the machine learning concept of diversity.We devise a multi-agent optimization procedure which is able to both reduce the redundancy among the sensor readings and maximize the diversity in a distributed and adaptive manner. The mobile sensor positions are adjusted iteratively using a gradient type of updates Simulations results based on realistic environment conditions are performed, which conquer with the theoretical results, and illustrate the performance of our approach. Furthermore, we are interested to study the impact of using fuzzy logic approach in a routing protocol to evaluate the energy performance of an UWSN. We implement the FLOVP (Fuzzy Logic Optimized Vector Protocol) routing protocol which is an improved version of the VBF (Vector Based Forwarder) routing protocol. Simulation results confirm the energy efficiency of the FLOVP Routing Protocol compared to the original VBF Routing Protocol.

Keywords: Internet of Underwater of Tings, IoUT, Underwater Wireless Sensor Networks (UWSNs), TCP, TCP NewReno,U-NewReno, Automata Learning (AL),Redundancy, Diversity, Quality of Monitoring, Routing Protocol, Energy-efficient, Fuzzy Logic

Résumé

L'Internet des Objet sous-marins (IoUT) correspond à des réseaux de capteurs sans fil sous-marins composés d'un certain nombre de noeuds de capteurs,appareils de détection, robots et de véhicules intelligents inter-connectés avec les bases terrestres pour renforcer le réseau de l'internet des Objets (IoT). La recherche dans ce domaine a conduit au déploiement de plusieurs applications d'objets autonomes sous-marins.Ces objets ont plusieurs applications, telles que:l'exploration, la surveillance ou la collection des données sur des vastes zones maritimes inexplorées. Parmi ces applications, nous pouvons noter l'analyse des effets du changement climatique sur les récifs coralliens. Une autre application de ce domaine est la détection de l'activité sismique et la surveillance des pipelines. Malgré l'importance et l'utilité de ces applications, les noeuds des capteurs sous-marins demeurent très chers et moins déployés contrairement aux réseaux des capteurs terrestres. Dans l'environnement marin, les ondes électromagnétiques sont inutilisables (sauf à très basses fréquences et à faible débit) et les ondes optiques ne fonctionnent que sur quelques dizaines de mètres, même dans le domaine bleu-vert dont les caractéristiques de propagation sont particulièrement favorables. Les communications sans fil sous-marines sont établies par transmission d'ondes acoustiques qui constituent le seul support physique fiable pour la transmission d'informations sans fil dans l'environnement marin. La communication acoustique sous-marine est connue par sa bande passante limitée, le long temps de propagation et l'évanouissement du signal, en plus de ses faibles performances en termes de perte de paquets lors de la transmission. Dans le milieu aquatique, les protocoles traditionnels du transport TCP (Transmission Control Protocol) sont très sensibles à la perte de paquets, ce qui rend leur intégration dans les réseaux sous-marins très difficile. En effet, ceci est dû aux paramètres de communication terrestre utilisés par défaut dans le milieu marin alors qu'ils ne sont pas adaptés à ce type d'environnement. Cependant, il n'y a pas suffisamment d'études de recherches sur le comportement du TCP dans l'environnement sous-marin. Il est donc nécessaire d'étudier le comportement de ces protocoles afin d'ajuster leurs paramètres pour parvenir à un fonctionnement plus efficace dans ce milieu aquatique. D'autre part, les noeuds de capteurs sous-marins sont équipés d'une batterie limitée qui ne peut être ni remplacée ni rechargée. La puissance de transmission élevée et la longue durée de la transmission de paquets de données consomment une quantité d'énergie importante vu la difficulté du type de communication dans cet environnement. Résoudre l'enjeu de la conservation de l'énergie pour les réseaux de capteurs sous-marins consiste à développer des méthodes de routage et des techniques de communication efficaces qui aident à réduire le taux de consommation d'énergie. Dans cette thèse, nous visons à étudier et évaluer les performances des protocoles traditionnels TCP dans les UWSNs afin d'améliorer leurs performances dans l'environnement marin. Nous sommes donc particulièrement intéressés par l'étude de l'effet du contrôle de la fenêtre maximale et ajuster la valeur du temps de parcours RTT de TCP NewReno dans le réseau sous-marin sous différents scénarios. Les résultats de ces études ont permis de présenter un nouveau protocole de transport bien adapté à cet environnement sous-marin appelé U-NewReno. Dans le cadre de nos travaux réalisés, les simulations effectuées montrent l'efficacité de notre protocole proposé en terme de gain de livraison de paquets et de taux de retransmission de livraison de paquets. D'un autre côté, les environnements sous-marins sont soumis à des conditions variables qui pourraient dégrader la qualité des communications. Les noeuds de capteurs sous-marins doivent être capables de s'auto-configurer et de s'adapter aux plus dures conditions de cet environnement. De ce fait, nous introduisons un mécanisme de contrôle adaptatif pour contrôler la mobilité des capteurs thermocline afin d'améliorer la stabilité de la liaison dans les réseaux sous-marins, en utilisant la théorie de l'apprentissage Automata (LA). Les résultats rapportés montrent la capacité de notre algorithme à trouver les positions optimales des capteurs pour une meilleure surveillance du réseau. Par ailleurs, nous proposons également une approche d'optimisation pour améliorer la qualité de surveillance des UWSN en recourant au concept d'apprentissage automatique de la diversité. Nous concevons une procédure d'optimisation multi-agents capable à la fois de réduire la redondance entre les lectures des capteurs et de maximiser la diversité de manière distribuée et adaptative. Les positions des capteurs mobiles sont ajustées de manière itérative à l'aide d'un type de gradient de mises à jour. Des simulations basées sur des conditions d'environnement réalistes ont été réalisées, et leurs résultats coïncident avec les résultats théoriques, et illustrent la performance de notre approche. En outre, nous nous sommes intéressés à étudier l'impact de l'utilisation de l'approche de logique floue dans un protocole de routage pour évaluer la performance énergétique d'un réseau de capteurs sans fil sous-marin. Nous implémentons le protocole de routage FLOVP (Fuzzy Logic Optimized Vector Protocol) qui est une version améliorée du protocole de routage VBF (Vector Based Forwarder). Les résultats des simulations affirment l'efficacité énergétique du Protocol FLOVP par rapport au Protocole original de routage VBF.

Mots-clés: Internet des Objet Sous-marins (IoUT), Réseaux de Capteurs Sous-marins sans fil (UWSN), TCP, TCP NewReno, U-NewReno, redondance de l?information, la diversité, L'apprentissage des Automates (LA), Protocole de Routage, Efficacité énergétique, Logique Floue

Contents

Page

List of Figures

List of Tables

Acronyms and Symbols

ACK	-	**Ack**nowledgement
AODV	-	**A**d hoc **O**n- **D**emand **D**istance**V**ector
AUVs	-	**A**utonomousit **U**nderwater **V**ehicule**s**
BER	-	**B**it **E**rror **R**ate
BS	-	**B**ase **S**tation
CPU	-	**C**entral **P**rocessing**U**nit
CWnd	-	**C**ongestion **W**indow
DBR	-	**D**epth **B**ased **R**outing
DNA	-	**D**eoxyribonucleic **A**cid
E2ED	-	**E**nd **2** **E**nd **D**elay
FIS	-	**F**uzzy **I**nference **S**ystems
FL	-	**F**uzzy **L**ogic
FLOVP	-	**F**uzzy **L**ogic **O**ptimized **P**rotocol
FSK	-	**F**requency **S**hift **K**Keying modulation
FTP	-	**F**ile **T**ransport **P**rotocol
GPRS	-	**G**eneral **P**acket **R**adio **S**ervices
GSM	-	**G**lobal **S**ystem for **M**obile Communications
IEEE	-	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IoT	-	**I**nternet of **T**hings
IoUT	-	**I**nternet of **U**nderwater **T**hings
IW	-	Initial**Wind**ow
LA	-	**L**earning **A**utomata
LM	-	**L**earning **M**echanism
LMA	-	**L**imited **M**obile **A**gent
MANETs	-	**M**obile **A**d-hoc **NET**work **s**
METS	-	**Met**han **S**ensor
PSK	-	**P**hase **S**hift **K**Keying modulation
pH	-	**P**otentiel**h**ydrogen
PVC	-	**P**permanent **V**irtual **C**ircuit
QoS	-	**Q**uality **of** **S**ervice
RAT	-	Radio Access Technology
RIP	-	**R**outing **I**nformation **P**rotocol
RL	-	Reinforcement Learning
RTT	-	**R**ound **T**rip **T**ime

RTO	-	**R**etransmission **T**ime **O**ut
SMS	-	**S**hort **M**essage **S**ervice
SMSS	-	**M**aximum **S**egment **S**ize of the **S**ender
SRP	-	**S**uccessful **R**eceived **P**acket
SPL	-	**S**tochastic**P**oint**L**ocation
TCP	-	**T**ransmission **C**ontrol **P**rotocol
TSP	-	**T**otal **S**ent **P**acket
TWSNs	-	**T**errestrial **W**ireless **S**ensors **N**etworks
UoT	-	**U**nderwater of **T**hings
UWSNs	-	**U**nderwater **W**ireless **S**ensors **N**etworks
VBF	-	**V**ector **B**ased **F**orwarders
WWW	-	**W**orld **W**ide **W**eb
VANETs	-	**V**ehicular **A**d-hoc **NET**work**s**
WiFi	-	**W**ireless **F**idelity
WLAN	-	Wireless Local Area Network

Introduction

General context

Nowdays, the digital transformation evolution that the Smart city is experiencing with extremely innovative Internet of Things (IoT) solutions, an evolution that has encompassed the underwater environment as Smart Water to form what is called Underwater Internet of Things (UIoT) which is quite simply a network of smart wireless sensors and smart devices configured widely used in ocean research to provide actionable operational information such as underwater environment performance, condition and diagnostic information.

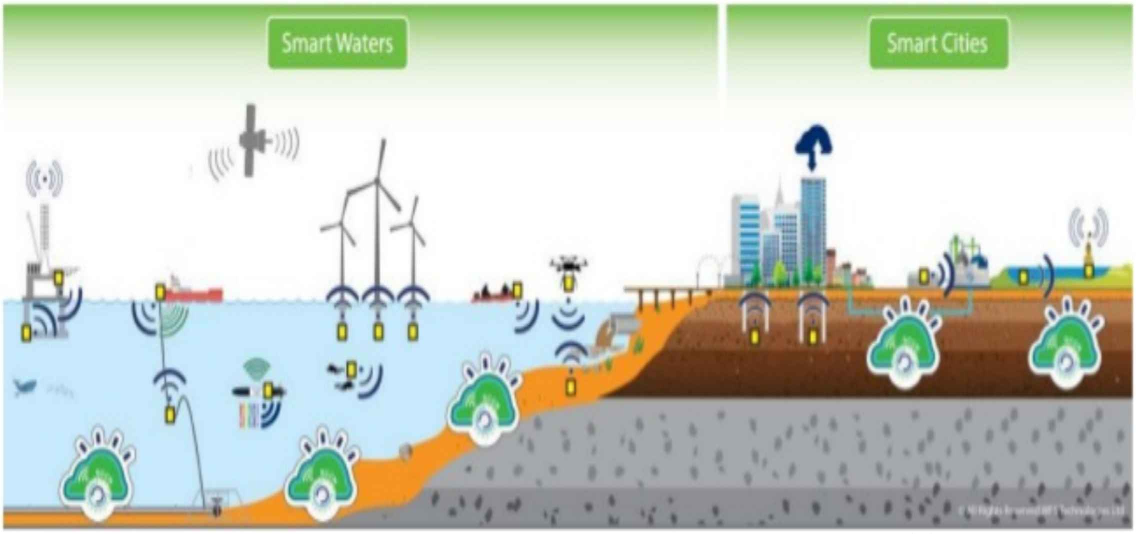


Figure 1 – *Extending data of Smart Cities with Smart water data* (1)

The Internet of Things brings many benefits to sensor platforms, both individually and collectively. With the current capabilities of embedded processors and single-board computers, Underwater of Things on the other hand is formed by oceans monitoring devices which became more intelligent. Many other applications are possible in search and rescue, environmental monitoring, surveillance operations, detection of oil spills, fishing or underwater archaeology. In addition, autonomous submarines are sought after by armed forces around the world.

The data come from a variety of sources: stationary sensors on the ribs, sensors installed on ships or boats of the merchant navy, on buoys, drift sensors deployed over long periods and very large areas ... More recently, a wide variety of underwater drones have emerged, some of these drift sensors very sophisticated and designed specifically to study the environment.

Analytic distributed on remote platforms at the periphery of the network improves operations at the sensor. UoT nodes can measure sensor performance, detect and correct calibration errors or drifts, anticipate faults, adjust changes in the environment in which they are deployed. The IoT and UoT are generally based on the connectivity between different communication technologies and may include media to acquire real-time or asynchronous data, all managed autonomously.

All information received from the sensors provides a representation of the oceans over time. The more data available, the more analytical tools can be used to understand the correlations between different variables and thus detect fine environmental trends that highlight weak signals about ocean changes.These data provide a basis for observing changes over time, for developing protection programs and evaluating their effectiveness.

Significant efforts are being made around the world to address oceans-related environmental issues. By its size, the sea universe generates a large volume of physical, chemical and biological data ranging from microscopic to macroscopic. There is structured data that comes from sensor networks and others more complex that can be transmitted by satellite, for example: via images and spectral data. Sounds can track the activity of marine mammals or measure underwater noise from boats, near-shore or offshore operations (such as gas and oil extraction) and renewable marine energy equipment that goes through waves or tides.

Motivation & Summary of the main goals

Scientists have complete information about what is happening on the moon, but they do not have enough knowledge about what is happening in the ocean. In recent years, the scientific community has placed more emphasis to explore the seabed. To do this, several sophisticated submarine robots equipped with multiple sensors and hydrophones are set up to discover the mystery of this environment.

In the future, aquaculture and aquaponics could play a very important role to meet the increasing demand of a growing world population and meet future demand for food. Underwater Wireless sensor networks (UWSNs) are increasingly used to measure various water parameters in order to monitor the evolution of an aquaculture. In addition,the field of aquaculture relies mainly on submarine sensors for the monitoring of their aquatic fish farms. That is why, this field of application is one of the main reasons that led us to conduct this study. In fact, our country is bounded by two maritime facades: on the north by the Mediterranean Sea (512Km) and on the west by the Atlantic Ocean (2.934Km). These two maritime coasts give a wide water surface in addition of more than twenty-five streams of rivers. Adapting a system network in this environment requires several studies for reliable communication. To conduct these studies and permit several applications, the introduction of several submarine sensor networks all around the water environment is required.

The general framework of our research is the transmission of data in the underwater wireless networks. We are particularly interested in studying the performance of standard protocols of the terrestrial transport network and observing their behavior in this environment in order to be inspired to propose a new protocol adapted to the marine environment with good performances. On the other hand, we were also interested in proposing new methods to enhance communication and quality of monitoring based on automated learning and multi-objetive function by adjusting the positions of sensors. Another axis of this research is to describe the performance of a routing protocol in terms of energy conservation using fuzzy logic. By this thesis, our aims is to offer:

- A clear comparative study of terrestrial network Transport Control Protocols in the marine environment while taking into account routing protocols;
- Adjustment of a terrestrial transport protocol parameters in order to enhance its performances in the submarine environment;
- A presentation of a new transport protocol well suited to this underwater environment;
- An approach based on automata-learning to improve marine communication by finding the best positions of sensors;
- A gradient based approach to iteratively adjust the positions for sensor nodes.
- An energy performance of a routing protocol based on fuzzy logic approach in UWSNs;

In the next section, we present approaches used to solve the main problems of our research, and we detail the methodology used to achieve our objectives mentioned above.

Thesis outline and structural overview

This thesis report is organized in eight chapters. Below we give a brief discussion of each chapter:

- The first Chapter presents a state of the art of Wireless Sensor Networks (WSNs) and the Underwater Wireless Sensor Networks (UWSNs).
- The second Chapter, outlines the communication's challenges that Underwater Wireless Sensor Networks still face.
- in the third Chapter, we firstly introduce a studies of different standard TCP mechanisms in UWSNs, in order to come out with better parameters of the functioning of one of these protocols in this marine environment. Secondly, after having de-

termined the best TCP protocol which will be the main axis of our study. We start our first improvement of this protocol to adapt it more with this submarine environment,then second improvement of this TCP protocol by studying another parameter, always with the aim of better improving the performance of our protocol in Underwater environment. And finally, we fully apply the previously exposed improvements to present a new underwater TCP protocol well suited to the underwater environment based on our previous results.

- In the Chapter four, to improve communication in this aquatic environment, application of adaptive learning strategies to modify the depth of deeply anchored Limited Mobility Agents (LMAs) using Pursuit Learning Solution.
- In chapter five we consider a multi-objetive function to improve the quality of monitoring information which aspires to minimize the covariance between all sensors,which mean reducing redundancy and in the same time maximize the diversity. We used in this case a gradient based approach to iteratively adjust the positions for sensor nodes.
- Chapter six leads us to the study of the energy efficient part of Underwater Wireless Sensor Networks when applying the theory of fuzzy logic in a routing protocol.

In the last part of the report, we summarize the work of this thesis, and outline some future research directions.

Chapter 1

An overview on Internet of Things and Internet Of Underwater Things

1.1 Introduction

The Internet-of-Things is an emerging revolution in the Information Communication Technology (ICT) sector under which there is shift from:an Internet used for ″ interconnecting end-user devices″ to an Internet used for ″ interconnecting physical objects that communicate with each other and/or with humans ″. Extending the concept of Internet of things (IoT) to the marine environment give us the Internet of Underwater things (IoUT). Smart Water things are deployed under the water form the Underwater Wireless Sensor Networks(UWSNs) that transmit collected data to networks above surface in real time.

This chapter explores the paradigm of Wireless Sensor Networks (WSNs) and Underwater Wireless Sensor Networks(UWSNs) which are the basic components of Internet of Things(IoT) and Internet of Underwater Things(IoUT).

1.2 Internet of Things: Wireless Sensor Networks

1.2.1 Definition

The Internet of things(IoT) is a new IT paradigm focused on the collective effort of a very large collection of self-organizing sensors. Based on the Wireless Sensor Network (WSN) which is a network consisting of a large number of sensor nodes with wireless communication between these nodes. The main entities that make up the sensor network are:

- The sensor nodes that make up the network.
- The base station that communicates with the user through the Internet or via satellite communication.
- The phenomenon is the topic of measures of concern to the user.

It is interesting, from this general description, to highlight the key features that define this kind of network.

1.2.2 Characteristics of Wireless Sensor Networks

The Wireless Sensor Networks (WSNs) are non-wired networks consisting of physical phenomenon detection units (temperature, humidity, radioactivity, etc.) deployed in a geographical area of interest. The purpose of WSNs is to monitor their deployment environments to report the detection of physical phenomena (forest fires, intrusions, radioactive leaks, etc.). WSNs remain one of the most active fields of study. Indeed, the advancement of micro-electromechanical systems and the availability of low-cost communication and computer equipment are all factors that allowed the passage of science fiction paradigms and ideas and commercialized products to be achieved.(9, 10).

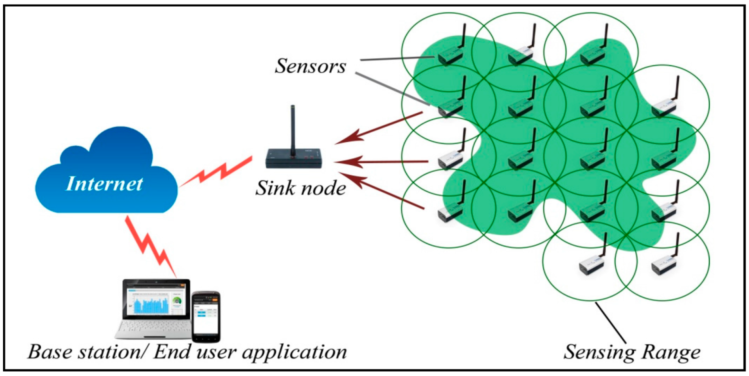


Figure 1.1 – *WSNs architecture*

The critical issue of WSNs is energy usage, which is highly influenced by communication between nodes(11). In fact, in order to reduce the energy consumed by the network, the different resources of the network must be optimally controlled in order to make the sensors operate at minimal cost.
To establish a WSN it is needed to have:

- **Mobility** (A dynamic topology): Node mobility is simply a very unique function of ad hoc networks. This versatility is a vital component of the functioning of the network. In an ad hoc network, network topology can change rapidly, unexpectedly and unpredictably, in addition, conventional network routing methods based on pre-established routes cannot work properly.
- **Network nodes**: In a traditional network, there is a strong distinction between terminal nodes (stations, hosts) that serve applications and internal nodes (routers, for example) of the network responsible for data routing. This distinction does not occur in ad hoc networks since all nodes can be used to provide routing functions.
- **Wireless links** (Limited bandwidth): Wireless networking technologies are important for the establishment of an ad hoc network. Despite very substantial improvement, their efficiency remains and will remain below that of wire-line technologies. Bandwidth is less important, while routing and mobility management produce more control and signaling flows than the wired network architecture.

1.2.3 WSNs architecture

In general, the sensor network consists of numerous nodes distributed in a region. These nodes are connected to one or more gateways which make it possible to link to other networks (Internet, satellite, etc.) and data recovery (12), as shown in figure 1.1.

Various prototypes and commercialized products have been developed in the literature to accelerate research on WSNs and to enable their incorporation into operational application scenarios (13). Although the different sensor implementations vary in their physical architecture, they share the following modules:

- **Source of energy**: These are typically lithium batteries. The sensors are most frequently deployed in hard-to - reach areas; these batteries are irreplaceable and the life of the sensor is limited to that of these batteries, although the WSNs literature includes numerous prototypes of renewable energy sensors. (14).
- **Processor**: The miniaturization effort has enabled the production of processors adapted to the size and cost constraints associated with wireless sensors. However, it should be noted that the related computing capacity remains limited (for example, only 8 MHz of frequency and 128 kbytes of programmable memory for a waspmote sensor based on the ATmega1281 microcontroller).
- **Transmitter / receiver**: This module enables network sensors to communicate with each other. Two families of transceivers are available:
  - Optical transceivers suffer from a significant handicap due to the need for visibility between sensors. Communication may be primarily influenced by terrain elevation issues and physical obstacles in the deployment area.
  - Radio Frequency transceivers make it possible to correct the various problems associated with optical transceivers but require modulation and demodulation equipment, bandwidth filtering and multiplexing, which has the effect of raising the associated costs. Several standards have been suggested, including: IEEE 802.15.4 (15), Bluetooth (16), etc.
- **Location equipment**: Without GPS equipment that respects the limitations associated with the WSNs as well as indoor deployment scenarios, various proposals for location algorithms based on range parameters (RSS, AoA, etc.).) have been formulated (17, 18, 19).
- **Unit of mobility**: Mobility can be autonomous or assisted (sensors attached to individuals or vehicles) and may increase the coverage of the area of interest and prevent the rapid depletion of energy sensors in the vicinity of the base station.
- **Capture Unit**: The primary purpose of the capture unit is the capture or measurement of physical data from the target object. It consists of two sub-units: the receiver (recognizing the physical quantity to be sensed) and the transducer (converting the receiver signal into an electrical signal). The sensor transmits analog signals, based on the observed phenomenon, to the Analog / Digital Converter. The latter translates these signals into digital data and transmits them to the processing unit.
- **Storage unit** (Memory): This includes the memory of the program and the mem-

ory of the data. The scale of this memory is mostly limited primarily by economic considerations and continues to improve over the years. (20).

1.2.4 Constraints and specification of WSNs

Considered to be an evolution of the Ad-Hoc networks, WSNs do not need network infrastructures and must be adequate for data routing in a multi-hop mode. These networks are restricted by several constraints and are therefore prone to vulnerabilities due to damage caused by their implementation environments. The following drawbacks and specifies are contrasted to conventional Ad-Hoc networks. (12, 21, 22)

- **Limited energy**: The sensors are fitted with limited energy batteries (several days to a few years). In addition, WSNs are most commonly deployed in areas that are difficult to reach or hostile to humans. In general, after deployment, it is difficult to interfere on WSNs to make adjustments to the battery. It is therefore necessary to rationalize energy usage within the WSNS in order to extend their lifespan. Communications are the most expensive energy practices according to measurements. It is therefore necessary to restrict the number and to program standby periods in order to save energy.

- **Scaling**: Sensors are deployed in a range from a few units to several tens of thousands. In addition, the network can be improved post-deployment by installing new sensors to replace faulty units, extending the coverage of WSNs to new geographic areas, or expanding the functionality of WSNs by integrating sensors. To detect activities of a different type than those envisaged during the initial deployment.

- **Fault tolerance**: There are different modes of deployment for WSNs. In fact, the sensors can be manually deposited as well as dispersed by a helicopter or projected by missiles. For certain deployment modes, a portion (20 % to 30%)of the sensors would not survive the deployment. In addition, the WSNs deployment environment will cause damage to the sensors. Humidity, heat, etc. are all climatic factors which accelerate the deterioration of the sensors. The WSN must therefore be fault-tolerant and self-configured to fix sensor failures.

1.2.5 Application domains of terrestrial sensor networks for IoT

WSNs can have a lot of applications. Among them we mention:

- **Military applications**, a network of sensors installed in a strategic or difficult-to-access sector makes it possible, for example, to track all movements (allies or enemies) or to assess the battlefield before sending reinforcements. (23), (24).

- **Monitoring applications**, the use of sensor networks in the field of security will dramatically reduce the financial expense for protecting places and people. The incorporation of sensors into large structures, such as bridges or buildings, can help identify fractures and changes in the structure following the earthquake or the aging of the structure. The implementation of a network of motion sensors may provide an alarm system that will be used to detect intrusions into the surveillance area.

- **Medical applications**, in the medical field, there are already multi-sensor capsules that can be swallowed in order to relay images from the inside of the human body without having to resort to surgery.(25) (26).
- **Environmental applications**,temperature sensors may be distributed from aircraft to identify potential environmental issues in the region protected by the sensors in order to avoid possible fire, flood, volcano or tsunami from occurring in time.(27) (7) (28).
- **Commercial applications**,sensor nodes can be used to enhance storage and distribution processes. Thus, the network can be used to know the location, state and path of the commodity. The customer waiting for the goods can then have an online delivery notice and know the location of the goods he has ordered. (29).
- **Home Automation Systems**, such as vacuum cleaners, microwave ovens, refrigerators, video recorders can be combined with technological advancements (30). These on-board sensors may communicate with each other and with an external network over the Internet to allow the user to monitor home devices locally or remotely. The implementation of motion and temperature sensors in so-called smart homes automates a variety of domestic operations, such as: the light is switched off and the music is switched off when the room is empty, the air conditioning and heating are adjusted to various measurement points, and the anti-intrusion alarm is activated when the intruder tries to reach the house.

1.3 Internet of Underwater of Things: Underwater Wireless Sensors Networks

In this section, we will present an overview of Internet of Underwater of Things system based on Underwater Wireless Sensor Networks (UWSNs) and explain the characteristics, challenges, applications and their different communication architectures.

1.3.1 Definition

Over the last decade, a new form of wireless network has aroused considerable interest in the scientific community, the UWSNs which is the basis to build the internet of underwater Things (IoUT). this networks are a special form of wireless sensor network for specialized applications in the marine environment. They are a new area of research that has been developed to provide effective underwater communication in civil and military applications. The acoustic wave is a medium for transmitting information in a network of wireless underwater sensors, which are confronted with many problems related to the requirements of each applications. environment.

1.3.2 Historic

The history of underwater vehicles is not recent. The evolution of research in this area can be quoted in chronological order. The first attempts at practical implementation of this concept have been recorded since the time of Alexander the Great, who, according to Aristotle, had built a primitive submersible for reconnaissance missions in 322 BC. A similar machine was built about 200 BC in China. In the modern period, the Englishman William Bourne took the first step towards the idea of an underwater vehicle in 1578, constructing a waterproof model that could be supplied with oxygen. But his proposals did not go beyond the stage of design.

Between 1620 and 1624, Cornelis Van Drebbel, the Dutchman, would have sailed a few meters under the waters of the Thames. In 1664, he finally offered the first submarine vehicle to advance, using 12 rowers fitted with special oars. Having an ovoid shape, it could be actuated to accomplish vertical motions. It was also tested experimentally.

In 1776, the first American underwater vehicle 'Turtle' was presented by David Bushnell and his brother (see Figure 1.2). Built of steel, it could move independently. It is the first to be fitted, on the one hand, with propellers for its propulsion and, on the other hand, with a valve for submersion and an ascent to the surface. It was the first autonomous submersible to participate in naval combat.(2).

Since that time,underwater vehicles have grown considerably from the point of view of their power source. Study aimed at removing human propulsion and harnessing other forms of energy. The diver of the French Navy, launched in 1863, was the first submarine to be fitted with a compressed air engine and a pressure tank PSI of 23,180 (12 bar)(31).

In 1888, Le Gymnote was the first French navy submersible to be fitted with a 55 HP electric motor powered by 564 accumulators. Smooth and tapered in appearance, the Gymnote was 18 meters long. Its circumference, the widest part of it, was barely 2 m. While it was intended for military use, the Gymnote was subsequently assigned to experimental missions.

The Spanish Isaac Peral, designed the first fully operational electrically operated submarine. Checked at sea on September 8, 1888, it had two torpedoes, new air systems, a propeller and control surfaces, thereby defeating future submarines. Given the short life

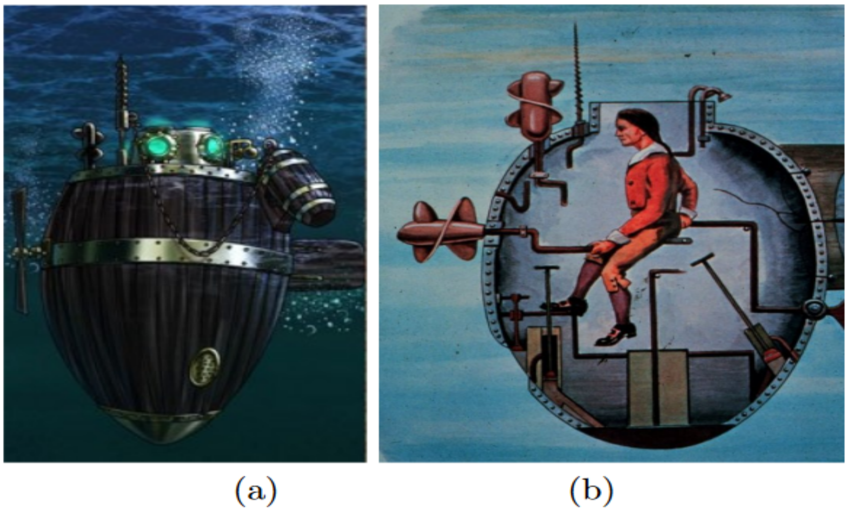


Figure 1.2 – *Bushnell submarine* (2)

of its batteries, it was able to reach 10 underwater nodes. With the turn of the century, the production and use of submarines has drastically changed. With the advent of many inventions and developments, such as the generalization of the periscope to the majority of submarines of the time and the predominance of diesel-electric propulsion in terms of energy, the rate of adoption of desk pads by many ships has increased dramatically. Nuclear power started to replace diesel-electric propulsion in the 1950s. The first autonomous underwater vehicles were built between 1960 and 1970 years (2, 31, 32, 33) with the following characteristics:

- The SPURV (Self-Propelled Underwater Research Vehicle, USA, 1977): weighing 480 kg, it could reach a speed of 2.2 m / s for 5 hours in a row with a maximum immersion capacity of 3000 m. It was used to make conductivity and temperature measurements applied to wave modeling (2) (see Figure 1.3 (a)).
- The killer whale (France, 1976): weighing 3 tons, it could reach a speed of 12 knots (6.17 m/s) for 7 hours in a row with a maximum immersion capacity of 6000 m and maintain an acoustic connection with the surface (33) (see Figure 1.3 (b)).

1.3.3 Different UWSN's architectures

The topology of a network is always a subject of research open to the scientific community. In the following, we present in particular two-dimensional and three-dimensional architectures, and also the mobile topology which uses autonomous underwater vehicles (AUV) which can improve the capacity of underwater sensor networks.

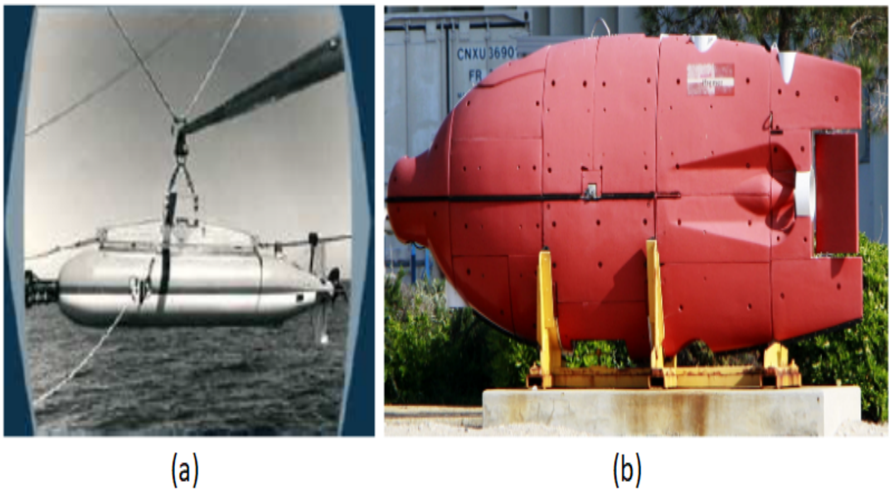


Figure 1.3 – *The first autonomous submarines. (a) The SPURV (USA, 1977), (b) The killer whale (France, 1967)* (2)

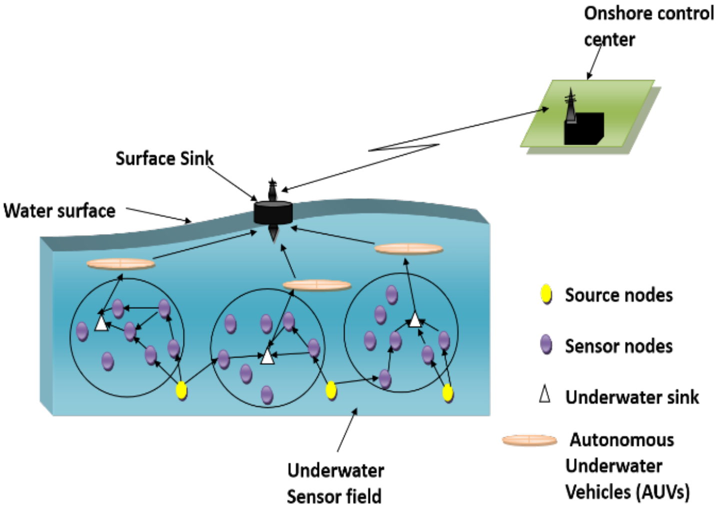


Figure 1.4 – *Example of General UWSNs architecture* (3)

Two-dimensional (2D) underwater sensor arrays

The architecture for 2D submarine sensor arrays is shown in Figure 1.5. A group of sensors are anchored to the bottom of the ocean, nodes are linked to one or more underwater wells (uw-sinks) via wireless acoustic links. Submarine sinks are devices that transmit data from the ocean floor to the surface station.

To accomplish this goal, the submarine sinks are fitted with two acoustic transceivers, the vertical and the horizontal transceivers. The horizontal transceiver is used by the underwater sink to communicate with the sensors to:
- Send commands and configuration data to sensors.

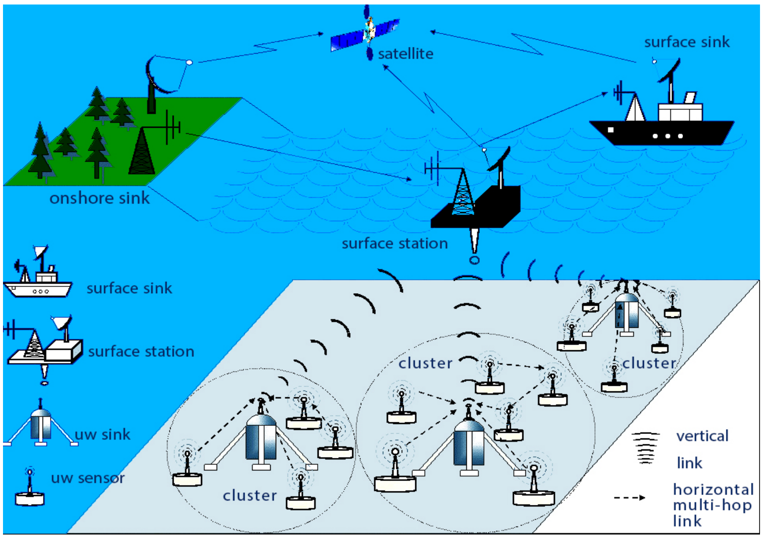


Figure 1.5 – *Two-dimensional(2D) UWSNs architecture* (4)

- Collect surveillance data.

The vertical link is used by the submerged sink to relay data to the surface station. The surface station is fitted with an acoustic transceiver capable of handling multiple communications in parallel, depending on the submarine sinks deployed.

The sensors can be connected to submarine sinks via direct links or via multi-hop paths. In the first example, each sensor transmits the collected data directly to the selected submersible sinks (34), the submersible sink may be very far from the sensor, thus, while the connection to a direct connection is the easiest way for a network of sensors, it cannot be the most energy-efficient solution. In addition, direct connections are likely to reduce network throughput due to increased acoustic interference and high transmission strength. In the case of multi-hop routes, as in the case of terrestrial sensor networks(35), the data produced by the source is transmitted by the intermediate sensors until the submersible sink is reached. The second case provides a number of benefits, such as energy efficiency and improved network capacity, but at the same time the complexity of the routing features can be improved.

Three-dimensions (3D) underwater sensor arrays

Three-dimensional underwater sensor arrays are used to detect and observe phenomena that cannot be properly observed using nodes anchored to the ocean floor so that a cooperative sampling of the 3D marine environment is required. Three-dimensional underwater sensor networks use sensors that float at various depths to observe a given phenomenon. One potential solution will be to connect each node to a surface buoy, using wires the length of which can be changed to adjust the depth of each sensor (36). However, as this approach allows for a fast and rapid deployment of the sensor network, numerous

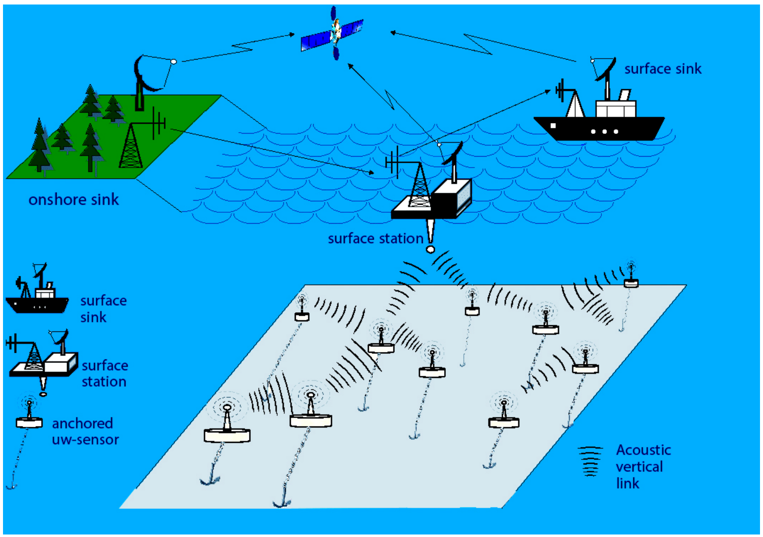


Figure 1.6 – *Three-dimensional(3D) UWSNs architecture* (4)

floating buoys can obstruct ships circulating on the surface, or they can be easily detected and disabled by enemies in military settings. In addition, floating buoys are vulnerable to weather conditions and tampering. For these reasons, a different solution is proposed, and in this architecture, shown in Figure 1.6, each sensor is anchored to the ocean floor and fitted with a floating buoy that can be inflated by a pump. The buoy brings the sensor to the surface of the water. The depth of the sensor can then be changed by varying the length of the wire that connects the sensor to the anchor by means of an electronically operated motor on the sensor. One difficulty in such an architecture is the influence of ocean currents on the system that controls the depth of the sensors.

There are several challenges to such architecture that need to be addressed in order to allow 3D tracking, including:

- **Scope of coverage:** Sensors must adjust their depths collaboratively in order to achieve 3D ocean coverage based on their ranges of detection. It is necessary to achieve a sampling at all depths of the desired phenomenon.
- **Communication coverage:** The sensors must be able to transmit information to the surface station through multi-hop paths. Network devices should then know their depths such that the network topology is permanently connected, i.e., at least one path from each sensor to the surface station still exists.

1.3.4 Different types of Underwater vehicles

Since the end of the 20th century, underwater vehicles have developed considerably from a technical point of view. We can divide them into two categories: submarines and submarine robots, and we are going to make a fast inventory of them. They are also the

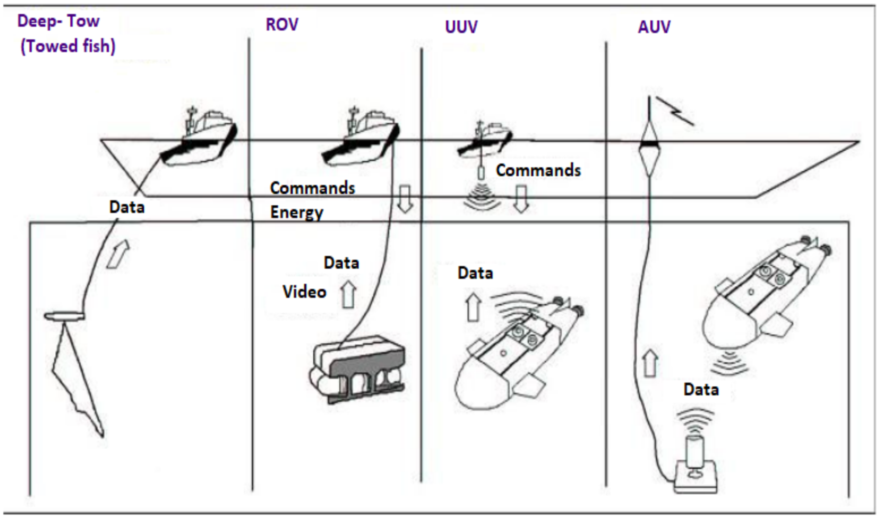


Figure 1.7 – *Different types of submarine vehicles links* (5)

predecessors of underwater vehicles in everyday operation around the world.(37), (5) .

- Submarines: The first category of underwater vehicles are submersibles and submarines:
- Submarines are large vehicles controlled by a crew that may reside there during the mission time, especially in the military context..
- The submersibles are lightweight and are designed for exploration of great depths. Their key tasks were: the search for hydro-thermal sources in the oceans; interventions on wrecks; etc.

- Underwater robots: Underwater robots are autonomous underwater vehicles. It is either automatically propelled and fitted with an on-board energy source or supplied by its carrier vessel. It can be remotely operated (usually wire-guided) or stand-alone. These robots are fast in data acquisition and have a high capacity to protect any sort of information (physical, acoustic, visual) in digital form according to their storage and processing capabilities. Some of them were used as a platform fitted with samplers or different sensors. This makes them a possible source of science and technological development. The robotic group also uses the American acronym "UUV" ('Unmanned Underwater Vehicle') for these vehicles.
The degree of autonomy (decision / energy) of these robots is defined by the existence of their surface connection (see Figure 1.7). We can then settle on the first classification of two types of underwater vehicles: remotely operated vehicles connected by a cable to the surface and autonomous vehicles connected by an acoustic connection.

- Remotely operated underwater vehicles (ROVs): These are machines controlled by an operator through a ground station or on a boat. They are connected to the surface via a

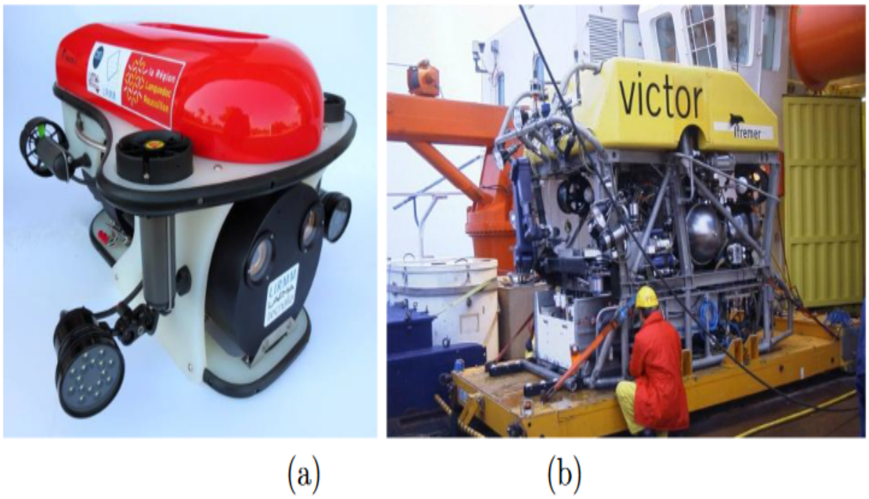


Figure 1.8 – *Example of a ROV robot. (a) L2ROV, from LIRMM, (b) Victor 6000, from Ifremer* (2)

cable through which passes motor controls, energy and acquired data (38). The presence of the cable induces disturbances on the robot's dynamics and complicates its control, which disturbs its stability and affects its ability to perform the required tasks perfectly (see Figure 1.7). However, its presence allows easier recovery of the vehicle. Some examples of ROVs can be cited as the "observe" (from Subsea Tech) as well as the "L2ROV" (from LIRMM) (see Figure 1.8 (a)). Thus, the characteristics of ROVs are as follows (2):

- They are usually over-actuated.
- They have the capacity to park at a fixed point (32).
- They are connected to the surface by an umbilical

In most models, they are often fitted with cameras and even manipulator weapons in most versions, giving them a great ability to manipulate objects. They can perform complex operations, such as::

- Maintenance of underwater structures: the ROV H2000 from Eca-Hytec, equipped with 2 manipulator arms, is capable of performing alteration parts.
- Deep inspections (1000m) in the offshore industry: ROV Victor6000 from Ifremer (see Figure 1.8 (b)).
- Monitoring and demining.

- Autonomous sensor networks with underwater vehicles: Unlike ROV, these Autonomous Underwater Vehicles (AUVs) are not fitted with a cable connection, can operate without clips, or remote control, they also bear their own energy and therefore have a broad range of applications in oceanography, environmental monitoring and underwater

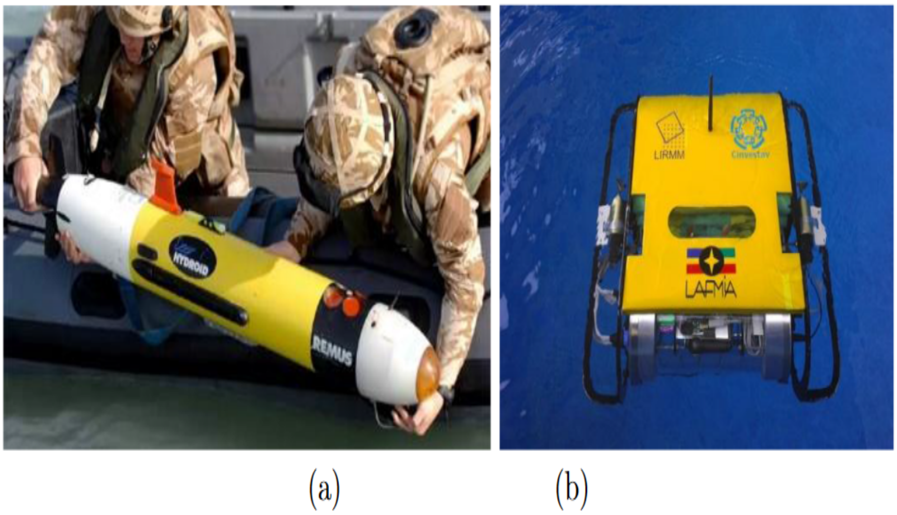


Figure 1.9 – *Examples of coastal AUVs: (a) Remus, hydroid robot camera and (b) Lirmia2, LAFAMIA / LIRMM* (2)

resource studies(31).
From a few hundred meters of depth, the structure, dimensions and characteristics of AUVs change. AUVs can thus be classified into 2 classes depending on the maximum immersion depth reached. We will then talk about coastal AUVs and Grandfonds AUVs.
Coastal AUVs: Remus (USA) of Hydroid, Gavia of Hyfmind and Lirmia2 from a collaboration between Lirmm and LAFMIA (see Figure 1.9 (b) and 1.9 (a)).

Experimental work has shown the relatively inexpensive feasibility of underwater AUVs fitted with a range of underwater sensors that can penetrate any depth in the ocean (35). Therefore, they can be used to enhance the capabilities of underwater sensor networks.
The integration and improvement of static sensor networks with AUV is a still unexplored area in research that almost requires new network coordination algorithms, such as:

- **Adaptive sampling:** This involves control methods for ordering mobile vehicles to locations where their data would be most useful. This technique has been proposed for novel surveillance tasks (39). For example, node density may be increased in an area where a high sampling rate is needed for a given phenomenon to be monitored.
- **Auto-configuration:** This includes monitoring procedures to automatically identify connectivity holes due to node failure or channel attenuation and to request the intervention of an AUV. In addition, the AUV may be used for the installation and maintenance of the network sensor infrastructure or for the deployment of new sensors. They can also be used as temporary relay nodes to reconnect. Solar energy

systems will improve the lifetime of AUVs, i.e., they do not need to be stored and recharged on a regular basis. Solar-powered AUVs are thus able to collect information continuously for periods of time over a span of several months.(40).

Underwater robots, including robotic vehicles and autonomous underwater drones, will help us better understand marine environmental problems, protect ocean resources from pollution, and use them effectively for human well-being.

1.3.5 Application domains of underwater acoustic sensor networks in IoUT

In the Internet of Underwater things, Underwater sensors have many application domains as they can help to measure different quantities such as water quality and study its characteristics, temperature, density, salinity, acidity, chemicals, conductivity, pH (elastic magneto sensors), oxygen (Clark type electrode), hydrogen, dissolved methane (METS) and turbidity. There are other sensors for underwater measurements, such as hydrothermal Sulphide sensors, Silicate sensors, Volta metric spectrophotometry sensors, Gold-amalgam electrode sensors for ion sediment measurements of metal with ionic selectivity (analysis) and so on. Here following the presentation of examples of the application fields and their definition.

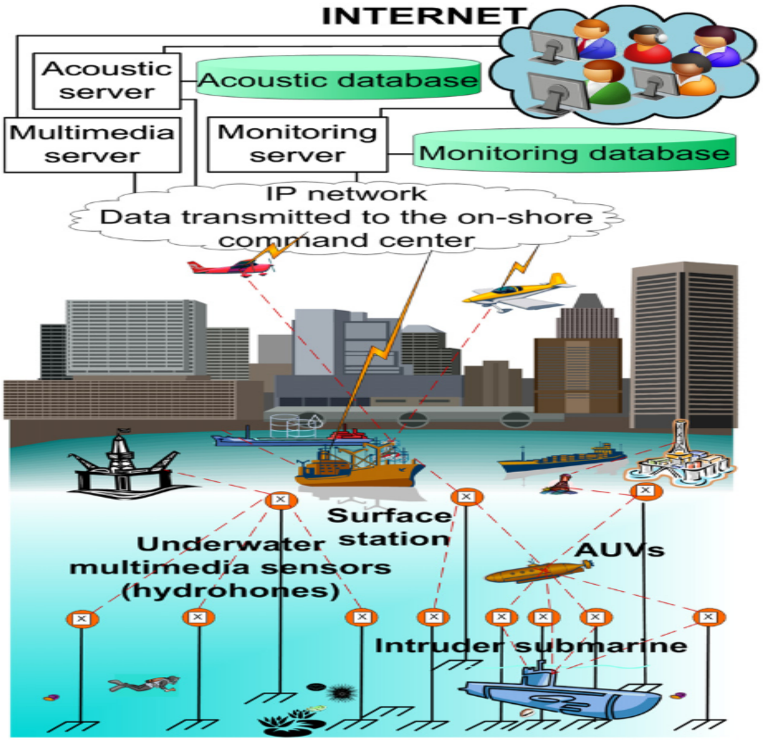


Figure 1.10 – *Harbor security scenario* (6)

- **Ocean Sampling:** Underwater sensors and Autonomous Underwater Vehicles (AUVs) have the synoptic capability to conduct cooperative, adaptive sampling in three-dimensional coastal marine environments to create geological and biological databases(41).

- **Environmental monitoring:** Underwater sensor networks allow the monitoring of various forms of pollution (chemical, biological and nuclear), the monitoring of marine currents and winds, weather forecasts, climate change detection, understanding and forecasting of the effects of human activities on marine environments, and biological monitoring, such as monitoring of marine biological or aquaculture activity.
- **Underwater explorations:** Underwater sensor arrays can help detect underwater oil deposits, identify routes for laying submarine cables, and explore valuable minerals. It can also be used for underwater archaeology and wreck studies.
- **Disaster prevention:** underwater sensor networks have the capacity to assess seismic activity from a distance and this enables the provision of tsunami warnings for coastal areas(42), as well as the analysis of the impact of sea-quakes.
- **Assisted navigation**: Sensors can be used to identify hazards on the seabed, locate hazardous rocks in shallow water, mooring positions, locate submerged wrecks, and perform bathymetry profiling.
- **Distributed tactical surveillance:** Autonomous Underwater Vehicles (AUVs) and fixed underwater sensors can work together to control surveillance (Harbor security(6) as example showed in Figure1.10), identification, positioning and intrusion detection areas. Underwater sensor networks can achieve greater accuracy than conventional radar or sonar systems, and can also detect and identify low-signature targets through a combination of measurements from various types of sensors.
- **Aquaculture and underwater robotics:** a diversification principle that parallels the successive mortality of various types of farmed fish, such as oysters, as depicted in Figure 1.11. Following the evolution of less and less costly underwater robots, which are becoming less and less costly, the underwater sensor network of small underwater structures has submerged several kilometers from the coast and on the shallows, carrying high-end algae, shellfish and mollusk cultures for medicine and cosmetics, all of which are monitored by robots and divers who would limit the use of robots.

1.3.6 Comparaison of WSNs in IoT and UWSNs in IoUT

Underwater environment brings new challenges that do not occur in terrestrial networks because of the radical difference in terms of propagation in the acoustic environment or even in radio. Table 1.1 summarizes the fundamental differences between TWSNs and UWSNs.

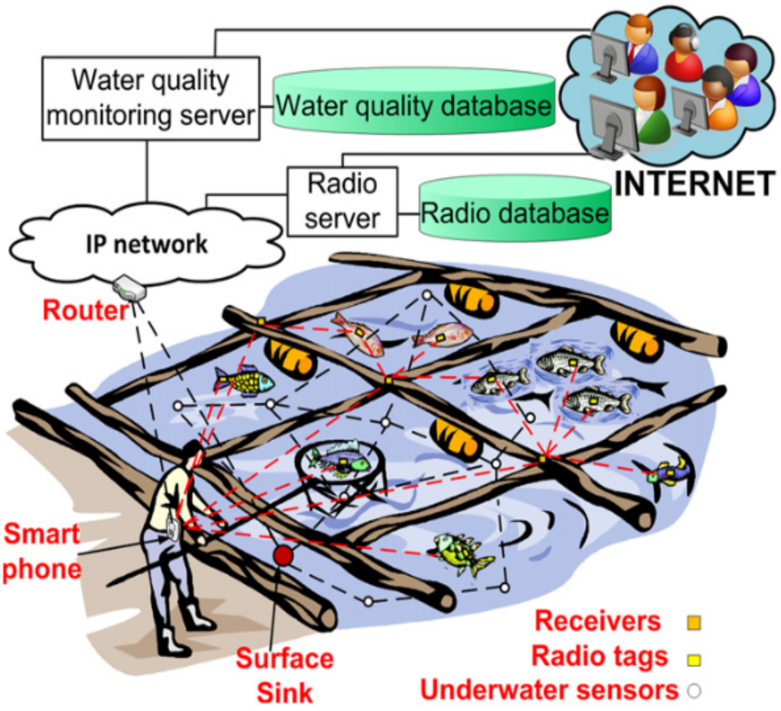


Figure 1.11 – *Fish farm scenario* (6)

Table 1.1 – *Basic communication's differences between UWSNs and WSNs* (7)

	(Terrestrial) WSN	**UWSN**
Communication Media	RF Waves	Acoustic Waves
Frequency	High	Low
Node size	Small	Large
Deployment	Dense	Sparse
Power	Low	High
Energy consumption	Low	High
Propagation delay	Low	High
Bandwidth	High	Low
Path loss	Low	High
Cost	Inexpensive	Expensive
Memory	Sensor nodes have low capacity	Sensor nodes require large capacity

1.4 Communication's Challenges in IoUT

1.4.1 Introduction

The hostile nature of this marine environment especially the submarine wireless channel makes submarine sensor connection a difficult task (43). Therefore, establishing such a network imposes very deep studies on the communication of the network and the transmission of data. Underwater environment is impacted by various aspects such as bandwidth usage limitation, surrounding noise and large acoustic propagation delays. However, communication itself is an outstanding challenge. Being able to connect these subsea devices and create a reliable communication channel between them are the keys to make all these applications viable, including unmanned vehicles and robots in a harsh environment for humans (44). In what follow, we will present a general view of the communication's challenges that face the Underwater Wireless Sensor Networks.

1.4.2 Underwater Communication techniques

The Aquatic Wireless Communication System includes 3 transmission techniques: radio, optical and acoustic.

- **Communication by radio wave:**
  Wireless communication standards for radio waves reach the realm of giga hertz in the air, for example standards: IEEE 802.11, Home RF, Bluetooth which operate in the 2.4 GHz band. Nevertheless, the attenuation for high frequencies, in water, turns out to be extremely high and arrives for a frequency of 2.4 GHz to 1695 dB / meter in the sea and to 189 dB / meter in fresh water, while assuming an average conductivity of 4 ω-1 / meter in seawater and 0.05 ω-1 / meter in fresh water. Thus, electromagnetic radiation does not penetrate the marine environment beyond a few tens of meters in the best case: Lambda (λ) of blue (the sea) is the wavelength less rapidly absorbed in the water of the light spectrum (hence the blue planet). For very low frequencies (30 to 300 Hz), the attenuation decreases, but the propagation of the electromagnetic wave over long distances then requires large transmission-reception antennas, therefore high transmission powers (45).
- **Optical wave communication:**
  Within the means of communication the most exploited can unearth optical waves. In the marine environment, signal attenuation is not posed in this type of wave, however there is a main disadvantage of light propagation in water caused by the phenomenon of scattering. This problem imposes high aiming precision on the transmitting node, which cannot be achieved due to the mobility of the sensor in water and the location varying from one to the other. In addition, the incompatibility of the environment, the variation of the refractive index and the existence of objects of marine microorganism create a handicap in the face of the propagation of the optical wave(46).
- **Communication by acoustic waves:**
  The development of long-distance communications from the beginning of the 20th century had to take into account a fundamental characteristic of our planet: the vast majority of its surface is made up of salt water, which is a good conductor of electricity (therefore unsuitable for the propagation of electromagnetic waves over

great distances) but also support for the rapid propagation of acoustic waves. These properties are used, in particular in the field of ultrasound, for the production of communication, detection and measurement systems: sonars. Marine animals also use sound waves for communication or echolocation.

The transmission of data through underwater acoustics is usually achieved in the form of digital signals. For transmitting information, it is encoded in binary symbols, the symbols being the subject of the emission of different acoustic signals; for example, the symbols ’0’ and ’1’ may correspond to the emission of two pulses of different frequencies (FSK Frequency Shift Keying Modulation) or even to the change of phase of a sine wave (PSK Phase Shift Keying Modulation).

Underwater acoustic systems for the transmission of digital signals benefit from methods developed in the field of telecommunications. Only the international submarine telephone standard requires analog SSB modulation about an 8 kHz carrier (with very low sound quality). Acoustic communication is the most flexible technique and is widely used in the network of underwater sensors due to the low attenuation in water.

1.4.3 Comparison of communication techniques:

The techniques of wireless signal transmission underwater are not based only on the acoustic wave, each of these techniques is characterized by these positive and negative points. The radio wave is specified by its propagation at different distances, its low frequency (30-300 Hz), it requires large antennas and wide antenna power. On the other hand, optical waves are able to transmit information underwater because they do not experience great attenuation at short distances, but their negative side is the phenomenon of dispersion. While optical signals require high precision for laser beams to transmit, also the optical waves are particular by a weak range. Therefore, although laser technology is perfect, the best solution for underwater communication is the acoustic wave under harsh conditions (8) as it can see in Table.1.2.

Table 1.2 – *Theoretical comparison of acoustic, Optical and Radio waves in seawater environments* (8)

	Acoustic Wave	**Radio Wave**	**Optical Wave**
Speed (m / s)	$1.55x10^3$	$3x10^8$	$3x10^8$
Bandwidth	KHz	MHz	$10-150MHz$
Effective Range	$1Km$	$10m$	$10-100m$
Power loss	$> 0.1dB/m/Hz$	$28dB/1Km/100MHz$	*Depends on turbidity*
Antenna Size	$O.1m$	$0.5m$	$0.1m$
Antenna Complexity	*Medium*	*High*	*Medium*
data Rate	$Upto100Kbps$	$Upto10Mbps$	$Upto1Gbps$

1.4.4 Underwater channel characteristics

Typical physical carriers for underwater communication signals as was detailed before, are radio frequency (RF) electromagnetic waves, optical waves, and acoustic waves. Among the three types of waves, acoustic waves are used as the primary carrier for underwater wireless communication systems due to the relatively low absorption in underwater environments. This fact explains an increase of the literature related to this technique of retransmission whose characteristics are the following ones:

Transmission or path loss: transmission loss is caused by two factors, attenuation and geometric propagation.
- The attenuation is mainly provoked by absorption due to the conversion of acoustic energy into heat and increases with distance and frequency (47). Figure 1.12 shows the acoustic attenuation with varying frequency and distance for a short range shallow water UW-A (Underwater-Acoustic) channel, according to the propagation model. The attenuation is also caused by scattering and reverberation (on rough ocean surface and bottom), refraction, and dispersion (due to the displacement of the reflection point caused by the wind on the surface). Water depth plays a key role in determining the attenuation as stated in (4).
- Geometric spreading: refers to the spreading of sound energy as a result of the expansion of the wave fronts. It increases with the propagation distance and is independent of frequency. (4) Outlines the two common kinds of geometric spreading: spherical (Omnidirectional point source), which characterizes deep water communications, and cylindrical (horizontal radiation only), which characterizes shallow water communications. Figure 1.12 illustrates the impact of both distance and frequency over the path loss.

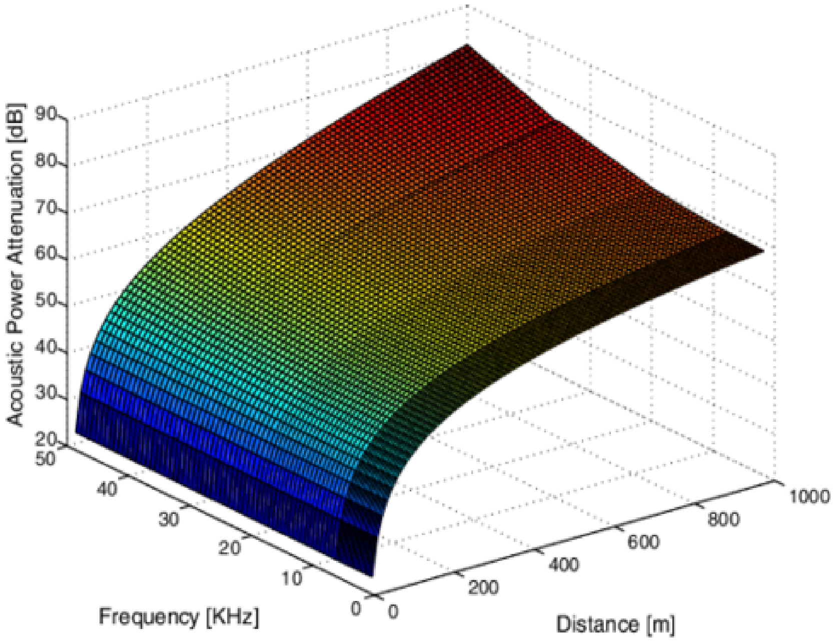


Figure 1.12 – *Path loss of short range shallow UW-A channels vs distance and frequency in band* $150kHz$

Noise: It can be classified as man-made noise and ambient noise. The former is mainly caused by machinery noise (pumps, reduction gears, power plants), and shipping activity (hull fouling, animal life on hull, cavitations), while the latter is related to hydrodynamics (movement of water including tides, current, storms, the wind, and rain), and to seismic and biological phenomena (4, 48) . In (49), the author claims that the of the ambient noise related to four sources:turbulence in the water, breaking waves, thermal noise and shipping can be approximated by using the empirical formulas (in dB) where f refers to the frequency in kHz:

$$10logNt(f) = 17 - 30logf \tag{1.1}$$

$$10logNs(f) = 40 + 20(s - 5) + 26logf60log(f + 0.03) \tag{1.2}$$

$$10logNw(f) = 50 + 7.5w^{1/2} + 20logf - 40log(f + 0.4) \tag{1.3}$$

$$10logNth(f) = -15 + 20logf \tag{1.4}$$

Where Nt, Ns, Nw, Nth stand for turbulence, shipping, wind and thermal noise respectively.
We then, present the total noise power spectral density for a given frequency f[kHz] as follow:

$$N(f) = Nt(f) + Ns(f) + Nw(f) + Nth(f) \tag{1.5}$$

Multipath: Multipath propagation can severely damage the acoustic signal, as it generates inter-symbol interference (ISI) (47). The two main reasons inducing the multipath effect in underwater environments are discussed in (49) which are sound reflections at the surface, bottom or other objects(cf. Figure 1.13)in the water and the varying sound speed(cf. Figure1.14). The geometry of the propagation of multipaths depends on the configuration of the connection. Vertical channels are distinguished by a low time dispersion, whereas horizontal channels can have long distances from several paths. The calculation of the spread is a powerful function of the depth and distance between the transmitter and the receiver.

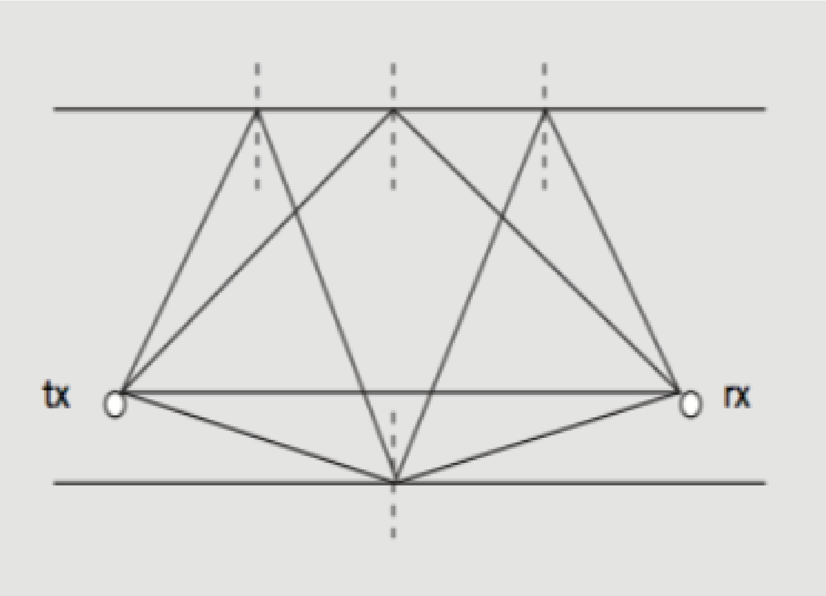


Figure 1.13 – *Multipath due to reflection on surface and bottom*

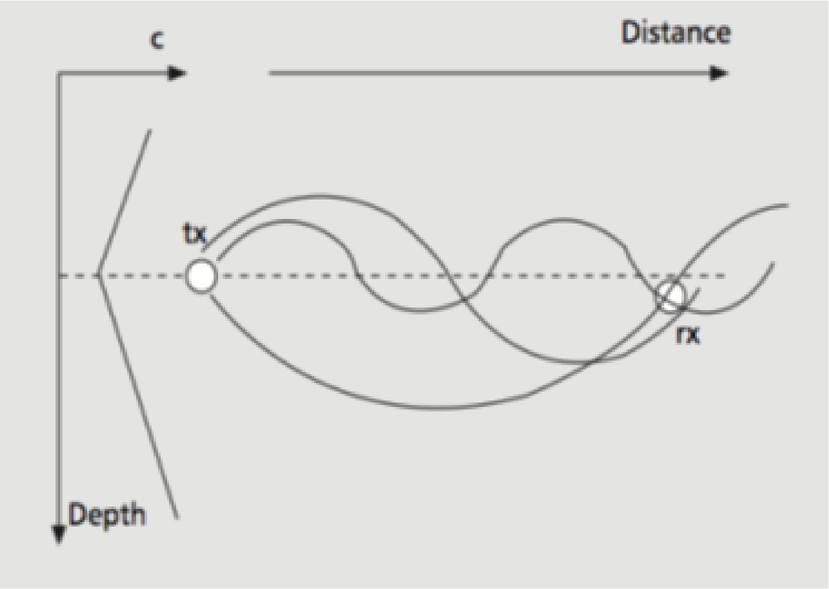


Figure 1.14 – *Multipath due to varying sound speed*

High delay and delay variance: The propagation speed in the UW-A channel is five orders of magnitude lower than in the radio channel. This large propagation delay (0.67s/km) and its variance can reduce the system throughput (48). The underwater acoustic propagation speed can be expressed empirically as in (47):

$$C(z,S,t) = 1449.05 + 45.7t5.21t^2 + 0.23t^3 + (1.3330.126t + 0.0009t^2) * (S - 35) + 16.3z + 0.18z^2 \tag{1.6}$$

Where $t = 0.1 * T$, T represents the temperature in °C, S is the salinity in ppt, and z is the depth in km. The propagation speed varies between (1450 m/s -1540m/s).

Doppler spread: The Doppler frequency spread causes degradation in the performance of digital communications and generates two effects: a simple frequency translation and a continuous spreading of frequencies, which constitutes a non-shifted signal. The former is easily compensated at the receiver when the effect of the latter is harder to be compensated for (50). Assuming a channel that has a Doppler spread with bandwidth B and a signal has symbol duration T, and then there are approximately BT uncorrelated samples of its complex envelope. When BT is much less than unity, the channel is said to be underspread and the effects of the Doppler fading can be ignored, while, if greater than unity, it is said to be overspread (4).

1.4.5 Constraints and challenges in UWSNs

Intense research is currently underway on the establishment of effective network solutions for underwater acoustic sensor networks. Although there are several recently developed protocols for wireless sensor networks, the different characteristics of underwater communication systems present different challenges, which can be summarized as follows:

- The propagation delay in the submarine communication is 5 times higher than the Hertzian chains.
- Due to the extreme characteristics of the underwater channel as gray region, the temporary loss of connectivity is often followed by high bit error rates.
- The high noise resulting from transport and machinery activity is a concern in the networks of underwater sensors.
- Underwater sensor devices are very expensive and their availability on the market is very limited.
- The bandwidth available for aquatic sensors is extremely limited.
- The undersea networks are seriously damaged, this is mainly due to the multi-trip.
- Battery power capacity is limited and generally the batteries cannot be recharged, also because solar energy cannot be harnessed.

During this research, we were interested in highlighting the following challenges:

- The speed of sound slower than the speed of light by several orders of magnitude, the transit time of packets between underwater wireless communication nodes is another major challenge. This can lead to packet collisions and thus lower direct network sales rates. Unfortunately terrestrial transport protocols medium are not adapted to this type of aquatic medium. the transport layer is totally unexplored area, unlike the physical, network and data link layers that have been significantly enhanced (35). Improving TCP in UWSNs may seem a big challenge to get out with a reliable data transmission protocol adapted to acoustic channel in this environment, this is the reason why **Chapter.**2 was devoted to study the behaviour of the traditional TCP transport protocols in this kind of environment. Subsequently we presented two improvements studies of NewReno TCP and a new TCP protocol better adapted and with more efficiency in this submarine environment.
- Underwater sensors are experiencing failures due to corrosion and fouling. Implementing marine networks with a system that adapts and evokes failures or link failures in submarine communication is very rare because of the high price that a UWSN agent can cost (51). an adaptive learning strategies will be presented in

Chapter.3 to modify the depth of deeply anchored Limited Mobility Agents (LMAs) in order to improve the communication of the underwater network.

- Redundancy and unnecessary data transmitted by Sensors; challenges that are very offen in UWSNs. Guaranteeing a unique collection of information allows efficient data transmission as well as more lifetime for the network used. In addition we have to ensure a good coverage of the network. This problematic is going to be studied in **Chapter.**4 where adaptation methods will be applied for better coverage with less data redundancy for better quality monitoring.
- With submarine wireless communication network protocol designs, energy savings are a major concern, especially for long-term aquatic monitoring and sensing applications and due to the mobility of its nodes, most energy-efficient protocols designed for terrestrial wireless networks are not feasible. Therefore, more attention should be paid to the architecture of routing protocol in submarine wireless networks. A study of the effect of routing protocol in term of energy saving will be raised in **Chapter.**5.

1.5 Conclusion

In this chapter, we presented a general view of WSNs and UWSNs with their different applications, problems and design domains. A distinction was made between these two worlds, and we presented a general view of challenges faced in the communication through an Underwater Wireless Sensor Networks. In the next chapter, we will present a comparative analysis of the actions of WSN transport protocols in the underwater environment with various metrics and scenarios and using various routing protocols. On the basis of this work, a thorough analysis and adaptation of the TCP parameters will conclude with the submission of a new Underwater TCP.

CONTRIBUTIONS

Chapter 2

TCP in Internet of Underwater Things

Contents

2.1 Introduction

Internet of Underwater Things (IoUT) are defined by Underwater Wireless Sensor Networks (UWSNs) which are a networks that consist of sensor nodes deployed in underwater environment to perform specific tasks such as monitoring physical phenomena in water and target detection. this network admits a large set of applications that ranges from scientific, environmental, commercial to military applications.In an underwater environment, the exploration and improvement of ocean observing systems is done using a specific submarine support dedicated to obtaining and exchanging important information. The transmission of data is done using submarine sensor. The use of this type of network is fundamental in various fields of application, namely the control of pollution of oceans and rivers, as well as the forecast of natural disasters such as the tsunami. The oil industry also uses UWSNs (52).

With the recent development of the last decade in acoustic transmission techniques, data transmission in underwater environment is becoming a reality. Previous studies have suggested that the TCP protocol is inadequate for UWSN. The reason behind this, is that TCP depends on accurately measuring the Round Trip Time (RTT) in order to appropriately adjust the congestion window. The long propagation delay of acoustic waves in water increases the RTT which adversely affects the TCP throughput. Furthermore, the high variability of the RTT makes it difficult to adjust the timeout value as the TCP congestion mechanism. It is difficult to distinguish between long propagation delays and missing acknowledgment. Furthermore, the TCP performance underwater is further reduced due the high error rates on the acoustic connections though, this phenomenon is also encountered in wireless radio networks(39).

Improving TCP performance in WSNs has been a focus of research activity for several years but research in underwater networks environment has not been deeply addressed. The characteristics of the underwater wireless network have a significant impact on TCP performance due to its differences from standard and wired wireless networks. The use of acoustic waves gives a very long propagation time, with a low bandwidth and a high probability of error and a specially limited energy with considerable packet losses due to what is error of privilege, resulting in delays of significant variations. These sudden delays violate most of TCP design assumptions.

In this chapter, we will firstly present a study of the behavior of two TCP variants under two different routing protocol families widely deployed in UWSN, the first being derived from the proactive routing protocol family: Destination Sequenced Distance Vector (DSDV) and the second comes from the reactive routing protocol family: Ad hoc On-Demand Distance Vector (AODV). In order to better discover the effect of UWSNs on the behavior of TCP Vegas and TCP New Reno and in order to provide some guidelines that

will service for performance tuning, we start by investigating the impact of Packet Size and the number of TCP connections performances over underwater environment using AODV and DSDV as routing protocols. Secondly, we will focus on evaluating the performance of TCP NewReno (53), as a transport layer protocol in a UWSN in order to improve its performance in this type of environment. Particular attention was paid to study the effect of controlling the maximum TCP window and to quantify the effects of various factors in a UWSN and simulating different scenarios to define the best parameter setting that ensures good communication in this subsea network.
The fifth section of this chapter proposes an improvement of TCP New-Reno to enhance its performances according to the UWSN characteristics by modifying the parameters values mainly adjusting the RTT timeout. In other words, we propose an appropriate setting of the TCP variant NewReno which is used especially for the UWSNs.Section 6 defines a new TCP adapted to this environment named Underwater New Reno tcp (U-New Reno tcp). Our proposed protocol is able to monitor the maximum size of the congestion window and adapt the RTT waiting time at the same time. The last Section gives a general conclusion of our works.

2.2 Related Work

There is a myriad available studies that focus on the performance of TCP in terrestrial WSNs. However, very few studies have tackled the counterpart problem in UWSNs. In this part, we shall present some previous works related to TCP performances using DSDV and AODV routing protocols in both UWSN and in WSN for the sake of completeness.In addition to some previous works which treated the improvement of TCP whether in the normal network WSN or in the submarine network UWSN.

Several studies were set up to assess and improve the routing performance of AODV and DSDV protocols in WSNs (54, 55, 56). In an ad hoc network, AODV being a reactive protocol it gives the mobile nodes an easy and quick adaptation to dynamic link disputes since it holds information from routes to active destinations only which allows mobile nodes to have accurate information in a short time (57).The proactive routing protocol DSDV is suitable for routing ad-hoc networks from the conventional Routing Information Protocol (RIP). It is based on adding the sequence number as a new attribute in the RIP routing table, based on this new attribute the mobile nodes can prevent the formation of routing loops because they now know the last value of the route saved to those already outdated (58). The use of AODV is highly recommended for the transmission of high video packets as it is proven in the comparative study in (59) with DSDV and OLSR another routing protocol for Optimized Link State Routing Protocol (60), since it gives good performances in terms of bandwidth with low packet jitter.

To address the challenges of information routing in UWSNs, different routing protocols have been proposed. Vector based forwarding (VBF)(61) is part of this underwater proposed protocols, (DBR) (62) is also a suitable routing protocol for this environment. In (63) Zhang, Li, and Chen present LAFR routing protocol based in link state and feedback adaptive. Furthermore, to the work in (64) gives a new routing protocol called DREE based on reliable distance and energy efficiency. In addition of a new routing protocol proposed in (65) to solve the problem of energy consumption in a reliable and efficient way. In (66), the authors describe a new efficient data collection assisted and in (67) a layer-based flood routing protocol based on the layer angle is presented. Thus, decreasing the energy consumption of the network and increasing the data transfer rate with a short delay are the basic requirements for the proper functioning of different types of marine applications that all these proposed protocols must meet(68), while different performance metrics for mobile ad hoc networks were considered and compared such as routing overhead, Packet Delivery Ratio, throughput and end to end delay (61, 62, 63, 64, 65, 66, 67).

In (69), authors present us a performance analysis specific to each type of TCP, starting with TCP Tahoe, passing by TCP Reno and TCP New Reno (70) and at the end TCP Vegas under congestion. In this work the authors resorted to NS-2 to simulate these four types of TCP and analyze the results in terms of throughput, packet drop rate, and latency. Different research studies to solve TCP congestion problems in MANET were reported in the literature. For instance, Hamamreh and Bawatna (71) present the most varied TCP that have the ability to maintain a good end-to-end behavior. Furthermore, the authors give a study analysis to adjust the value of the network size, define the mobility of the nodes with different conditions of the wireless channels in order to increase the performance of TCP over MANET. In aquatic communication several studies have proved that among essential characteristics in a long link is the data rate as well as a wide propagation delay, moreover all applications that are based on TCP their features severely degrade the end to end yield. To deal with this problem, a new Linear Coded Digraph Routing (LCDR) was presented by Chin-Ya, Parameswaran and Kewal in (72). Based on the multiple path routing solution combining this protocol with the local sequencing solutions it also uses the network coding principle and adapts a management mechanism for the detection of data duplication.

During congestion, a source node cannot restore its congestion window in a short time to return to its normal state which is negatively reflected on the return of TCP performance when RTT is too long (73). In (74) to mitigate and avoid congestion, the authors propose an efficient model to improve the functioning of TCP, this model uses the segmentation in a way that its transmission of data is faster and to increase its throughput. The proposed mechanism is shown to improve the transmission mechanism of TCP in a considerable way. In (75), Albuquerque et al. study the effect of TCP Packet Size in a congested link environment and correlated errors channel, they present the relationship of the ratio α with respect to round trip links in a network of a single TCP Reno connection. Albuquerque et al. support their study by performing extensive simulations using ns-2 Network Simulator.

To resolve the problem of congestion, an improved variant of TCP is proposed in (76) called TCP NewBR. In this work, authors select the more accurate ACK time interval based on the bottleneck links to adaptively estimate the available bandwidth. NewBR TCP uses modified algorithms for faster retrieval depending on the length of the bottleneck link queue.

Another variant was the subject of study in wireless networks. Authors in (77) have studied the SmoothTCP variant comparing it with the standard TCP in simulated media with and without parasitic errors. The experimental results show that in both cases, SmoothTCP performs better with respect to the variation of round-trip time.

Authors propose in (78) a method which limits the maximum window size. As a result, this leads to increase the throughput. In this multi-hop architecture, the author used fixed nodes numbered from 0 (the transmitter) to N (the receiver) with a range of 70 m between nodes and a link bandwidth set at 150 kbps. They come out with two criteria: the effect of the number of hops and the effect of the maximum sending window limitation over Hybla. Then, they compared the results between improved Hybla and those related to NewReno. It has been shown that: First, although Hybla presented better throughput than NewReno when the number of hops is between 2 and 5, it still requires an adequate setting of RTT0 to reduce the buffer overflow. But this represents a challenging issue because the RTT is not stable in UWSN environment. Second, Hybla did not experience any significant improvement when applying the maximum sending window limitation. Contrariwise, NewReno throughput is improved in all the packet size cases.

2.2.1 Network Model

In this work, to describe the system model used, the architecture depicted in Figure 2.1 as architecture of our underwater sensor network regarding all studies and improvements of this Chapter.

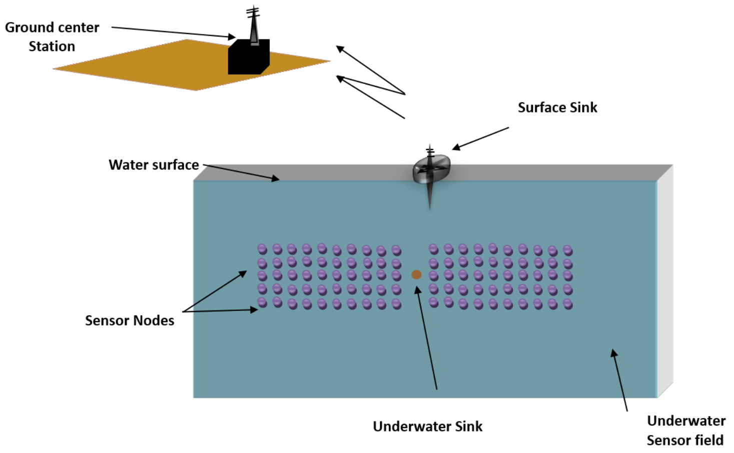


Figure 2.1 – *Network Architecture*

This architecture is composed of a cluster of 100 submarine nodes placed at the same underwater level, this cluster contains the TCP sources and for the data collection we set one single underwater sink in the middle of this cluster. After collecting the data, the submarine sink station will transmit this data to the surface sink which will diffuse these data to the ground center station using radio communication. The transmission range between sensor nodes varies between 75 and 84 m. All sensors are static and placed on a fixed common height, and the initial energy is 10 kJ for each node.

2.3 Assessing the performance of different TCP congestion mechanisms in UWSNs

2.3.1 Introduction

In this work, we start by studing the behaviour of different standard TCP Variants,mainly:
- **Tahoe**: That is caracterized by it's Slow-Start phase + the Congestion avoidance phase + the Fast retransmit phase
- **Vegas**: That is a proactive approach of Reno, modifying the Slow-Start phase + the Congestion avoidance phase + the Fast retransmit algorithms
- **Sack**: That is an enhancement of Tahoe TCP, where the sender only needs to retransmit the segments that have actually been lost.
- **Reno**: That is composed by Tahoe + Fast recovery phase
- **NewReno**:That is based on TCP Reno + Fast recovery modified phase

After many simulations we come out that TCP Vegas and NewReno, give as the best performance results compared to the other variants. In what follow, we will consider these two TCPs Vegas and NewReno for our study.

2.3.2 System Model Analysis

Network Model

the network model used in this work will define the architecture of UWSN as showed in Figure 2.1 in the Network Model of the SubSection 2.3.2. In addition to that, to route data from sending nodes to sink node, AODV and DSDV were implemented as routing protocols then, BroadcastMac, UnderwaterPhy were respectively used for MAC layer protocol to access channel and physical layer.

TCP model analysis

In this work, we consider the performance of TCP Vegas and TCP New Reno in underwater environment under different settings for three main reasons.

First, TCP Vegas is known to be a viable choice in UWSN due to its ability for fast reactivity when it comes to retransmitting a lost segment.

Second, TCP Vegas is able to anticipate congestion, and to adjust its transmission rate accordingly. Third, TCP Vegas has an efficient slow-start mechanism which avoids packet losses while trying to determine the available bandwidth (79). In addition, whenever packets are sent by the sending host, TCP Vegas examines these RTTs (propagation delay) and checks the size of its window. TCP Vegas detects that the network starts to be congested and limits the size of the window when RTTs take large values, and it is assumed that the network is free of congestion and increases again the size of the window when RTTs take this time small values and the window size is updated in the congestion avoidance phase as described in what follow:

TCP Vegas detects incipient congestion by comparing the measured throughput to its notion of expected throughput. Vegas defines a given connection's Base RTT as the minimum of all measured round trip times.Expected throughput is defined as:

Expected = CWND/BaseRTT

CWND= Size of the current congesion window

the current flow throughput is estimated using the actual round trip time as follow:

Actual = CWND/RTT

RTT= the actual round trip time of a packet

Vegas then, compares the actual round throughput to the expected throughput and computes the differenc as :

Diff= (Expected - Actual) BaseRTT

$$\textbf{if} \quad (diff < \alpha) \quad \textbf{then} \quad cwnd(t+t') = cwnd(t) + 1 \tag{2.1}$$

$$\textbf{if} \quad (\text{diff}> \beta) \quad \textbf{then} \quad cwnd(t+t') = cwnd(t) - 1 \tag{2.2}$$

$$\textbf{else} \qquad cwnd(t+t') = cwnd(t) \tag{2.3}$$

Were α and β are two thresholds, $\alpha < \beta$, roughly corresponding to having too little and too much extra data in the network, respectively.

For our second choice, which is TCP New Reno, its default mechanism relies on the size of the window which is changed cyclically in a typical situation since its window size continues to increase until the packets are lost. Two phases can be distinguished where the window size of TCP New Reno is increased, firstly in the slow start phase and secondly in the congestion avoidance phase. When at the instant (t + t ′) TCP receive an ACK packet (acknowledgment) the size of the current window cwnd (t + t ′) is updated from cwnd (t) for each different phase as follow:

For the Slow Start Phase:

if cwnd (t) < ssthresh (t)

cwnd (t + t′) = cwnd (t) + 1

For the Congestion Avoidance Phase:

if cwnd (t) > ssthresh (t)

cwnd (t + t′) = cwnd (t)-1

*With, **ssthresh (t)** refers to the phase change value of TCP from the slow phase to the congestion avoidance phase.*

- cwnd (t)=1 & ssthresh (t)= cwnd (t)/2 When packet loss is detected by retransmission timeout expiration;

- cwnd (t)=ssthresh (t) & ssthresh (t)= cwnd (t)/2 When TCP detects packet loss by a fast - retransmit algorithm expiration

Metrics

Underwater environment are much more challenging than traditional wired networks. We have used here some basic metrics to analyze our simulated scenarios.The TCP throughput, the Packet Delivery Ratio and the End To End Delay were the measures used for performance evaluation. We can describe these metrics as follow:

- **TCP Throughput:** as described in the equation (2.4) refers to the amount of delivered packets to our underwater sink divided by the total time taken.

$$Th(bit/s) = \frac{DP * PS^{8}}{TTS} \tag{2.4}$$

With, Th is *the Throughput* (bit/s), DP is *the Delivered Packets,* PS is*the Packet Size* and TTS is *the Total Time of Simulation.*

- **The Packet Delivery Ratio (PDR)**: as described in equation (2.5), its value presents the proportion of received data by the underwater sink and those generated by the different source nodes as recorded in the trace file (80).

$$PDR = \frac{SRP}{TSP} \tag{2.5}$$

With, PDR is *the Packet Delivery Ratio*, SRP is *the Successful Received Packet,* and TSP is *the Total Sent Packet.*

- **The End to End Delay (E2E Delay)**: it reports the average duration it takes for a packet of data to arrive from source nodes to the underwater sink, and it is calculated as follow (see equation (2.6)):

$$E2ED = \frac{ATD}{TSP} \tag{2.6}$$

With, E2ED is refereed to *the End to End Delay* , ATD is *The Packet's Average Time Duration to rich destination.*, and TSP represents *the Total Sent Packet.*

2.3.3 Simulations and Results

Simulation Setup

Performing real-life experiments is quite challenging and costly in under water environments, therefore, we resort to simulations. We used Aqua-sim environment based on NS-2.30 to simulate our proposed scenarios. Aqua-Sim, for aquatic networks is a simulation tool for acoustic signal attenuation and packet collisions in UWSNs (81). Further, In addition, Aqua-Sim is coded with the same code languages as NS2.30 namely Otcl and C ++ which makes integration with it very convenient and easy for accustomed ns2 users, in addition, whatever changes are made in the wireless package it cannot affect it since all its files are independent, nor NS-2 packages and vice versa (81). In this work, we considered two main different scenarios to measure the aforementioned metrics and we set the experiment parameters according to Table 2.1

Table 2.1 – *Simulation Parameter*

Parameter	Value
Channel	UnderwaterChannel
Propagation	UnderwaterPropagation
PHY	UnderwaterPhy
Antenna	OmniAntenna
Distance	75m, 84m
Frequency	25khz
MAC protocol	BroadcastMac
Mac bit rate	10kbps
Delay	25 us
Routing protocols	AODV/DSDV
TCP agent	Vegas/ NewReno
Simulation Time	500s

- The first scenario as depicted in the flow chart in Figure 2.2 (a) describes the study of the impact of Packet Size. The network starts with a value of TCP Packet Size equal to 10, this parameter takes the values of, 50, 100, 150 and 200.
- The second scenario as illustrated in Figure 2.2 (b) is interested in studying the behavior of TCP Vegas and TCP New Reno when changing the number of TCP connections in the Underwater Network. We start with 8 as initial value of the number TCP Connections and after each simulation we measure the value of each Metrics then, we increment the number of TCP transmitters by 2 and we repeat the simulation until the last value of TCP connections number which is equal to 20.

Results and Discussion

This section reports the simulation results and analysis of different scenarios studied in this work.

Varying the TCP Packet Size

This scenario presents the measures of throughput, PDR and end to end delay while varying TCP packet size. For all the variants of TCP: Vegas and New Reno, we fixed the Size of packet in TCP Sink to 250 and we simulate with 16 TCP connections in a Network of 100 nodes.

We started by calculate the average throughput described in (1), the results are depicted in Figure 2.3 where the evolution and behavior of TCP Vegas, and New Reno describe the average throughput while varying the TCP Packet Size. It is clear that Throughput with DSDV is much better than the throughput with AODV. Both of them give us the highest performance with a value of 100 as Packet Size.

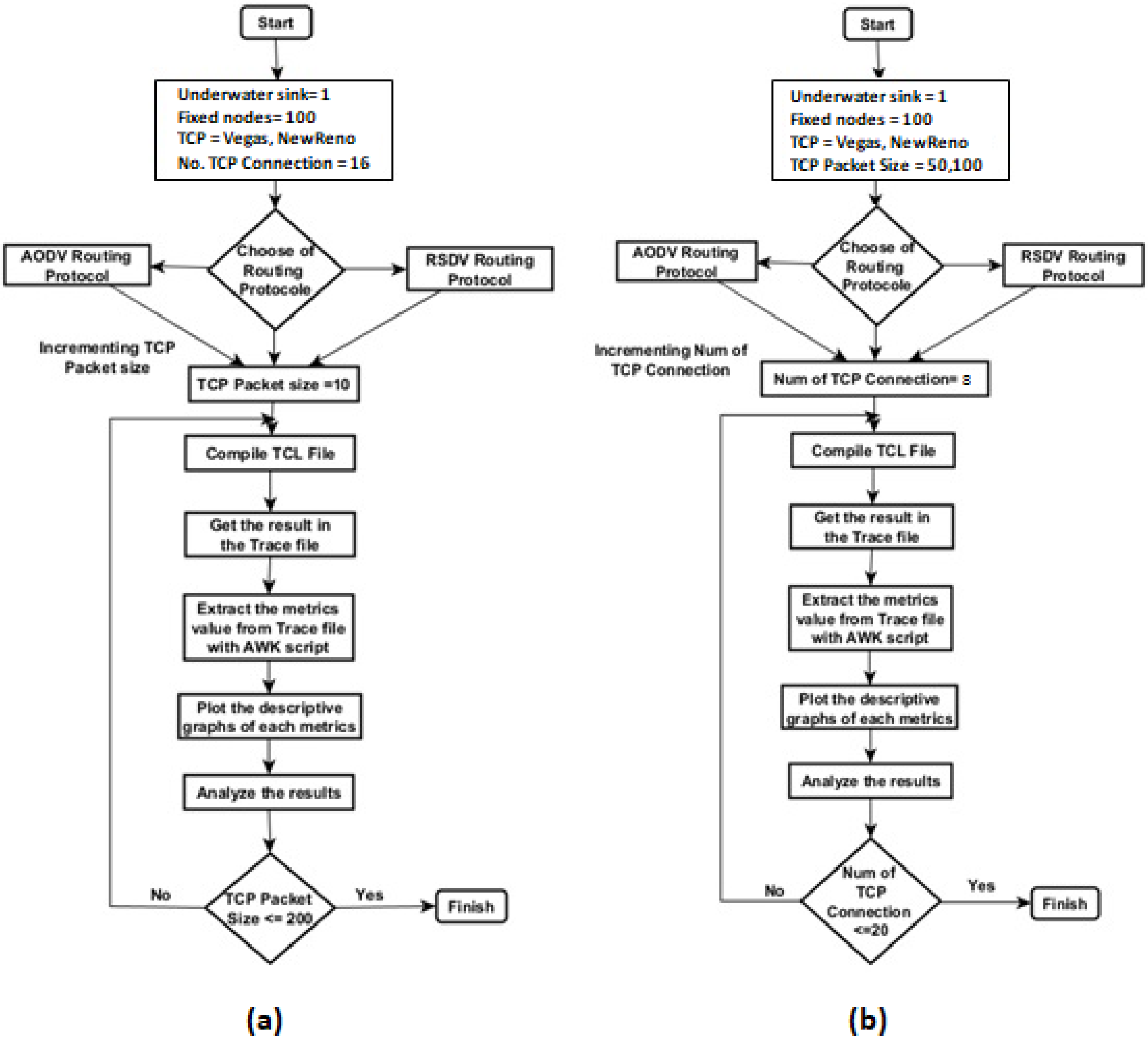
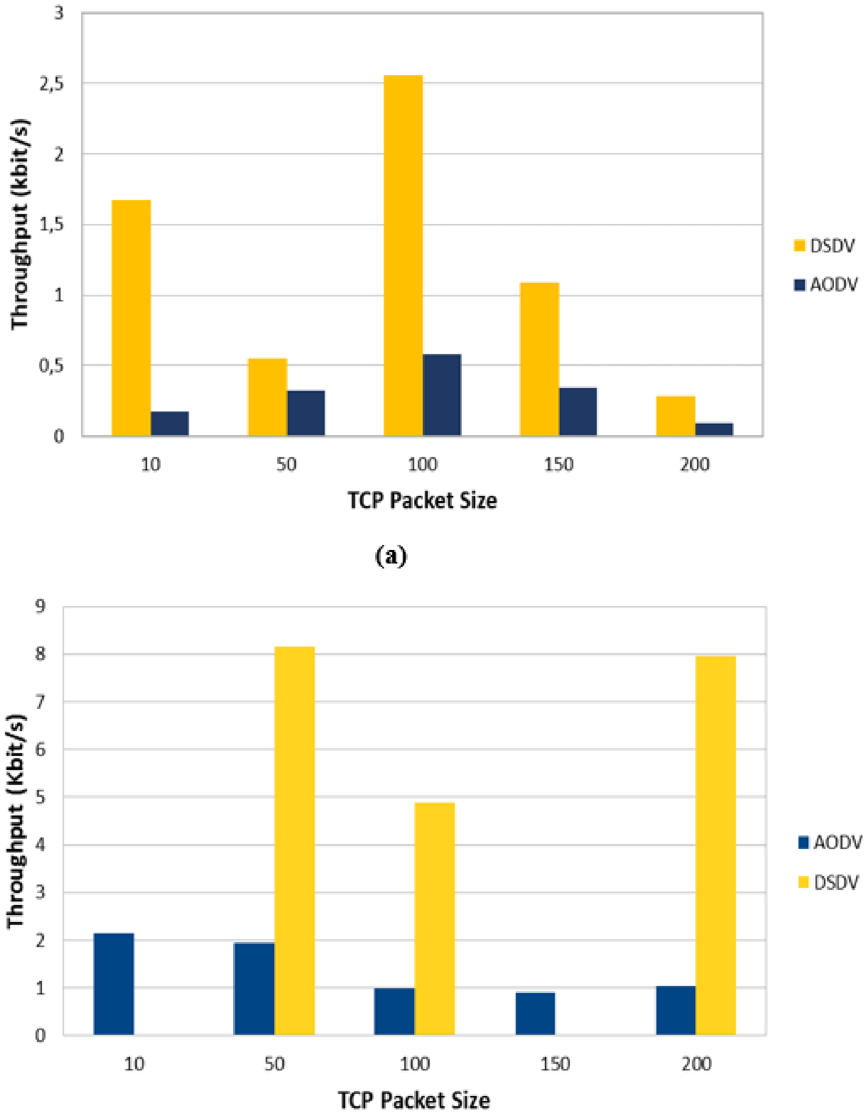


Figure 2.2 – *Flow Chart of our scenarios*



(a)

(b)

Figure 2.3 – *Throughput of (a) Vegas, (b) New Reno*

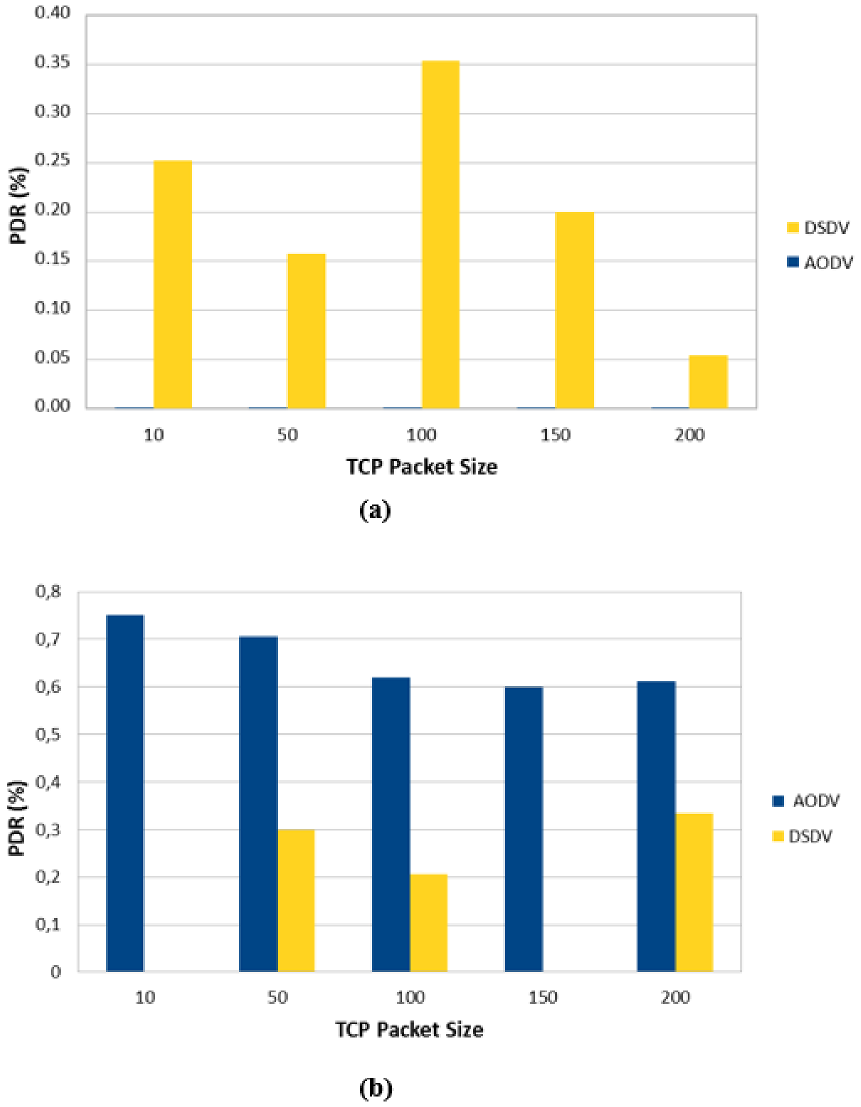
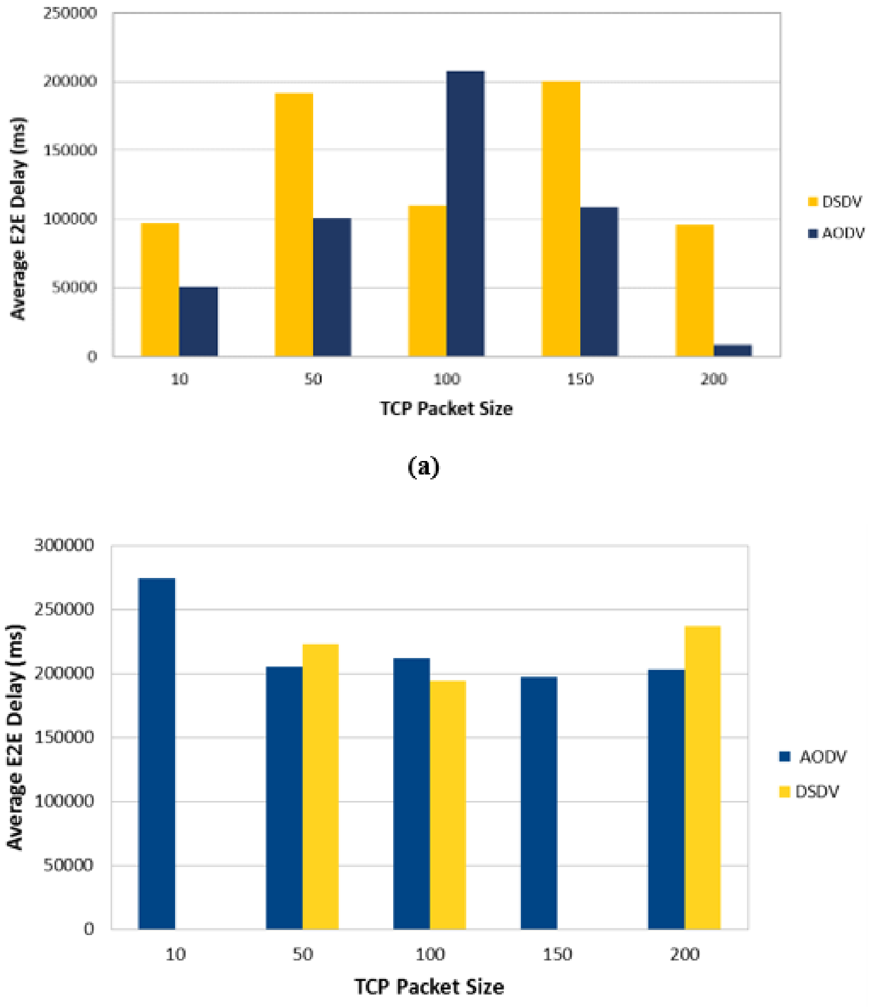


Figure 2.4 – *Packet Delivery Ratio of(a) Vegas, (b) New Reno*

Figure 2.4 depicts the evolution of PDR while Packet Size increases from 10 to 200. As we can observe with (a) TCP Vegas no change has been noticed for AODV but this variation did affect DSDV which gave us good results and especially with the value of 100. The Delivery Ratio has more than 35% of packet delivering with DSDV when the packet size is equal to 100. In (b) TCP New Reno, higher PDR is given while using AODV Routing Protocol which takes up 70% when Packet Size is 10, in other hand, the DSDV routing protocol allows New Reno TCP to rich 35% of delivering Packet.



(a)

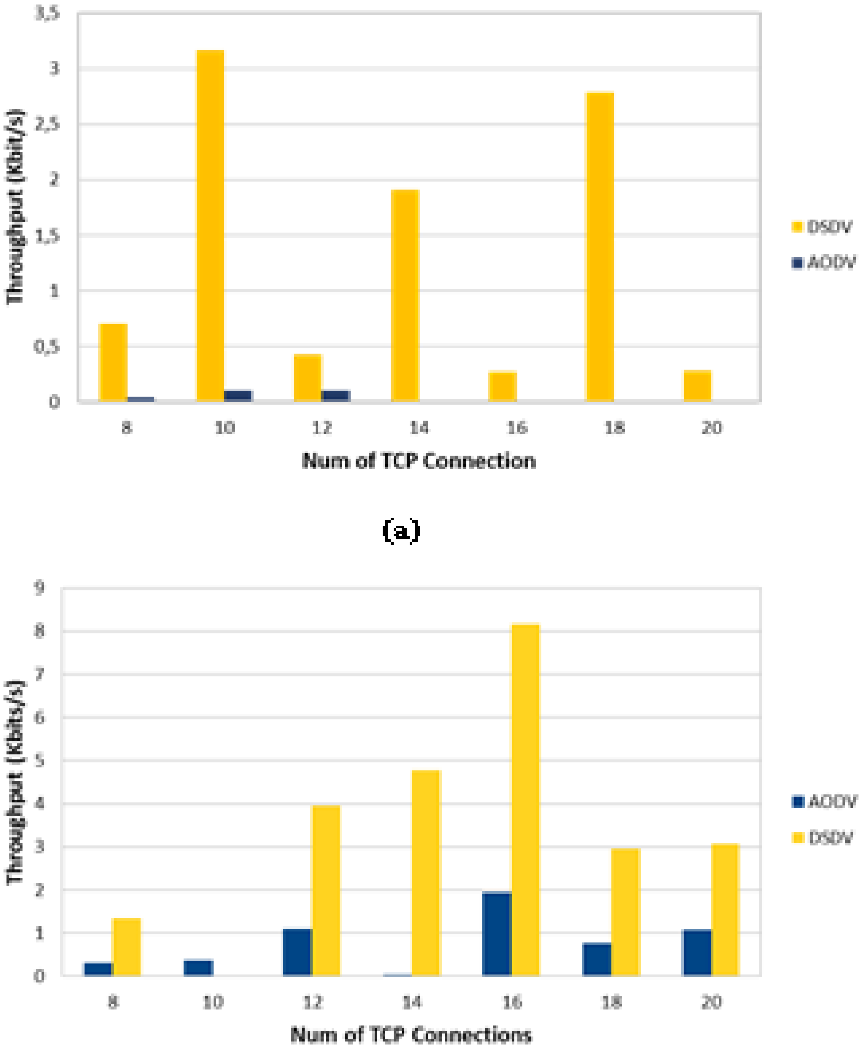
(b)

Figure 2.5 – *Average End to End Delay (a) Vegas, (b) New Reno*

Figure 2.5 indicates the average end to end delay of TCP (a) Vegas and (b) New Reno which are roughly similar for the two routing protocols. With AODV as Routing Protocol it is clear that it takes a slightly more time because it needs to discover the routes before sending data which is not necessary for DSDV while increasing the TCP Packet Size.

Varying the number of TCP connections

In the second scenario we evaluate the impact of TCP connections number, as we start our evaluation by simulating eight TCP connections in the network and we continue to increment until having 20 TCP connections with a network of 100 static nodes. The TCP Packet Size used with TCP Vegas is defined by 100 and with TCP New Reno is 50 based on the results of the first scenario where we found that those sizes yield good result regarding the number of transmitted packets with a good throughput.

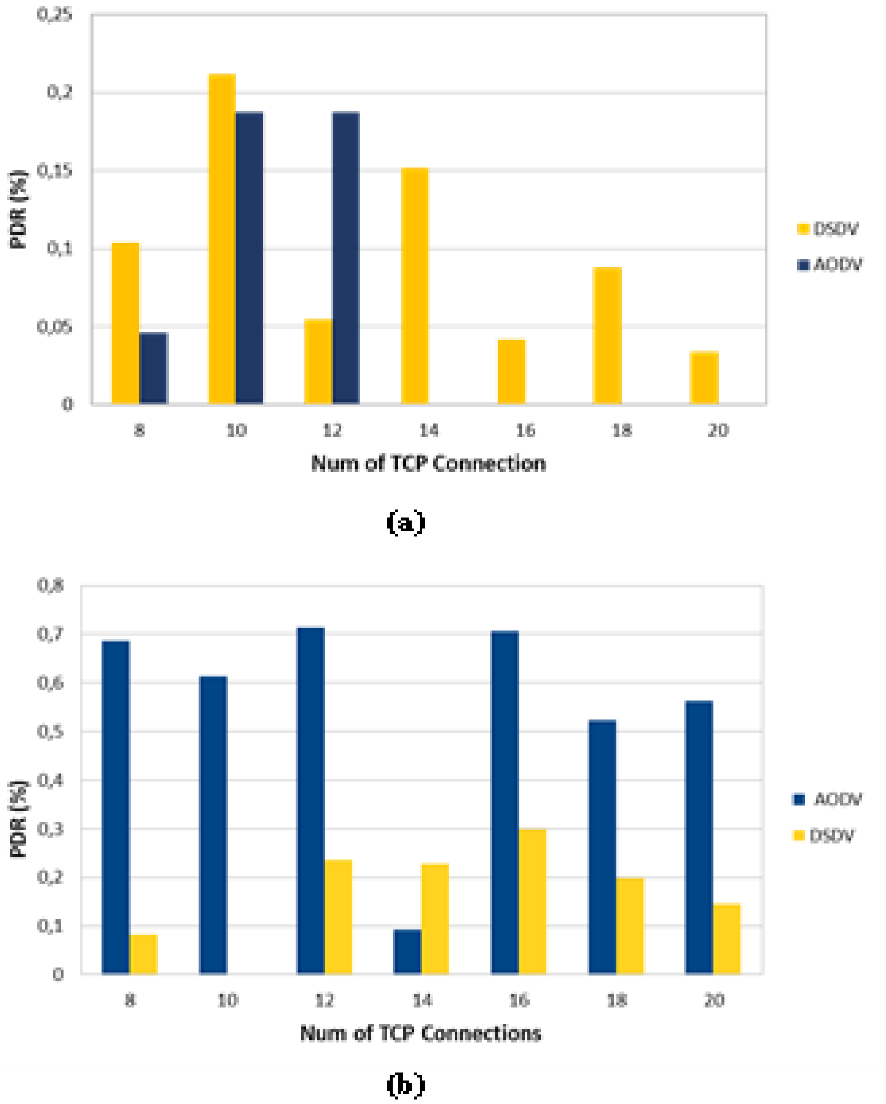


(a)

(b)

Figure 2.6 – *Throughput of* (*a*) *Vegas,* (*b*) *New Reno*

The graphs in Figure 2.6 show that the best throughput for both Routing Protocols is obtained with 10 TCP connections in the network for (a) Vegas and increasing the number of TCP connections gives better throughput with a best value at 16 TCP connection in the network for (b) New Reno. And as we can see, the average throughput takes great values while using DSDV Routing Protocol in front of AODV Routing Protocol for both TCP Vegas and TCP New Reno. In addition, we note that the rate of throughput is more important with TCP New Reno which takes more than 8 Kbit/s in front of a maximum value of 3.2 Kbit/s for TCP Vegas.

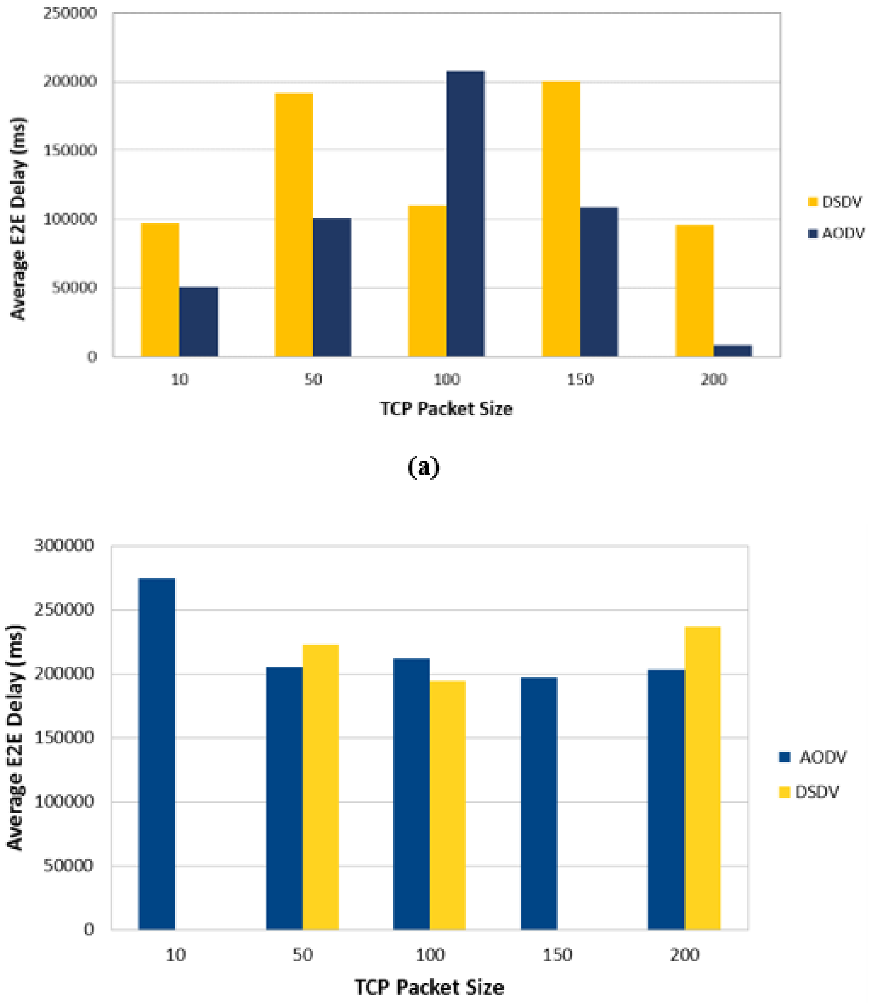


(a)

(b)

Figure 2.7 – *Packet Delivery Ratio of(a) Vegas, (b) New Reno*

The Delivery Ratio depicted in Figure 2.7 shows a large increase of delivered Packet with (a) TCP Vegas while using AODV when increasing the number of TCP connection from 8 to 12. However we can see that the best result for both protocols is obtained in 10 TCP connections, and goes to 12 TCP connections for AODV. With TCP New Reno, as depicted in (b) the PDR takes best results with 16 TCP Connections for both Routing Protocols,0,3% of Delivered Packet with DSDV and 0,7% with AODV.



(a)

(b)

Figure 2.8 – *Average End to End Delay (a) Vegas, (b) New Reno*

Figure 2.8 describes the average end to end delay where, using both Routing protocols give a similar progression in general while Number of TCP Connections increase but due to the reactive nature of AODV its average end to end delay exceeds.

2.4 Controlling Maximum Window of TCP NewReno in UWSNs

This section presents the first improvement of the TCP NewReno, we will focus on the control the Maximum TCP Window of NewReno in this environment.

2.4.1 Proposed approach

Our goal is to have better flow control management by appropriate transition between slow start and congestion avoidance phases. For this purpose, we will vary between 1 and 20 the values of the size of the upper limit of the congestion window with a step of 1. To implement the proposed solution we also defined some parameters used in(82). These parameters include the values of the Initial Window (IW) and the Maximum Segment Size of the Sender (SMSS) defined in Equation 2.7 which give us the generated setting parameters that will be used in the UWSNs simulations as described in the Pseudo Code 1. For the network model used to evaluate our improvement, we used the same network model architecture described in Figure 2.1 in the Network Model of the SubSection 2.3.2.

$$IW = \begin{cases} 4 * SMSS, & SMSS \leq 1095, \quad (MAXsegment = 4) \\ 3 * SMSS, & 1095 \leq SMSS \leq 2190, \quad (MAXsegment = 3) \\ 2 * SMSS, & SMMS > 2190, \quad (MAXsegment = 2) \end{cases} \tag{2.7}$$

Parameters Evaluations

In this work, we chose to evaluate the performance of TCP NewReno in UWSNs, we start by studying the behavior of its default parameter in this marine environment. The results of this initial simulations allowed us to determine the metrics that will be studied based on the trace file results. As first conclusion we found that the delivery of data is not stable and sometimes no packet is delivered. Another observation always with packet delivery, we have noted that in certain cases there is a high rate recording of duplication of the packets delivered, which leads us to define the studied metrics in this work as following: The number of packet delivery and the retransmission rate of the segments that were successfully received.

Packet Delivery: To study the performances of the TCP as transport protocol in the UWSNs begins by studying this metric since, when the number of received packets increases it affects the good function of the TCP with a good development of the adjustments of these parameters. This metric can also be calculated by counting the sequence numbers received during one simulation, since each packet is referenced by a sequence number specific to it. Thus it is concluded that the performance of TCP in terms of packet delivery depends on the number of packets received or the number of sequence numbers received in each simulation.

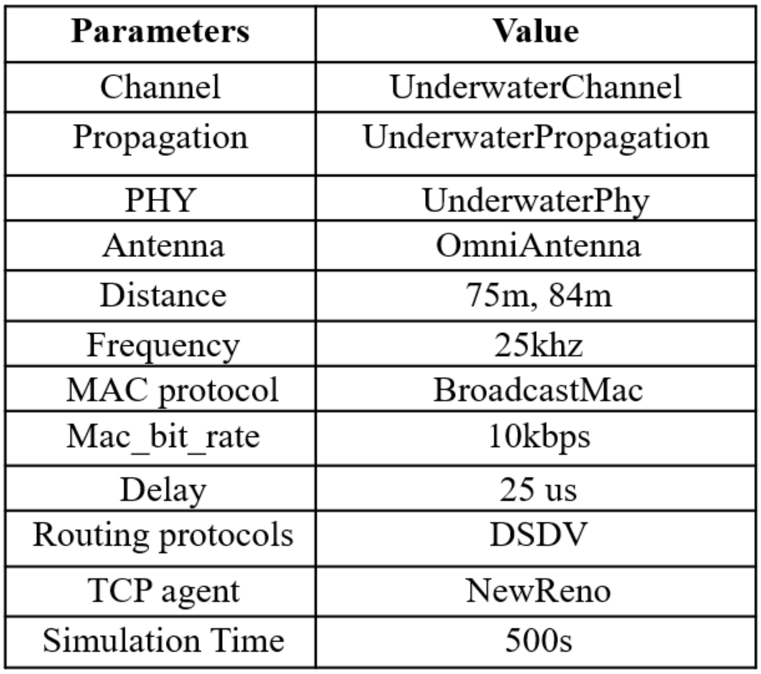
Retransmission Rate: UWSNs are known by the problem of energy efficiency, studying performance in terms of energy efficiency is feasible by the metric packet retransmission rate. It is the average number of times each packet has been received. The less value of this metric is, the better the performance of the TCP will be.

2.4.2 Simulation and Results

Simulations Environment

We chose to evaluate the reactive variant of TCP NewReno (70)since its derivative and widely deployed, and its performances were evaluated in different conditions similar to our needs. To carry out our simulations, we will use the reactive routing protocol DSDV (58) as studied in our previous work in(3).
To check our system we opted to use Aqua-sim (81), an NS2 based simulation software for UWSNs to analyze the simulations by measuring the QoS parameters. General simulation parameters are described in Table 2.2.

Table 2.2 – *Simulation parameters*



Parameters	Value
Channel	UnderwaterChannel
Propagation	UnderwaterPropagation
PHY	UnderwaterPhy
Antenna	OmniAntenna
Distance	75m, 84m
Frequency	25khz
MAC protocol	BroadcastMac
Mac_bit_rate	10kbps
Delay	25 us
Routing protocols	DSDV
TCP agent	NewReno
Simulation Time	500s

To study the TCP behavior, we have systematically changed the maximum size of windows and the value of the SMSS, and IW constant to indicate the impact of these parameters on the performance of the subsea network. In the underwater simulator tool, the parameters SMSS, IW, upper limit of the congestion window, are respectively named: 'packetSize_', 'windowInit_', 'window_'.

Initial simulations with the default values of TCP NewReno and the new ones were performed as showed in Table 2.2 which involves different simulation parameters, from the lowest network load to a very high level.

Table 2.3 – *Value of NewReno TCP parameters*

Parameter	Value
Window_	1-20
SMSS	1000,1500, 2500
IW	2*SMSS,3*SMSS, 4*SMSS
TCP NewReno sources	25,50

In order to better study the behavior of TCP NewReno, we performed different types of network scenarios, we tested and varied New Reno TCP number connections as shown in the TCP NewReno parameter's in Table 2.3. In first scenario we include 25 TCP nodes NewReno sources and in the second scenario we used 50 nodes sources of TCP NewReno, in order to demonstrate the effect of number TCP connections on the performance of the underwater network.

Results and Discussion

In this part simulations results and analysis are presented according to each scenarios.

First Scenario: In this first scenario, 25 transmitters of TCP NewReno are used in our submarine network, the control of the maximum window is analyzed with the variation of Packet size SMSS between, 2500, 1500 and 1000 bytes, the results present simulations with defaults parameters and those after the changes. In this scenario, as we can see for the Packet delivery, Figure 2.9 shows the relevance of controlling the congestion window limit. Indeed, the simulation with the TCP default parameters did not record any segment

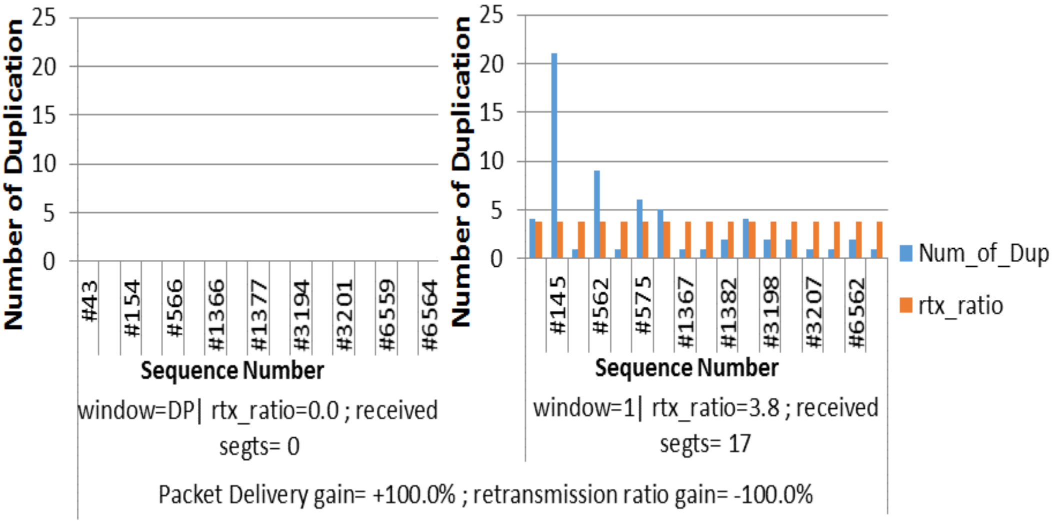


Figure 2.9 – *Performances before & after with SMSS= 1000 bytess*

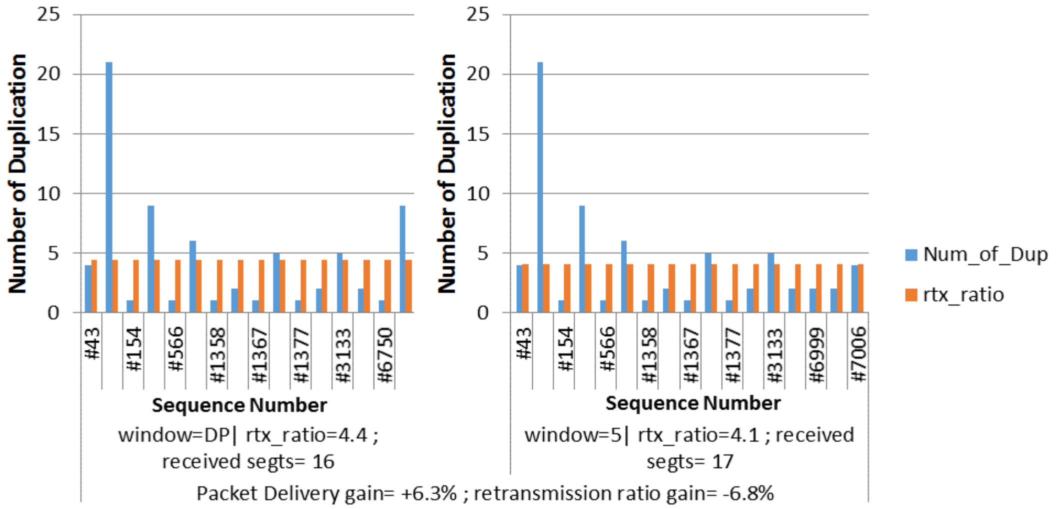


Figure 2.10 – *Performances before & after with SMSS= 1500 bytes*

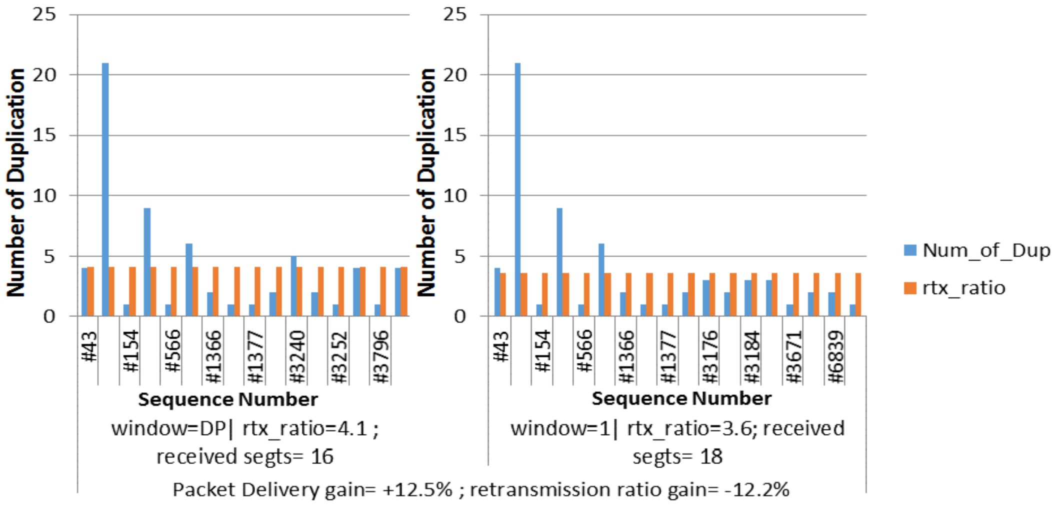


Figure 2.11 – *Performances before & after with SMSS= 2500 bytes*

reception. Conversely, the reception of segments is obtained for a limit fixed to 1 for this case. The same applies to the other two cases with improvements of +6.3% as Figure 2.10 shows, and +12.5% as presents Figure 2.11. It should be noted that the optimal improvements at this stage are achieved for small values of the window size upper bound between 1; 5. An explanation could be that the traffic that is related to the source nodes increases, which would cause more risk of collision or interference hence the need to inject less traffic at the same time. The principle of reducing retransmission rate, is achieved in all cases when it is applied with a Control of the upper window limit as follows: -100% in Figure 2.9, -6.8% in Figure 2.10 and -12.2% in Figure 2.11.

Second Scenario: The results of the simulations in this scenario concerning the effect of maximum window control in a network that has 50 nodes of TCP NewReno Source, and it refers to simulations with default parameters against those after changes with the variation of Packet size SMSS between, 2500, 1500 and 1000 bytes.

Figure 2.13 and Figure2.14 show similarities concerning the Packet delivery. Indeed, no reception is recorded during simulation before improvement. However, once an adequate size of the upper window limit size is found, the packet delivery starts to be observed. There are 12 and 16 packets respectively for 1500 bytes and 1000 bytes cases. The 2500 bytes case shows a shift from 10 to 9 packets or -10% (cf. Figure 2.12), which represents a lower disadvantage given the perceived gain when taking into account the other metric. The relatively small values of the maximum size of the congestion window for all

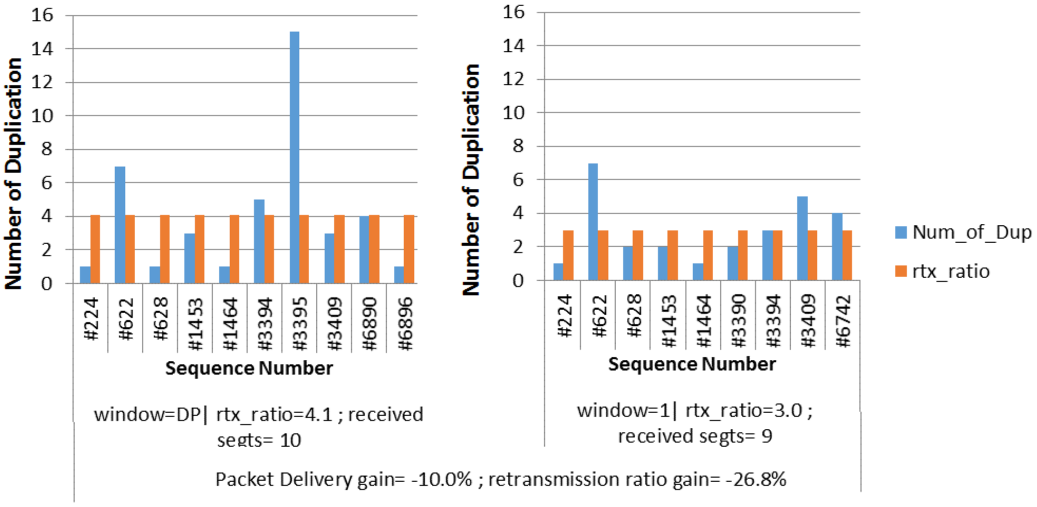


Figure 2.12 – *Performances before & after with SMSS= 1000 bytess*

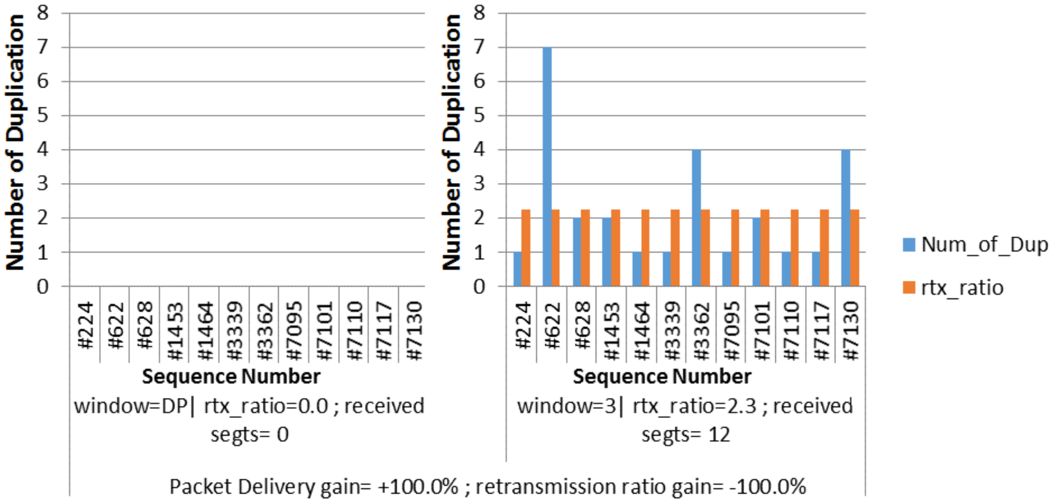


Figure 2.13 – *Performances before & after with SMSS= 1500 bytes*

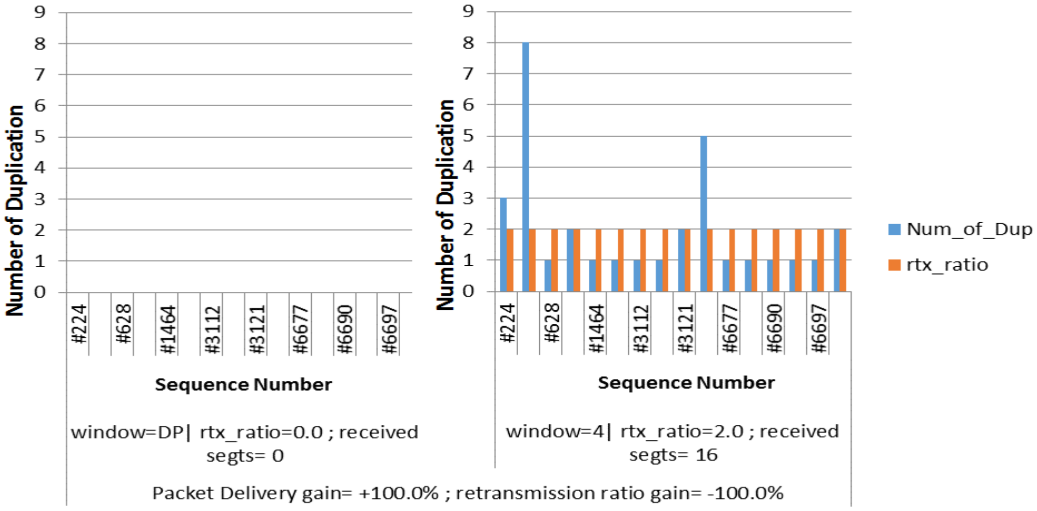


Figure 2.14 – *Performances before & after with SMSS= 2500 bytes*

cases show consistency. This could be the fact that this configuration, in addition to the disadvantageous characteristics of the channel, is conducive to more collision, interference, long queues caused by the volume of traffic generated. Thus, avoiding unnecessary packet loss could find its meaning in the use of congestion window limited to relatively small values. Only the 2500 bytes case shows improvements of -26.8% of Retransmission ratio this is what Figure 2.12 depicts where window value is equal to 1.

2.5 Adapting the appropriate RTT timeout of TCPNewReno in Submarine Communication Networks

In this section, we present another improvement of NewReno TCP while adjusting the RTT time Out to this TCP in UWSNs.

2.5.1 Proposed work

Our study used the underwater architecture described in Figure 2.1 in the Network Model of the SubSection 2.3.2. This architecture is based on 100 static nodes emerged and fixed at the bottom of the water where we vary the number of TCP NewReno transmitters, in the middle of this cluster we have an underwater sink for the collection of packets sent.

The characteristics of the channel in the UWSN differ particularly in terms of propagation time that varies in a manner that affects the RTT timeout. As a result, the estimation of the Round-Trip Time (RTT) waiting time becomes a necessity to improve the management of the flow control.
A rough calculation of the value of RTT timeout is done by adjusting the 'rtxcur_init' pa-

rameter which initializes the 't_rtxcur' parameter. This value varies between 1 and 20 with a step size of 1 by taking into account the directions proposed in (83) which includes the introduction of the value of the initial window (IW) and the maximum size of the sender segment (SMSS) as presented in equation 2.8.

$$IW = \begin{cases} 4 * SMSS, & SMSS \leq 1095, \quad (MAXsegment = 4) \\ 3 * SMSS, & 1095 \leq SMSS \leq 2190, \quad (MAXsegment = 3) \\ 2 * SMSS, & SMMS > 2190, \quad (MAXsegment = 2) \end{cases} \tag{2.8}$$

For the value of the maximum window, it was guided by the result found in our contribution(84) where several studies and simulations have allowed us to find the adapted values taking into account the nature of the underwater environment. To adjust these changes, we implemented the Pseudo Code 1 to modify the source code in Aqua-sim (81) tool on ns2 simulator network (85) coding on C++ language.

Algorithm 1 Adjusting of NewReno TCP parameters.

```
Input: $rtxcur\_init$: Parametter that initialise the round-trip time (RTT)
Input: $IW, SMSS$: Metric measurements
for <i=1; 20> do
    $rtxcur\_init \leftarrow i$
    if ($SMSS > 2190$ ) then
        $IW \leftarrow 2 * SMSS$
    if ($SMSS <= 2190$) then
        $IW \leftarrow 3 * SMSS$
    if ($SMSS <= 1095$) then
        $IW \leftarrow 4 * SMSS$
```

To evaluate the adapting value of these parameters we used two performance metrics which are the Packet Delivery and the Packet Retransmission rate.

- **Packet Delivery:** It indicates the number of packets received in each communication or simulation. This metric defines the good functioning of the studied TCP. When the number of received packets increases one can say that the performances of the protocol TCP are good (84).

- **Retransmission rate:** It represents the average time of each packet has been received. For this metric the smaller its value, the better the performance of the TCP used (84).

2.5.2 Simulation and Resutls

Similation Setups

The study was conducted by running simulations using tools called Aqua-sim which is based on the NS2-30 network simulator(85). The effectiveness of Aqua-sim to simulate acoustic signal attenuation, packet collisions in underwater sensor networks and support three-dimensional deployment are discussed in (81). The simulations are designed to compare the performance of the behavior of the standard version of TCP NewReno with the adapted version after the change of the values of the study parameters in this work. We then, adapt changes by performing experiments in multiple scenarios with different parameter values to measure the performance of the new version of TCP NewReno in UWSNs.

The scenarios studied in this research aim to test the behavior of TCP NewReno after adapting the estimate of RTT in different environments. We chose to vary the number of TCP densities in the network in each scenario. The first scenario had 12 NewReno TCP transmitters and the second scenario had 25 NewReno TCP sources. In each scenario we used the implementation as explains the Flow chart below in Figure 2.15.

The simulation parameters of this work is summarized in Table 2.4. It describes the general simulation parameters on underwater environment, which consist of some parameters and values such as: the type of the used Channel, the type of the Propagation and kind of Antenna and so on. Based on previous work (3) ,we choose to use DSDV (58) as Routing protocol. Regarding the values used for New Reno TCP parameters, Table 2.5 gives us an idea about the values that were used in this research.

Results & Discussion

The results of the first scenario concern a network that contains 12 sources of TCP NewReno. To estimate the delay time RTT we will start by setting the parameter 'rtxcur_init' from 0 to 20 which initializes the parameter 't_rtxcur' and we vary the packet size between 1000, 1500 and 2500 bytes Figure2.16. 5 and Figure2.18 . 7 describe the results before and after adapting the appropriate 'rtxcur_init' value of TCP NewReno parameters. As we can see for the Packet Delivery, those two cases show a relative increase in gain +11.1% and +25.0% respectively.

On the other hand, the case while using a Packet Size of 1500 bytes the results depicted in Figure 2.17 shows a decrease of -12.5%, which may seem to be a disadvantage because the possibility of improving the number of delivered packets (up to 9) is possible for 'rtxcur_init' values at 8. However, the disadvantage is the retransmission rate which proves to be high.

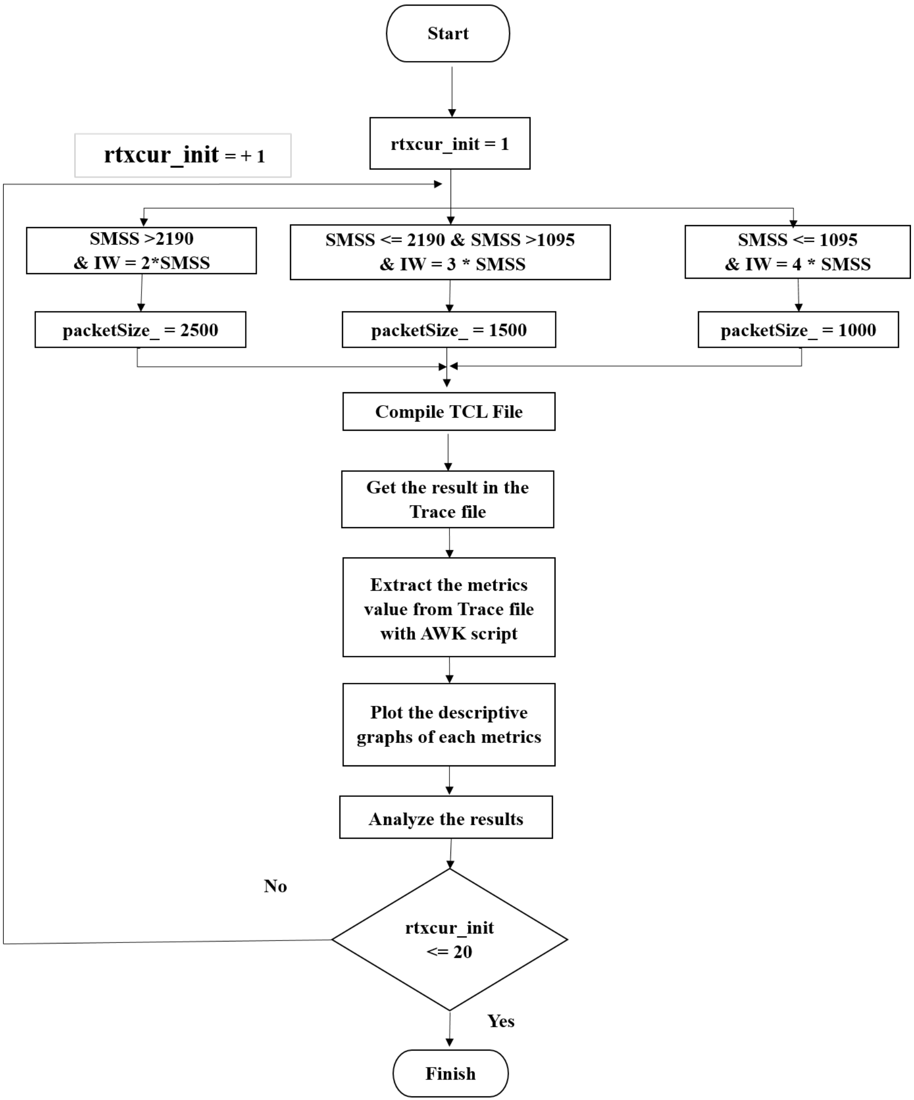

Start
rtxcur_init = 1
rtxcur_init = + 1
SMSS >2190 & IW = 2*SMSS
SMSS <= 2190 & SMSS >1095 & IW = 3 * SMSS
SMSS <= 1095 & IW = 4 * SMSS
packetSize_ = 2500
packetSize_ = 1500
packetSize_ = 1000
Compile TCL File
Get the result in the Trace file
Extract the metrics value from Trace file with AWK script
Plot the descriptive graphs of each metrics
Analyze the results
No
rtxcur_init <= 20
Yes
Finish

Figure 2.15 – *Flow Chart of the implementation*

Moreover, taking into consideration the 'rtxcur_init' parameter it presents divergent results. When the 1500 bytes case shows an improvement with a change from 1.8 to 1.6 or -11.1% gain in Retransmission Figure 2.17 6, the other two cases show improvements of +55% and +20.8% respectively illustrated by Figure 2.16 and Figure2.18. These situations show the difficulty of correctly estimating the value of 'rtxcur_init' in this configuration.

The second Scenario uses 25 nodes sources of TCP NewReno. The improvement of the packet delivery is observed for the appropriate value of 'rtxcur_init' as well in this second scenario. We note an evolution from 18 to 20 which is a gain of +11.1% as shown in Figure 2.19, from 17 to 23 which is a gain of +35.3% as described in Figure 2.20 and from 17 to 27 which represents a gain of +58.8% as depicted in Figure 2.21.

Table 2.4 – *Simulation Parameters*

Parameters	Value
Channel	UnderwaterChannel
Propagation	UnderwaterPropagation
PHY	UnderwaterPhy
Antenna	OmniAntenna
Distance	75m, 84m
Frequency	25khz
MAC protocol	BroadcastMac
Mac_bit_rate	10kbps
Delay	25 us
Routing protocols	DSDV
TCP agent	NewReno
Simulation Time	500s

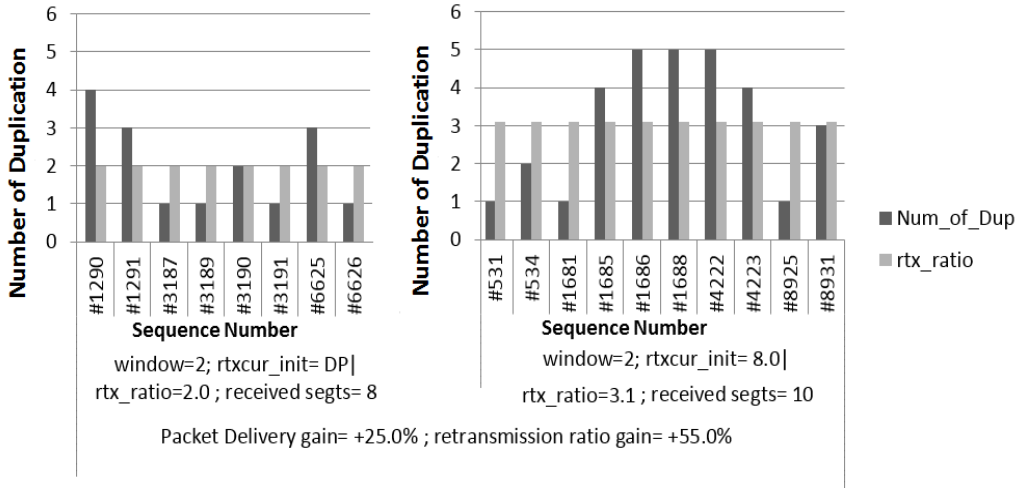


Figure 2.16 – *Performances before & after with SMSS= 1000 bytess*

The value of 'rtxcur_init' is uniformly 1sec for the last two cases (cf. Figure 2.20 and Figure 2.21), while for the 1000 bytes case this value is 15 sec. However, in the 1000 bytes case, the best performance with a small value of this parameter is reached at 2 sec with a number of received segments is remaining at 18 but the reverse is the retransmission ratio which increases considerably.

The principle of reducing the retransmission rate is achieved in all cases when it is

Table 2.5 – *Value of NewReno TCP Parameters*

Parameters	Value
Window_	1 2 5 12 16
SMSS	1000 1500 2500
IW	2*SMSS 3*SMSS 4*SMSS
Number of TCP NewReno sources	12 25

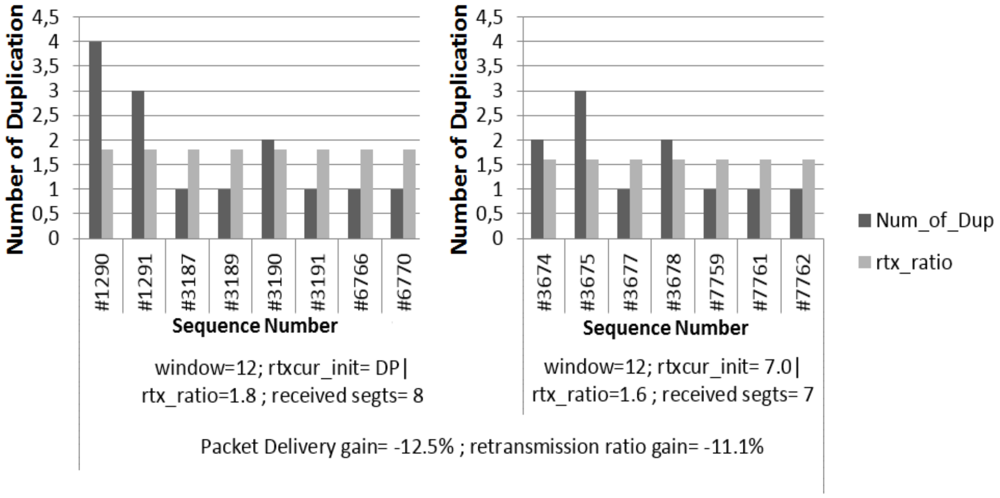


Figure 2.17 – *Performances before & after with SMSS= 1500 bytes*

applied, adjustment the value of 'rtxcur_init' as follows: -2.6% in Figure 2.21 and -4.9% in Figure 2.20. However, an exceptional increasing by +10% is observed for the 1000 bytes case in Figure 2.19.

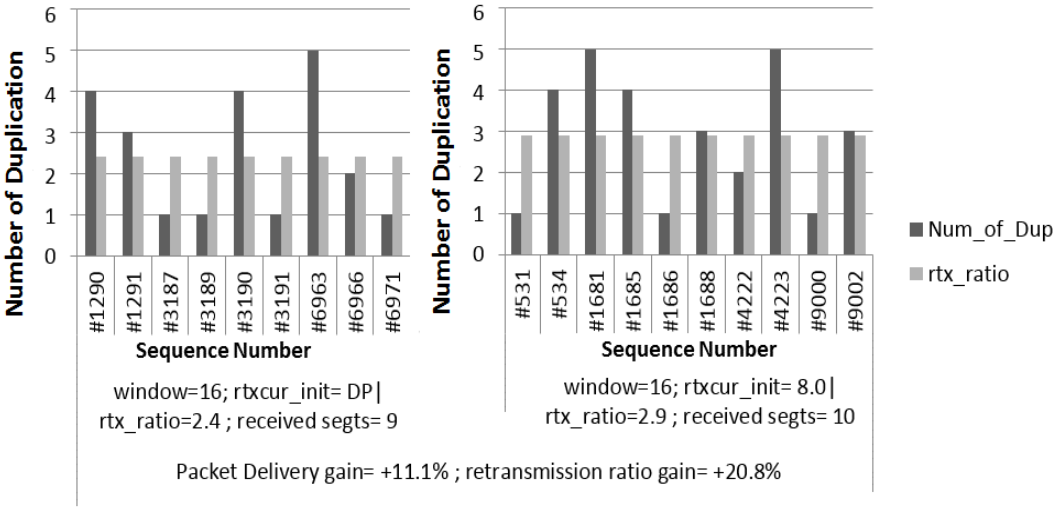


Figure 2.18 – *Performances before & after with SMSS= 2500 bytes*

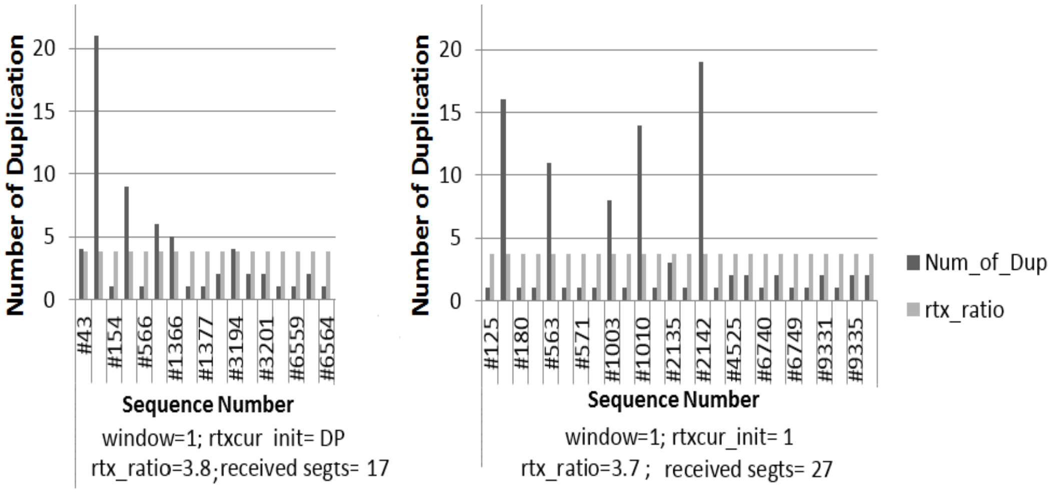


Figure 2.19 – *Performances before & after with SMSS= 1000 bytess*

2.6 TCP U-NewReno: a Transmission Control Protocol to Enhance Transmission Communication in UWSNs

2.6.1 The Proposed Protocol: TCP Underwater NewReno (U-NewReno)

Preliminaries

It is in layer 4 of the transmission protocols where the TCP guarantees a reliable delivery of the packets while keeping the order of these packets with an end-to-end, connection-oriented transmission for all network applications by using the IP service (86). This means ensuring a connection establishment where TCP uses the "three-way negotiation" mecha-

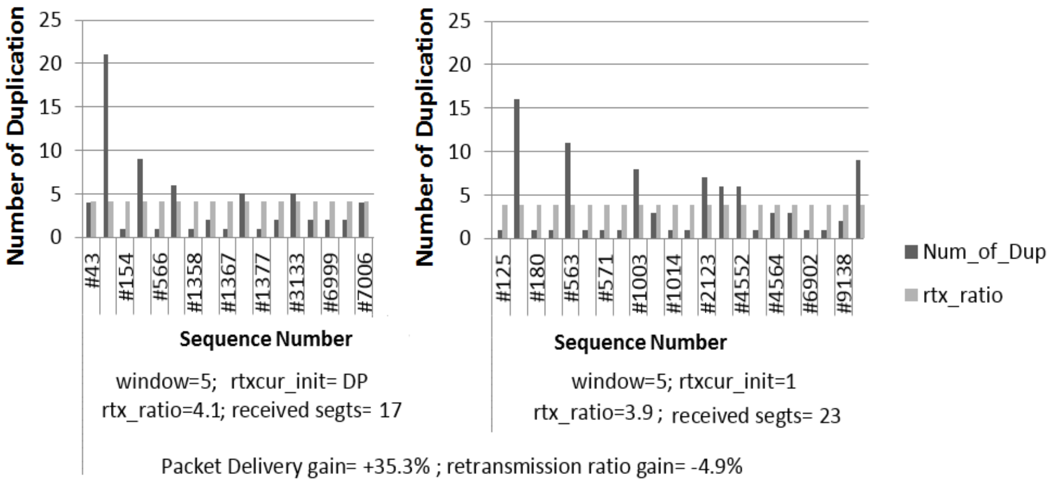


Figure 2.20 – *Performances before & after with SMSS= 1500 bytes*

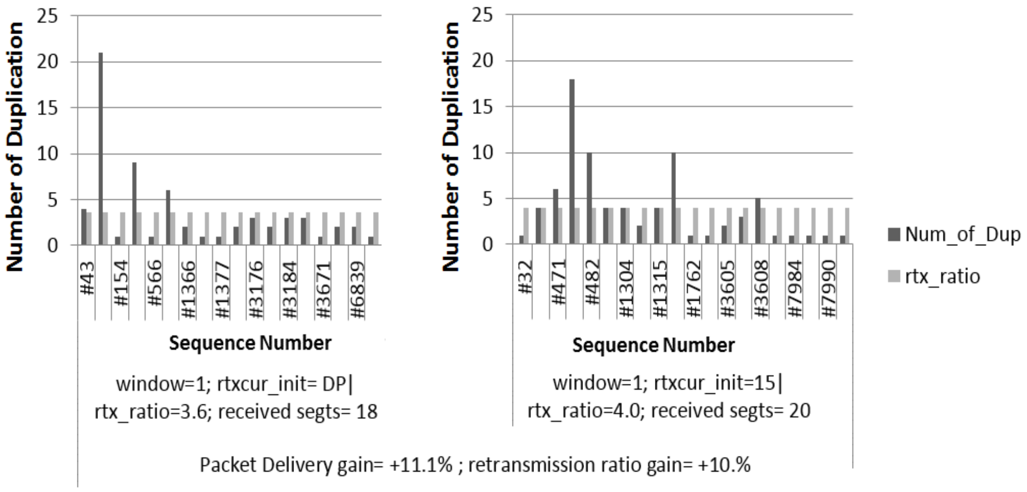


Figure 2.21 – *Performances before & after with SMSS= 2500 bytes*

nism as explains the Figure 2.22 and the close of TCP connection as depicted in Figure 2.23.

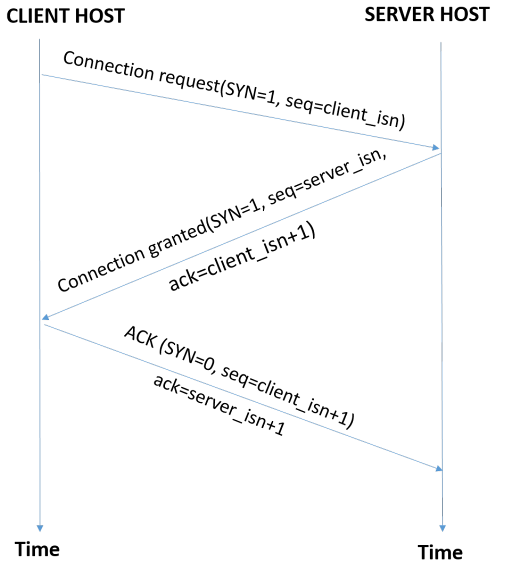

CLIENT HOST
SERVER HOST
Connection request(SYN=1, seq=client_isn)
Connection granted(SYN=1, seq=server_isn,
ack=client_isn+1)
ACK (SYN=0, seq=client_isn+1)
ack=server_isn+1
Time
Time

Figure 2.22 – *Three-way handshake mechanism*

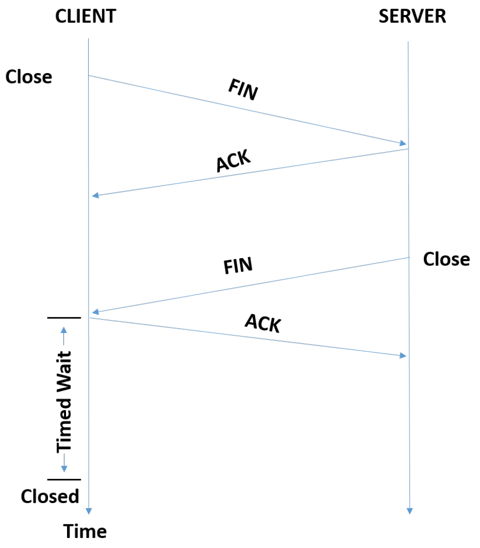


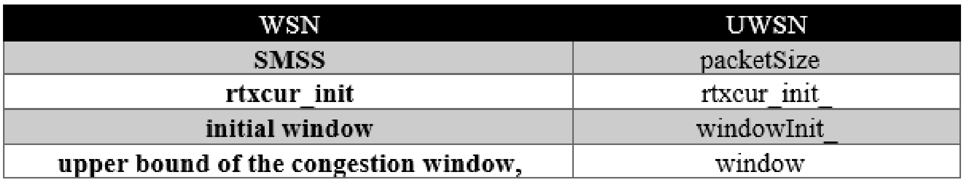
Figure 2.23 – *Closing the TCP connection*

As we know, New Reno TCP is an extended version of the known Reno Transmission communication protocol based on what was said in RFC 3782(53).In the fast recovery phase TCP New Reno is identified by its enhanced retransmission process, which allows it to recover several packet losses in a data window. The appearance of a partial ACK occurs if multiple packets are lost, so a partial ACK does not recognize all pending packets at the beginning of a fast recovery (the sender quickly leaves the recovery station) in addition, sender must wait until the end of the waiting period.

In the Slow-Start state is a mechanism that allows you to increase (exponentially) the flow of the source. The Slow-Start phase ends with a generation of lost packets by saturating the reception buffers of the source. As soon as a first loss is detected, the source switches to the congestion avoidance mode to smoothly increase its flow rate.

In the Fast retransmission, this state is characterized by the retransmission of the lost packet and assigning the max value of (FlightSize / 2, 2) to slow-start threshold (ssthresh) where FlightSize refer to the number of unacknowledged segments transmitted already. Thus, ssthresh will decrease to +3 and will be assigned to the Congestion Window (cwnd) then the sender gets into state of fast recovery. After receiving several duplicate ACKs (typically three), the sender infers that there has been a loss and enters to the Fast Recovery phase, where NewReno retransmits the last unacknowledged segment, puts ssthresh at half the congestion window's size, and reduces also its cwnd by half.

Table 2.6 – *Nomenclature of TCP parameters in WSN and UWSN*



WSN	UWSN
SMSS	packetSize
rtxcur_init	rtxcur_init_
initial window	windowInit_
upper bound of the congestion window,	window

Motivation and objective

Our motivation comes from our two previous works (84), (87) in which we proved the feasibility of varying the CWND parameter to improve the TCP in an acoustic transmission (84) and we managed to adapt the initial value of RTT parameter in our second work (87). Applying the results of these two works simultaneously for a more adequate adaptation is the main objective in this work. As a result, we come out with U-NewReno an adaptive underwater version of New Reno TCP to enhance the performance of the communication of transmission in the underwater wireless sensors network.

Implementation of the algorithm

To implement our algorithm, we used equation 2.9 which determines the value of the initial window (IW) in relation to the maximum size of the transmitter segment (SMSS) (83), in order to firstly try to better manage the flow control by tuning the appropriate transition from slow-start and congestion avoidance mode, and secondly minimize the abusive RTT timeout and fix its appropriate initial value. In submarine vocabulary, for acoustic signals the parameters of the TCP used are presented in the following Table.2.6 Figure 2.24 describes the flow chart of the simulation with the adjustment of TCP parameters in order to implement our new Parameters were set up by following Algorithm.2 to adjust U-NewReno TCP parameters.

$$IW = \begin{cases} 4 * SMSS, & SMSS \leq 1095, \quad (MAXsegment = 4) \\ 3 * SMSS, & 1095 \leq SMSS \leq 2190, \quad (MAXsegment = 3) \\ 2 * SMSS, & SMMS > 2190, \quad (MAXsegment = 2) \end{cases} \tag{2.9}$$

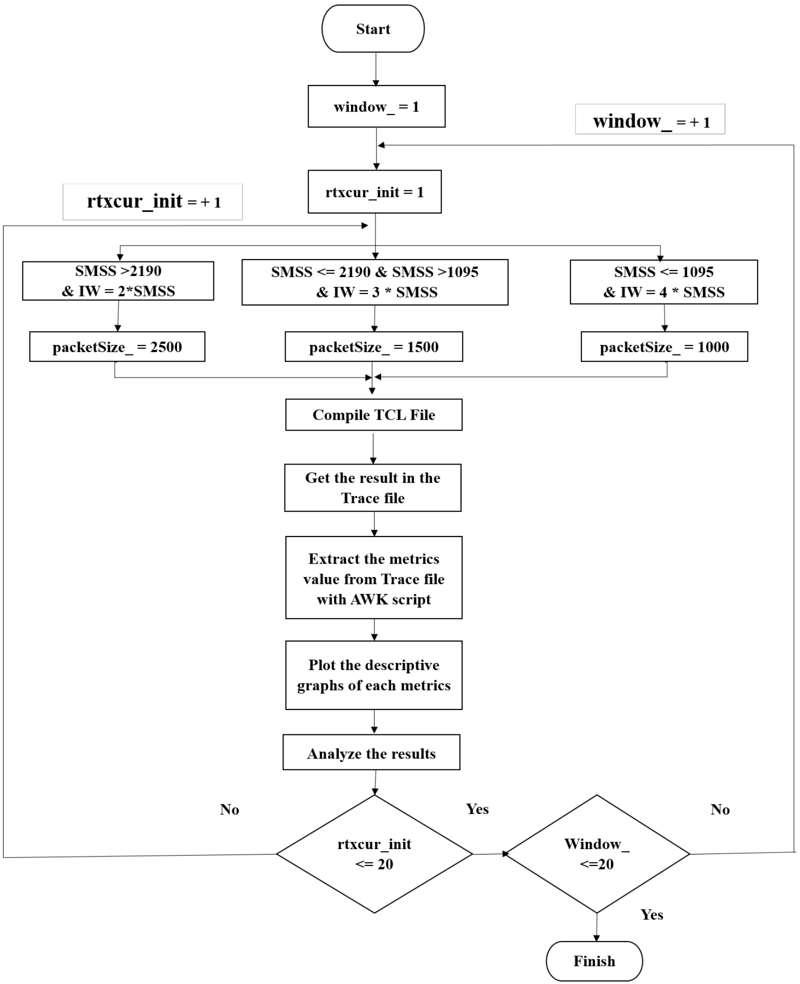


Figure 2.24 – *Flow chart of the simulation with the adjustment of TCP parameters*

2.6.2 Performance and Simulation Setup

Performance and general design

The general design to test the implementation of U-NewReno TCP as exposed in Figure 2.1 in the Network Model of the SubSection 2.3.2. Where the transmitted nodes will sens TCP NewReno packets to the sink station.

To access the channel, we use the predefined protocols in the Aqua-sim simulator and also for the physical layer, we use the underwaterPhy module end as Routing protocol we used DSDV.

To evaluate our work, we compare U-NewReno with the initial NewReno. we used 3 scenarios to analyze our protocol, each scenario contains different number of nodes, scenario A with 12 nodes, scenario b contains 25nodes, and finally scenario C with 50 nodes. On the other hand, three data packet lengths of 1000, 2000 and 2500 bytes were analyzed in each scenario. The execution time of the simulations was 300 seconds. Table 2.7 describes

Algorithm 2 Adjusting of the U_NewReno TCP parameters.

```
Input: window_: Upper limit of the congesion window
Input: rtxcur_init: Parametter that initialise the round-trip time (RTT)
Input: $IW, SMSS$: Metric measurements
for <i=1; 20> do
    window_ ← i
    for <j=1; 20 > do
        rtxcur_init ← j
        if ($SMSS > 2190$ ) & ($IW = 2* SMSS$) then
            packetSize_ ← 2500
        if ($SMSS <= 2190$) & ($SMSS > 1095$) & ($IW = 3 * SMSS$) then
            packetSize_ ← 1500
        if ($SMSS <= 1095$) & ($IW = 4 * SMSS$) then
            packetSize_ ← 1000
```

general simulation parameters and Table 2.8 describes the simulation parameters of U-NewReno TCP.

Evaluated metrics

In what follows we present the metrics measured to evaluate the performance of our U-NewReno TCP which are the number of packet delivery and the retransmission rate of segments received.

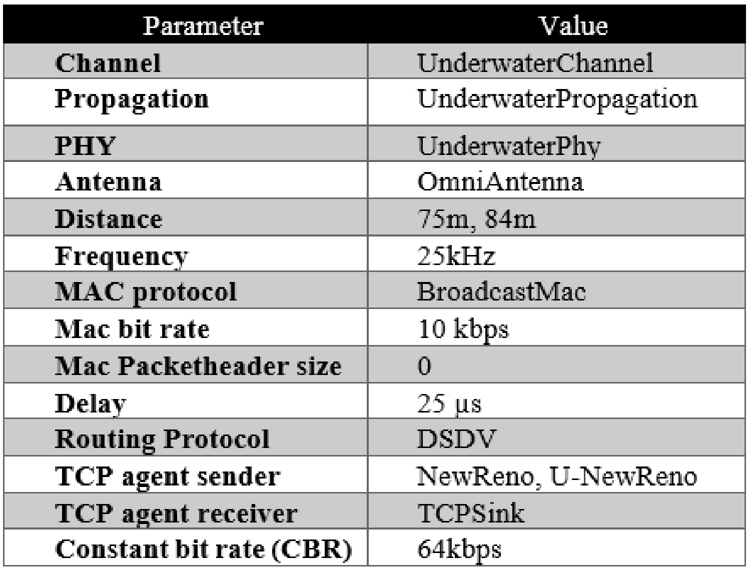
Packet delivery: this parameter helps us to follow the influence of the changes made on the number of transmitted data, in other words the number of segments received. Consequently, the more packets are delivered, the better the performance of our TCP.

Retransmission ratio: a very important metric in submarine communication since it informs us about the number of times a segment is received. In the same time, it gives information on lost energy which is one of the major problems in the marine environment. This value should be as low as possible.

Simulation tool

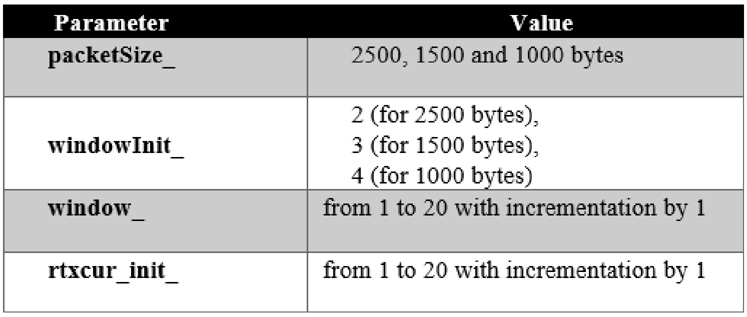
We used Aqua-sim (81) as a simulation tools which is based on NS2 to simulate and evaluate our proposed TCP U-NewReno. The advantage of using Aqua-sim consists in the presence of a simulation package which works in parallel with the CMU wireless package supporting as well NS2 which allows the independence of the wireless package. Currently, it is composed by 4 files: The Underwater Common, the Underwater Mac, the Underwater Routing and the Underwater tcl. We find that Aqua-Sim has the same object orientation of its structure as that of NS2, in addition the Network implementation is also done by C ++ classes as well as NS2(88).

Table 2.7 – *Description of essential simulation parameters*



Parameter	Value
Channel	UnderwaterChannel
Propagation	UnderwaterPropagation
PHY	UnderwaterPhy
Antenna	OmniAntenna
Distance	75m, 84m
Frequency	25kHz
MAC protocol	BroadcastMac
Mac bit rate	10 kbps
Mac Packetheader size	0
Delay	25 µs
Routing Protocol	DSDV
TCP agent sender	NewReno, U-NewReno
TCP agent receiver	TCPSink
Constant bit rate (CBR)	64kbps

Table 2.8 – *Simulation parameters of U-NewReno TCP*



Parameter	Value
packetSize_	2500, 1500 and 1000 bytes
windowInit_	2 (for 2500 bytes), 3 (for 1500 bytes), 4 (for 1000 bytes)
window_	from 1 to 20 with incrementation by 1
rtxcur_init_	from 1 to 20 with incrementation by 1

2.6.3 Results and analysis

In this section, the result of the simulations in different scenarios are presented.

Results of scenario (A):

The ideal objective to evaluate retransmission rate is to find a diminution of this factor in each different case. A goal well realized in the three cases described in Figure 2.25, Figure2.26 and Figure2.27 which they show that using the TCP U-NewReno can manage to reduce the rate of retransmission of segments. For the number of segments delivered, it is also showing an increased number in favor of U-NewReno for the 2 cases in Figure2.25 and Figure2.27, while it remains intact in the case shown in Figure 2.26

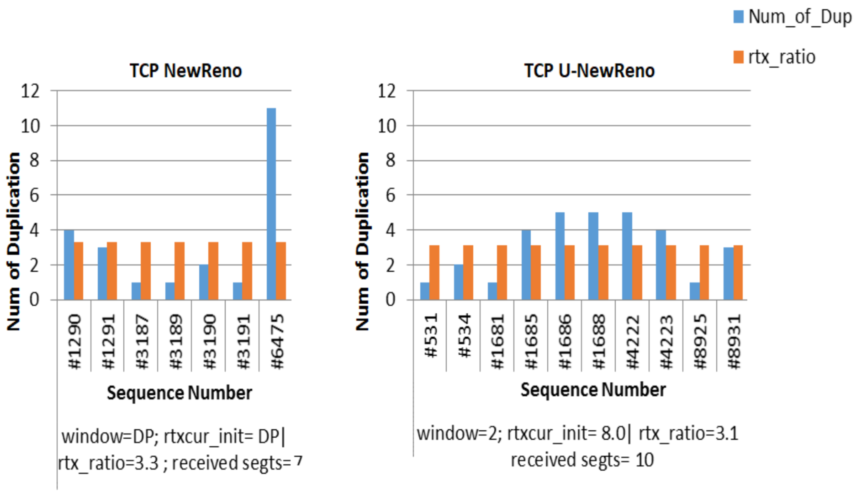


Figure 2.25 – *Performances of TCP NewReno vs.TCP U-NewReno with SMSS=2500 bytes*

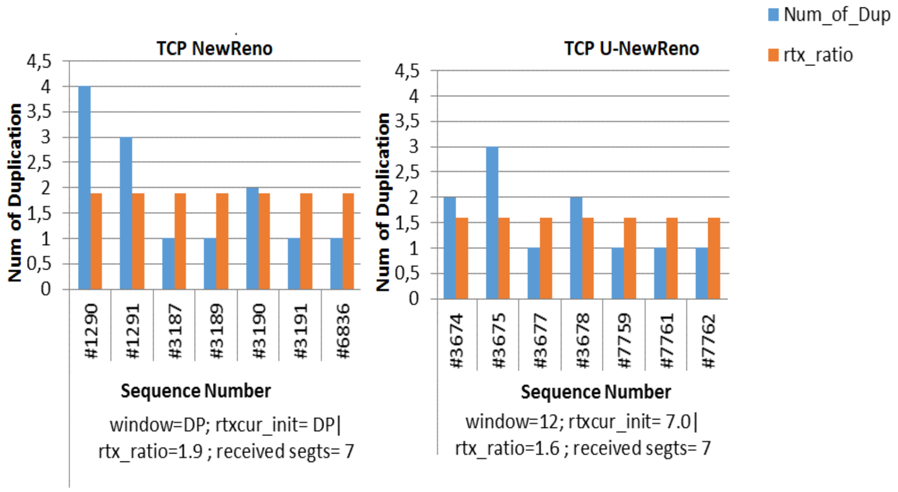


Figure 2.26 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=1500 bytes*

Results of scenario (B):

In this second scenario, it is also apparent that the performance of U-NewReno exceeds those of NewReno especially in terms of number of segments received. With our TCP U-NewReno, we have 27 segments received while NewReno does not deliver any segments as shown in Figure 2.28. On the other hand, in Figure 2.29; NewReno received 16 segments but under U-NewReno we were able to reach 23 segments received with a decreasing of retransmission rate from 4.4 to 3.9 still in favor of U-NewReno. The same improvement is found in Figure 2.30 with 8 more segments for U-NewReno and low retransmission rate.

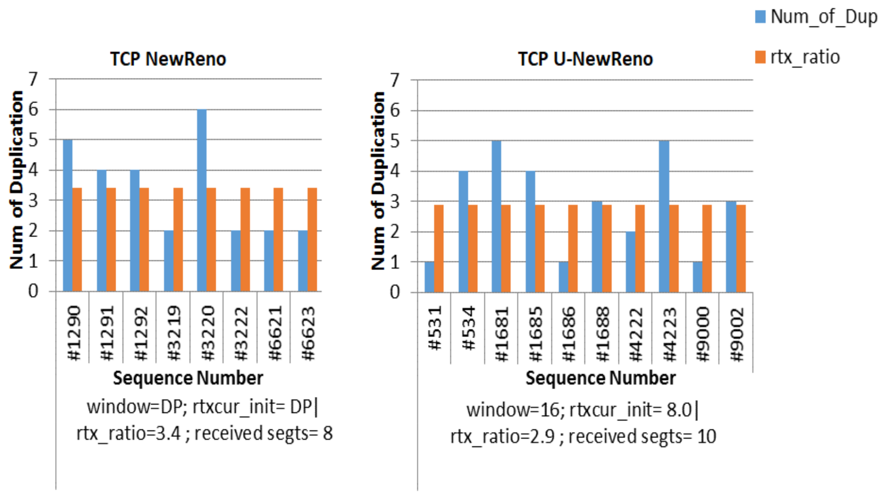


Figure 2.27 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=1000 bytes*

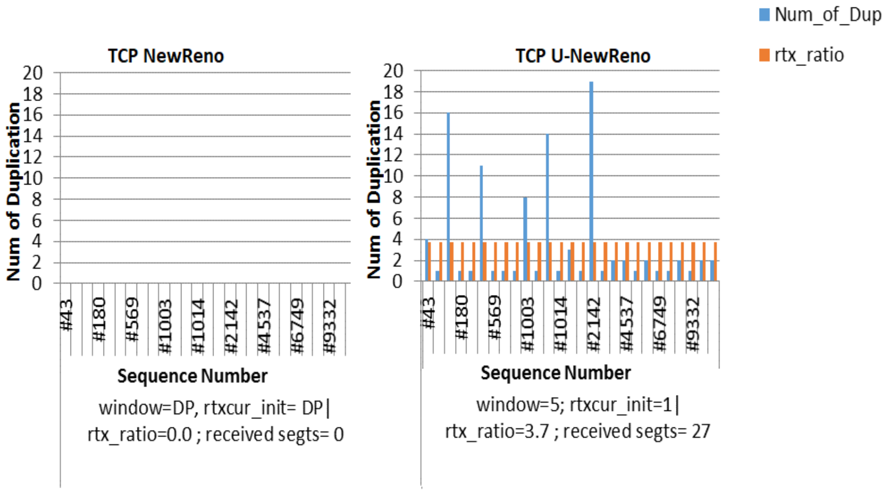


Figure 2.28 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=2500 bytes*

Results of scenario (C):

In this scenario, the adaptation of the two values with our algorithm to determine the optimal size of the segment and the good initial value of the RTT could give good results as indicated in Figures 2.31,2.3212, and 2.33. In Figure 2.31 it shows a very significant improvement in the number of segments delivered, 19 segments for U-NewReno with only 2.5 retransmission rates ahead of 4.1 for New Reno which receives only 10 segments. In Figure 2.32, it describes also the increase concerning the Delivery of the segments which was zero for NewReno and with U-NewReno we manage to receive 22 packets but with 3.5 as retransmission rate. On the other hand, Figure 2.33 also shows a Delivery of 22 segments for U-NewReno with 3.2 retransmission rates, whereas we do not receive any

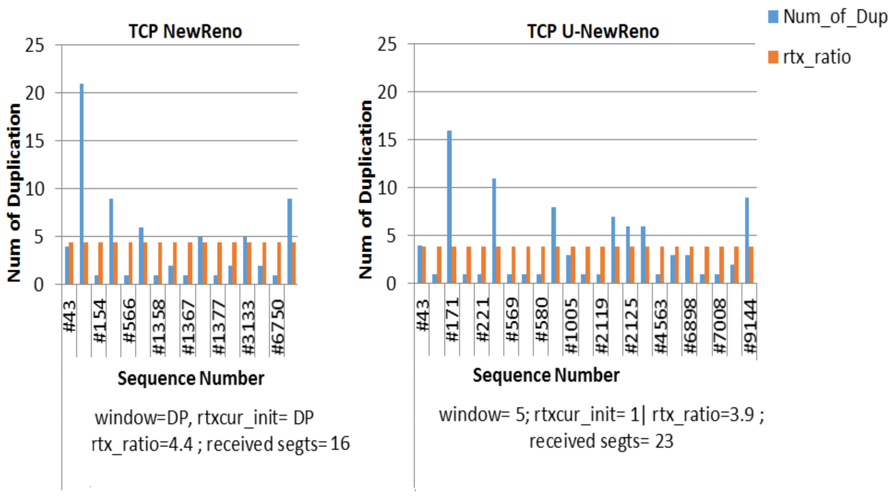


Figure 2.29 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=1500 bytes*

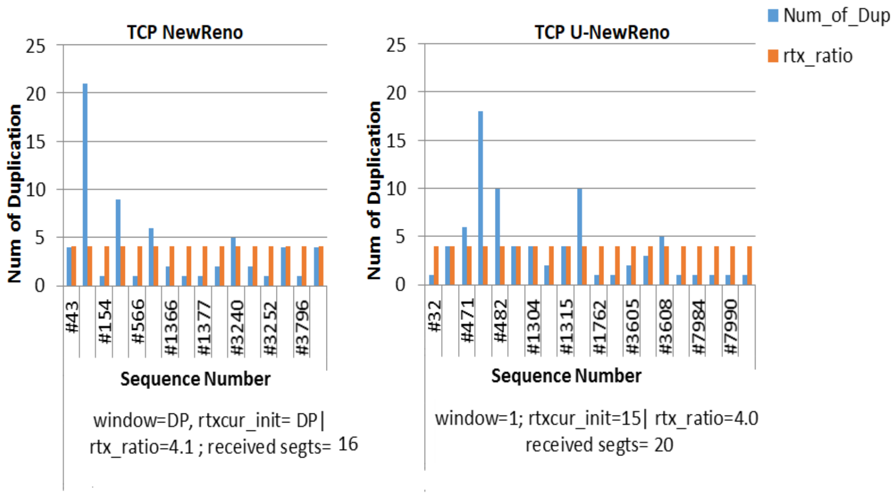


Figure 2.30 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=1000 bytes*

segments with NewReno.

To summarize, considering the two metrics evaluated. Our TCP U-NewReno was able to clearly reinforce the performance of NewReno. We can see that in each scenario, the results of U-NewReno exceed those of NewReno. This allows to U-NewReno to be well adapted to this aquatic environment. In addition, it can be noted that the number of source nodes influences the number of packets received from the relationship established with 12 source nodes and 25 source nodes.

-20 packets received against 10 for the 1000 bytes, namely a value factor of 2.

-20 packets received against 10 for the case 1500 bytes, namely a value factor of 3.3.

-27 packets received against 7 for the 1000 bytes, namely a value factor of 2.7.

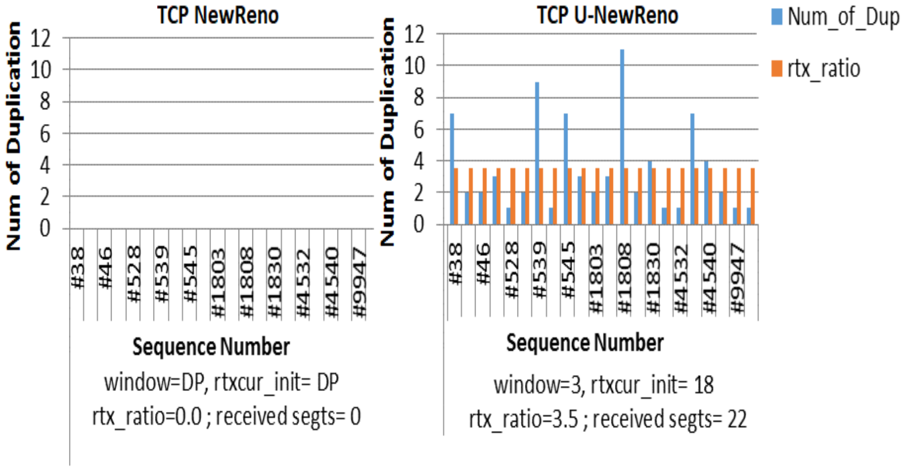


Figure 2.31 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=2500 bytes*

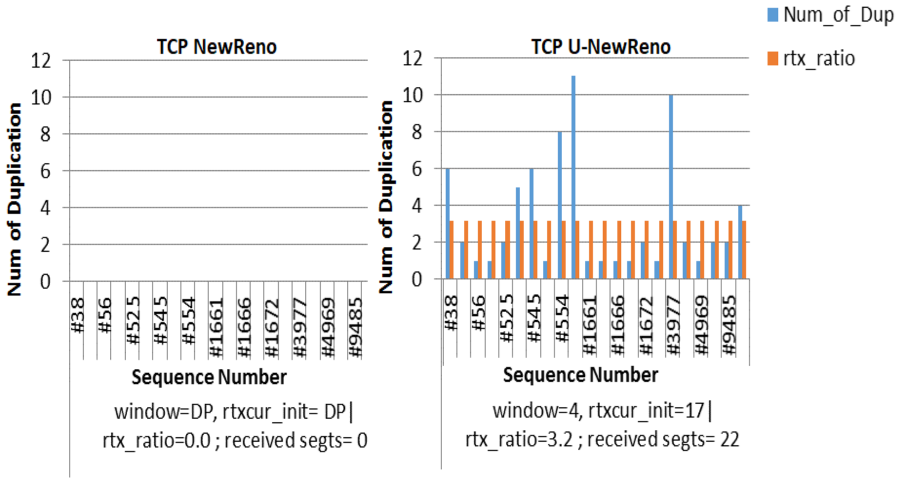


Figure 2.32 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=1500 bytes*

But, if we look at the cases of 50 source nodes and 25, no relation is noted, and it shows that there is a decrease in the number of packets received from 22 to 19 for 1500 bytes and from 23 to 27 for the case of 2500 bytes. With 1000 bytes a slight improvement in the number of packets received is visible, 22 packets received for U-NewReno against 20 for NewReno. Finally, Figure 2.34 presents the most important results of the performance of U-NewReno in terms of packet deliveries and packet re-transmission rates compared to the normal NewReno TCP.

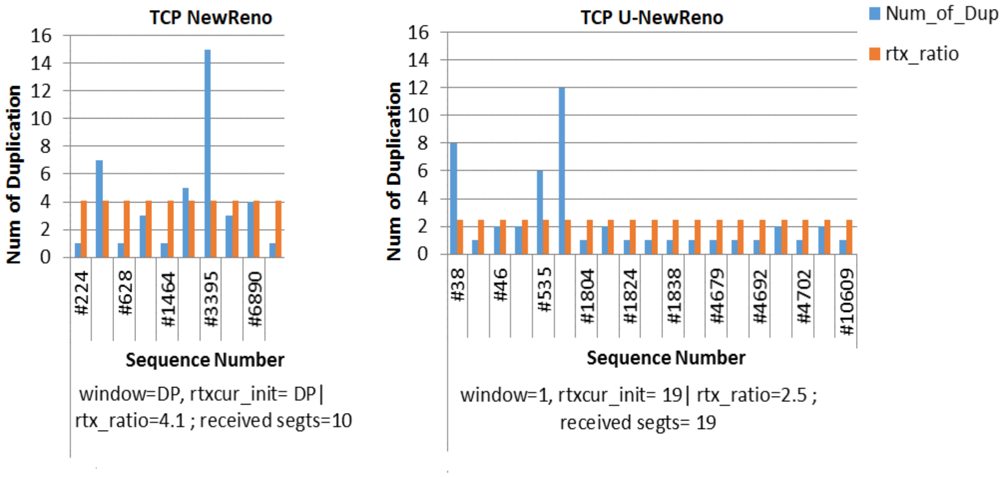


Figure 2.33 – *Performances of TCP NewReno vs. TCP U-NewReno with SMSS=1000 bytes*

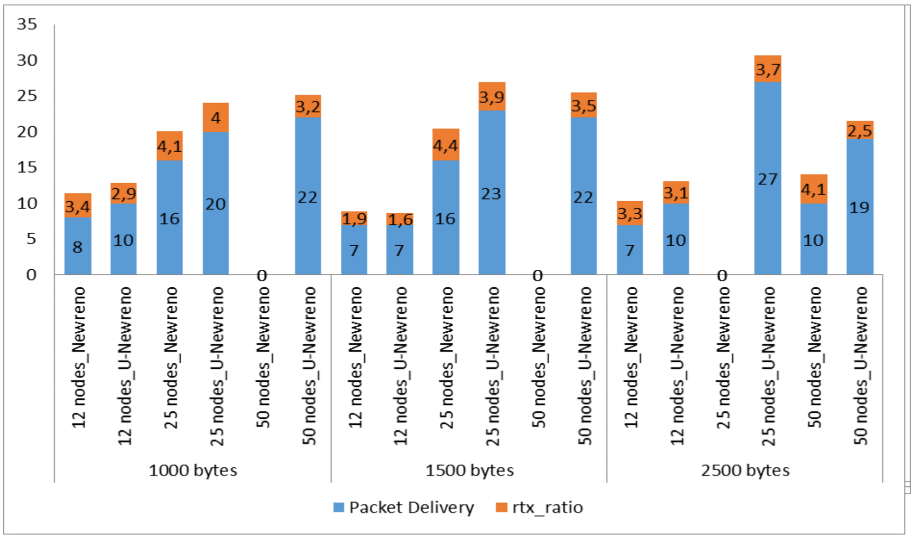


Figure 2.34 – *Summary of U-NewReno and NewReno performances relative to the SMSS size*

2.7 Conclusion

In this Chapter we investigate the performances of TCP in UWSNs, we start in section 2.3 by studying two different TCP congestion mechanisms: TCP Vegas and TCP New Reno in this network to evaluate the impact of varying TCP Packet Size and the density of TCP connections number in the network. In this work, we were able to determine the number of TCP nodes we also define the TCP Packet Size value that give good performance for TCP Vegas and for TCP New Reno in a sub-sea network. In addition, we found that the most reliable routing protocol namely DSDV with 10 TCP Vegas connections and a value of 100 for the Packet Size was greater than AODV and also for TCP New Reno, DSDV gives better results than AODV with 16 TCP New Reno Connections using 50 for its Packet Size. The results found with NewReno parameters are more reliable to transmit more information than those transmitted with Vegas.

In the next section 2.4, the behavior of TCP New Reno has been studied by controlling the maximum windows of the sender while varying the SMSS and the number of TCP sources. All improvements were obtained with a small value of window varied between 1 and 5. In addition to being able to determine the appropriate values of the maximum size of the congestion window, it is wise to determine the optimum number of source nodes with respect to the coverage in terms of nodes for the area to be monitored. This leads us to conclude that the number of source nodes has an impact on TCP performance in multi-hop UWSNs communication. Through this study, we have demonstrated that the selection of parameters plays an important role in improving the performance of TCP New Reno in a submarine network.

In Section 2.5 we were able to find an appropriate value of the 'rtxcur_init' used for initializing the RTT timeout to improve the performances of TCP NewReno in an UWSN. The simulations results, after adjusting the value of 'rtxcur_init' and then finding an appropriate RTT timeout, show that with this adaptation, TCP NewReno offers better performance in terms of packet received and retransmission packets compared to the results found while using the standard parameters of TCP NewReno in Underwater environment.

Finally, in Section 2.6 we have proposed U-NewReno a transmission communication protocol which uses an algorithm to determine suitable values for the maximum size of the congestion window and the RTT while specifying its initialization value in order to improve the performance of the traditional TCP NewReno in the underwater environment. Therefore, we can say that the performance of TCP in UWSNs is also influenced by the number of source nodes. The results of our simulations reveal that for the distribution of our topology the ideal number of source nodes which gives better coverage is 25 nodes which is equivalent to a quarter of total network nodes.
The next chapter will study another aspect of the challenge encountered in the Internet of Underwater Things environment which is; saving energy of the network through the use of Fuzzy logic theory in a routing protocol.

Chapter 3

A Pursuit Learning Solution to Underwater Communications

Contents

3.1 Introduction

In the last recent years, several applications of oceanographic monitoring have emerged. However, the feature of underwater environment has many challenges and obstacles in addition to those known in traditional Wireless Sensor Networks (WSNs). The use of acoustic signals in UWSNs to communicate is considered slower than radio signals used in WSNs which causes a longer propagation delay (89).

In terrestrial WSNs, establishing ad-hoc Mobile Networks (MANETs) constitute an important research topic. Similarly, in underwater environments, MANETs have several potential applications such as naval security and seabed mining operations. Currently, the adoption of MANETs in UWSN is very limited also because of the stochastic nature of the underwater acoustic environments that create difficulties for communication (90). When it comes to the routing part in underwater communications, much of the existing work in focuses on fixed rule algorithms for transmission that do not leverage the mobility of agents, or utilize the opportunity to alter their rule structure with mobility changes (48). Another problem that researchers face in submarine communication is the changing acoustic qualities of oceanic channel both seasonally and with local weather phenomena (90, 91). In addition, the ocean currents can cause the displacement of the sensor nodes, other factors are also present and affect the performance of the network sensors such as water temperature, the noise and also the attenuation of the signal without forgetting the factor of the 3D architecture deployment which makes overall these properties under a strong sensitivity (92).

In spite of the fact that these acoustic properties are inevitable, the purpose of this work is to prove that adaptive learning strategies can be used to modify the depth of deeply anchored Limited Mobility Agents (LMAs) in order to improve the communication of the underwater network. Nowadays, to change the depth of the nodes, the majority of sensor networks that allow a variable depth of anchorage to do this process often resort to human intervention, which makes this process very expensive and causes little movement of the sensor compared to the life of the sensor. Thus, this human intervention can be replaced by programmed engines that can be used for task and communication detection. To achieve this purpose and in order to improve the link stability in a network of underwater acoustic sensors, in this chapter we propose an approach to take advantage of the acoustic sound speed changes along the thermocline of an acoustic environment under-marine.

An adaptive strategy would not discriminate the cause of faults in the network, thus using a learning strategy will allow LMAs to choose and adapt to the best depth of operation which will avoid fault avoidance and also the collision in UWSN. The high cost of agents in UWSNs gives a prime motivation for using learning strategies to avoid faults in aquatic environment. Implementing marine networks with a system that adapts and evokes failures or link failures in submarine communication is very rare because of the high price that a UWSN agent can cost (51).

Our current work is inspired by the latter work (51) that proposes a Learning Automata (LA) that controls the mobility of thermocline using three actions: surface, dive or keep the same position. In this work, we only use two actions as we rather map the current problem to the Stochastic Point Location (SPL) problem (93). In SPL, only two directions are allowed and the SPL does not allow the learner to remain in the same position. Thus, instead of using the action that consists in staying in the same location, similar effect can be obtained by using SPL and oscillating back and forth around the optimal position (93). Therefore, introducing a third action: staying, for the learning algorithm is not necessary and can be avoided. Furthermore, in contrast to (51), the LA designed in this work uses the concept of pursuit LA that involves estimating the average reward of an action instead of merely using the feedback from the environment in isolation. In fact, pursuit LA exploits more effectively the information from the environment than traditional LA schemes that are myopic and use merely the last feedback from the environment instead of considering the whole history of the feedback.

According to the description of Partan et al. in (94), shadow areas, multipath interference and bubble cloud regions close to the surface are among several physical limitations of underwater acoustic communication. These physical properties not only cause binding failures in UWSNs, but are clearly characteristic of a varying duration of stochastic environment (95, 96). In addition, in order to improve communication in terrestrial telephone networks, Narendra et al. in (97, 98, 99) have relied on learning algorithm that takes into account the time changing congestion. In (97, 98, 99), in order to establish an adaptive rule routing in a stochastic demand telephone network, authors use a Mean Action Learning Automaton. The results of this work showed performance improvements over traditional fixed-rule routing. In the same direction, the work reported in (91, 100) proposes to give anchored nodes in the seabed the power to autonomously modify the operating depth of their locations which allows a network agent the ability to avoid collisions and defects caused by physical phenomena.

The reminder of this Chapter is organized as follows. In Section 3.2, we introduce the SPL problem. Section 3.3 gives the details of our LA based algorithm for controlling the mobility of thermocline sensors. Section 3.4 gives experimental results that confirm the convergence of the algorithm and shows its behavior. Section 3.5 concludes this chapter.

3.2 Legacy Stochastic Point Location Solutions

To place our work in the right perspective, we start this section by providing a brief review of the main concepts of the SPL problem as first introduced in (93).

We assume that there is a Learning Mechanism (LM) whose task is to determine the optimal value of some variable (or parameter), λ. We assume that there is an optimal choice for λ – an unknown value, say $\lambda^* \in [0, 1)$.

The question which we study here is that of learning λ^*. Although the mechanism does not know the value of λ^*, we assume that it has responses from an intelligent "Environment", Ξ, which is capable of informing it whether any value of λ is too small or too big.
To render the problem both meaningful and distinct from its deterministic version, we would like to emphasize that the response from this Environment is assumed "faulty." Thus, Ξ may tell us to increase λ when it should be decreased, and *vice versa*.

However, to render the problem tangible, in (93) the probability of receiving an intelligent response was assumed to be $p > 0.5$, in which case Ξ was said to be *Informative*.

Note that the quantity "p" reflects on the "effectiveness" of the Environment.

Thus, whenever the current $\lambda < \lambda^*$, the Environment correctly suggests that we increase λ with probability p. It simultaneously could have incorrectly recommended that we decrease λ with probability $(1 - p)$. The converse is true for $\lambda \geq \lambda^*$.

Oommen (93) pioneered the study of the SPL when he proposed and analyzed an algorithm that operates on a discretized search space while interacting with an informative Environment (i.e., $p > 0.5$).

The space in which the search is conducted is first sliced by subdividing the unit interval into N sub-intervals at the positions $\{0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}, 1\}$, where a larger value of N will ultimately imply a more accurate convergence to the unknown λ^*.

The algorithm then orchestrated a controlled random walk on this space. Whenever the mechanism was told to go to the right (or left), it obediently moved to the right (or left) by a single step (i.e., by $\frac{1}{N}$) in the discretized space. In spite of the Oracle's erroneous feedback, this discretized solution was proven to be ϵ-optimal.

More formally, the scheme presented in (93) obeyed the following updating rules:

Let $\breve{}(t)$ be the value at time step "t". In other words, $\breve{}(t)$ is an estimate of the unknown value of λ^* at time step "t". Then,

$\breve{}(t + 1) := \breve{}(t) + 1/N$ if Ξ suggests to increase $\breve{}$ and $0 \leq \breve{}(t) < 1$;

$\breve{}(t+1) := \breve{}(t) - 1/N$ if Ξ suggests to decrease $\breve{}$ and $0 < \breve{}(t) \leq 1$.

At the end states the scheme obeys:

$\breve{}(t+1) := \breve{}(t)$ if $\breve{}(t) = 1$ and Ξ suggests increasing $\breve{}$;

$\breve{}(t+1) := \breve{}(t)$ If $\breve{}(t) = 0$ and Ξ suggests decreasing $\breve{}$.

The analytical results derived in (93) proved that if the "Oracle" was itself *Informative*, the discretized random walk learning was asymptotically[1] optimal. Thus the mechanism would converge to a point arbitrarily close to the true point with an arbitrarily high probability.

In (101), Yazidi et al. presented a hierarchical solution to solve the SPL problem. The solution can be seen as a stochastic version of the bisection search and is shown to outperform legacy solutions.

In this approach, the learner queries the environment each time at three locations: end points of the current interval and the midpoint. Based on a decision table, a new interval is chosen. Consequently, the current interval might be pruned further or the search might be backtracked to a larger interval containing the previous visited interval.

The SPL has been successfully applied in different domains such as binomial estimation (102), quantile estimation (103, 104), stochastic root finding (105) and solving the non-linear stochastic knapsack problem (106, 107).

3.3 Solution: Learning Automata Control of LMA

In this section, we formally present our LA based solution to controlling the mobility of thermocline sensors to improve the link stability in underwater networks.
We design a LA with two actions $\alpha_k^i \in \{\alpha_0^i, \alpha_1^i\}$ such that the LA can respond to the environment by telling the Limited Mobile Agent (LMA) to choose one of to control states $\phi_i \in \{\phi_0, \phi_1\}$ corresponding to dive, or surface, respectively such that control maps onto actions as: α_0^i to ϕ_0 corresponding the dive command and α_1^i to ϕ_1 corresponding to a surface command.

The informed reader would observe that the problem has analogy to SPL as the LMA is allowed to choose on direction at each time step. At each depth level i, we attach an LA. We suppose that the minimum depth is 0 and the max depth is D.
Therefore there are $D+1$ LA attached to the locations $\{0, 1 \ldots D\}$.

In the same manner as in (51), we envisage the use of timeout to ensure that a learning

[1] As in the case of the field of LA, all the theoretical results reported here are limiting results, i.e., for example, when $N \to \infty$.

action occurs at all iterations of the algorithm and in order to allow exploring the different locations. In other words, each action of LA is valid for a certain amount of time and then a new action is chosen.

When diving, we will move to location $min(i+1, D)$ where D is the max depth. When moving to the surface, the new location is $max(i-1, 0)$, where 0 corresponds to the depth at the surface. Note that this is similar to SPL where the end state are self-loops.

Our algorithm is inspired by the family of pursuit LA algorithm (108, 109, 110). However, instead of pursuing the action with the highest reward among the offered actions, we pursue the action that leads to an increase in the reward compared to the previously visited state at time instant $t-1$

Construction of the Learning Automata

At each depth i we associate a 2-action S-Model (111, 112) Learning automaton, $(\Sigma^i, \Pi^i, \Gamma^i, \Upsilon^i, \Omega i)$, where Σ^i is the set of actions, Π^i is the set of action probabilities, Γ^i is the set of feedback inputs from the Environment, and Υ^i is the set of action probability updating rules.

1. *The set of actions of the automaton*: (Σ^i)
   The two actions of the automaton are α_k^i, for $k \in \{0, 1\}$, i,e, α_0^i and α_1^i
2. *The action probabilities*: (Π^i)
   $P_k^i(t)$ represent the probabilities of selecting the action α_k^i, for $k \in \{0, 1\}$, at time step t. Initially, $P_k^i(0) = 0.5$, for $k = 0, 1$.
3. *The feedback inputs from the Environment to each automaton*: (Γ^i)

   Whenever the LMA moves to a location i, the environment will return a continuous value representing the performance at that location which is a noisy measurement. Formally, the response from the Environment at time t and at location i is denoted by $\beta^i(t)$.

   We suppose that whenever the i^{th} LA denoted takes action surface, the next LA at position $min(i+1, D)$ will be activated. Similarly, if the action is dive, the next LA at position $max(i-1, 0)$ will be activated. This allows the LMA to find the most stable link in a stochastic environment through adaptation.

   Let $\bar{\beta}^i(t)$ be the estimated average reward obtained for location i since the first time step. It can be given by:

$$\bar{\beta}^i(t) = \frac{\sum_{l=1}^{t} J(l,i)\beta^i(l)}{\sum_{l=1}^{t} J(l,i)}$$

where $J(l,i) = 1$ if the location i action was deployed at the l^{th} time step.

4. *The action probability updating rules*: (Υ^i)

If α_k^i for $k \in \{0,1\}$ was chosen then, for $j \in \{0,1\}$. The LA update equations are given by:

$$P_j^i(t+1) \leftarrow P_j^i(t+1) + \theta(\delta_{jk} - P_j^i(t)) \tag{3.1}$$

where $0 < \theta \ll 1$ and:

$$\delta_{jk} = \begin{cases} 1 & \text{if } \bar{\beta}^i(t) > \bar{\beta}^{i^*}(t) \\ 0 & \text{else} \end{cases} \tag{3.2}$$

Here i^*, denote the locations visited by the LMA at time step $t+1$ as a result of the action taken at time step t. This location can be:

- $Min(i-1, 0)$ whenever α_0^i was taken corresponding to the dive command
- $max(i+1, D)$ whenever α_1^i was taken corresponding to the surface command.

Therefore, in other words, if the newly visited location i^* has better average reward than the location i we will increase the probability of the action leading to this location. However, if if the newly visited location i^* has inferior average reward than the location i we will decrease the probability of the action leading to this location.

In simpler terms, we have two cases.

Whenever $\bar{\beta}_k^i(t) > \bar{\beta}_k^{i^*}(t)$

$$\begin{aligned} &P_k^i(t+1) \leftarrow P_k^i(t) + \theta \times (1 - P_k^i(t)) \\ &P_{1-k}^i(t+1) \leftarrow 1 - P_k^i(t+1). \end{aligned}$$

Otherwise (i.e, $\bar{\beta}_k^i(t) \leq \bar{\beta}_k^{i^*}(t)$),

$$\begin{aligned} &P_k^i(t+1) \leftarrow P_k^i(t) + \theta \times (0 - P_k^i(t)) \\ &P_{1-k}^i(t+1) \leftarrow 1 - P_k^i(t+1). \end{aligned}$$

3.4 Experimental Results

We test our algorithm for one uni-modal performance function, and for one bi-modal function. In all experiments, the learning parameter $\theta = 0.01$. For obtaining steady probability we run the algorithm for 10^6 iterations. We suppose that the maximum depth is

100 called max_D and minimum depth called min_D at the surface is 0. In all the experiments, we observe a noisy version of the performance and therefore we use an additive noise function that follows a normal distribution, i.e, mean 0 and variance 1.

Noisy Uni-modal Performance Function

We suppose that the initial location for the LMA at time 0 is $\frac{min_D + max_D}{2}$. We used a Gaussian function with mean 60 and standard deviation 4. Figure 3.1 depicts the latter function.

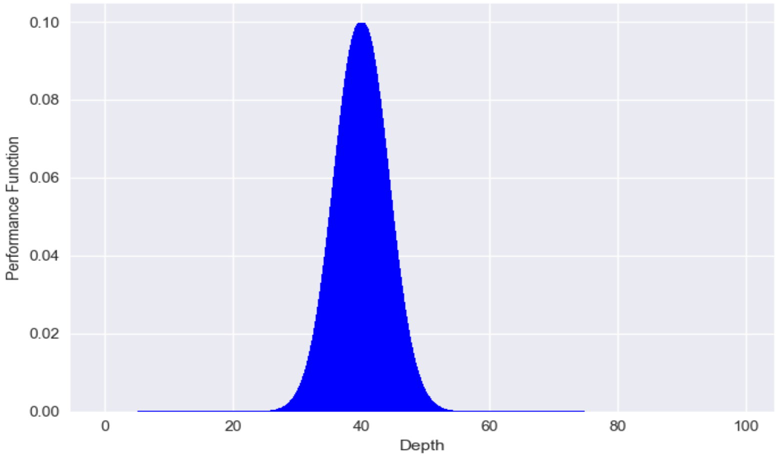


Figure 3.1 – *Unimodal Performance Function*

Figure 3.2 depicts the steady state probability over the different possible positions that reflects the percentage of time the LMA spends at each location when controlled by our pursuit LA scheme. We observe that most of the probability mass is concentrated around the max of the uni-modal function, namely 60 which corresponds to the max of the uni-modal function.

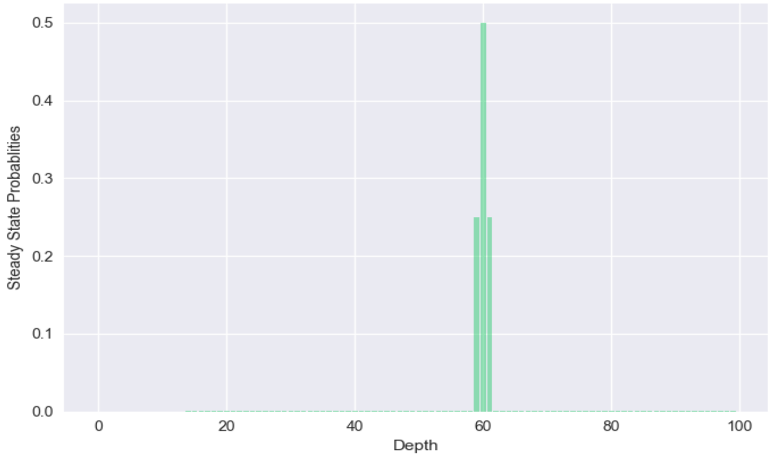


Figure 3.2 – *Steady Probability*

Testing with Flatter uni-modal function near the optimum

In this experiment, in order to obtain a flatter uni-modal function near the optimum we increase the variance to 30. From Figure 3.3, we see that the steady probability over the different location forms a distribution that is no longer a peak shape around the extreme but more a flat curve. However, we see clearly a peak around one of the extremes.

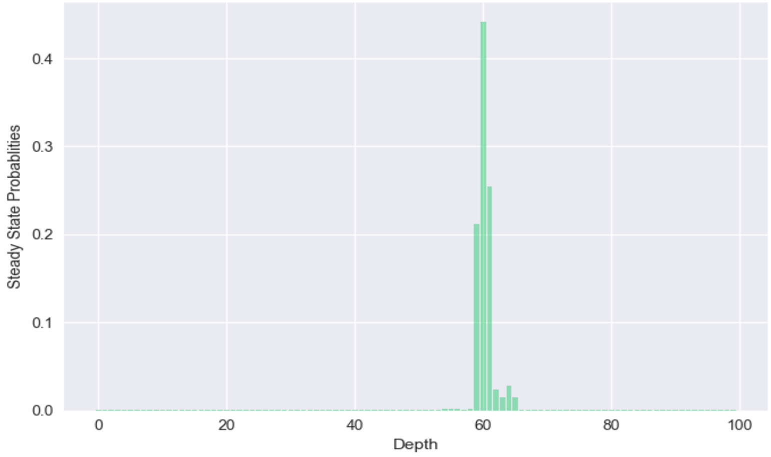


Figure 3.3 – *Steady state probability for uni-modal performance function with larger noise*

Noisy Bi-Modal Performance Function with equal extreme values

In this experiment, we use a bi-modal which is the superposition of two Gaussian functions: one with mean 20 and standard deviation 4 and one with mean 60 and standard deviation 4. Note that the function admits two extrema, namely 20 and 60 with the same performance. In other words, the noisy bi-Modal performance function has two equal extreme values. Thus, it is desired that the algorithm converges to one of those two extreme values. Note: the fact that the two extreme values are equal makes the problem more difficult because the algorithm needs to be designed in a such a manner that we guarantee convergence to one of the two extrema instead of not converging to any of them.

Figure 3.4 depicts the performance function but without the additive noise.

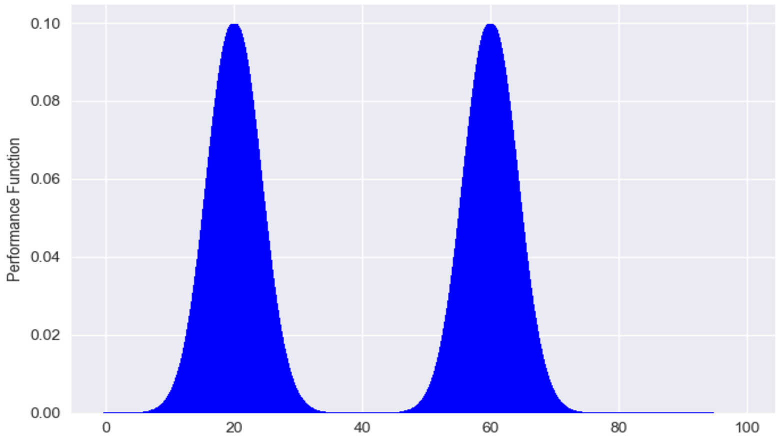


Figure 3.4 – *Bi-modal Performance Function with no additive noise*

In the following experiments we will verify that convergence in this case to one of the extrema is depending not only on the initial position but also on how flat is the performance function.

Figure 3.5 illustrates the steady probability of the LMA over the different positions when the initial depth is 20. As expected, we observe that the LA scheme converges to the position around 20 which is the maximum value of the performance function closest to 60.

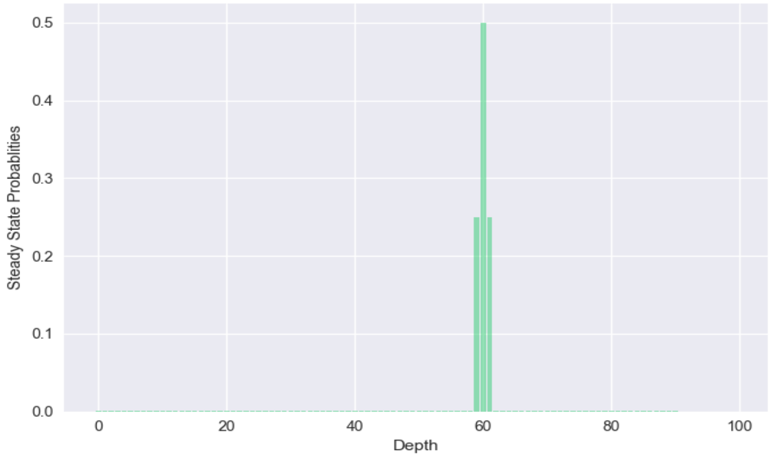


Figure 3.5 – *Steady Probability: initial state 20*

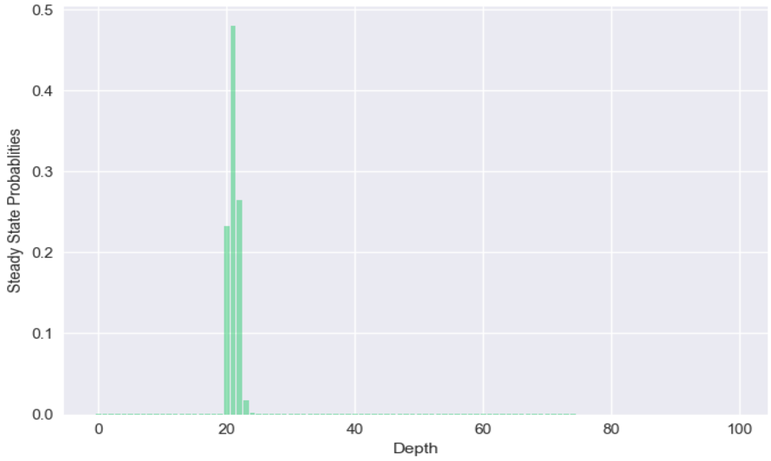


Figure 3.6 – *Steady Probability: initial state 60*

A similar results is illustrated in Figure 3.6 where the initial state is 60. We observe that the scheme concentrates the walk around 20.

Figure 3.7 depicts the trajectory of the LMA over time when starting at the middle depth, i.e, $\frac{min_D + max_D}{2}$ which corresponds in this case to depth 50. We observe that the algorithm quickly converges the location 60 which corresponds to one of the extremes.

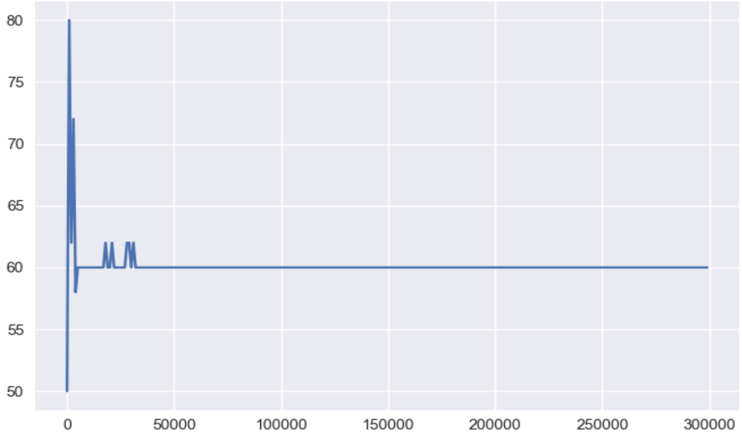


Figure 3.7 – *Trajectory of the LMA when starting at the middle depth*

Testing with Flatter Bi-Modal Performance Function

We increase the variance of the two Gaussian functions from 4 to 15. In other words, in this experiments we used a bi-modal which is the superposition of two Gaussian functions: one with mean 20 and standard deviation 15 and one with mean 60 and standard deviation 15.

By increasing the variance, the function becomes flatter and it becomes more difficult for the LA algorithm to distinguish a maximum performance point from a non-maximum performance point.

Figure 3.8 depicts the non-noisy version of the performance function i.e, with no additive noise which is clearly flatter than the function depicted in Figure 3.4 where the variance was 4.

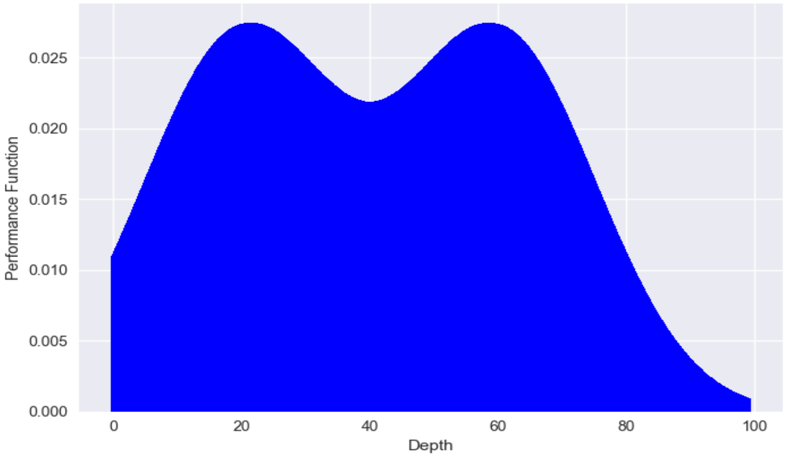


Figure 3.8 – *Bi-modal Performance Function with no additive noise and increased variance*

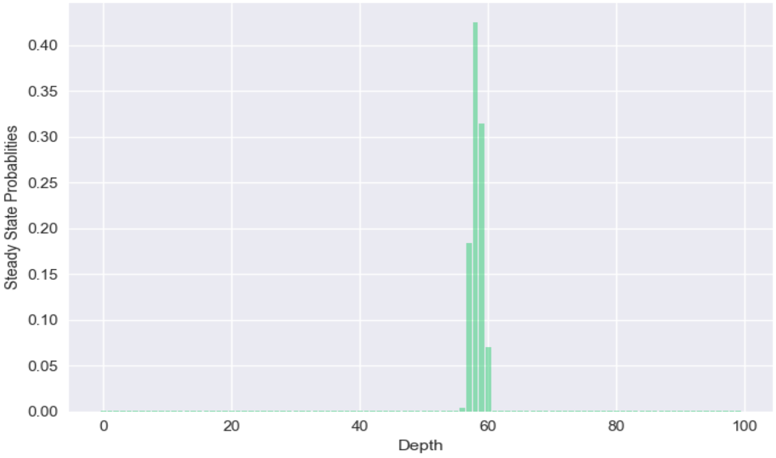


Figure 3.9 – *Steady Probability: initial state 20*

In Figure 3.9, we depict the steady state probability when the initial position is 20. Interestingly, from Figure 3.9, we see that the LMA converges to the neighborhood of the extrema 60. This is counter intuitive and unexpected as we would expect that since the initial state is 20 which is closer to extrema 40 than 60, the convergence will be around the extrema 40 instead of the far away extrema 60. This can be explained by the fact that the function is flatter in this case which allows the scheme to explore more the solution space and for a longer time.

3.5 Conclusion

In this chapter, we introduce an adaptive control mechanism to control the mobility of thermocline sensors to improve the link stability in underwater networks. We model the problem as a variant of the SPL problem (93, 101, 105). In a similar manner to SPL, the LMA is only allowed to move only into two directions. Our solution has a pursuit LA flavor. In contrast to classical SPL solutions, our pursuit LA exploits more effectively the information from the environment than traditional LA schemes that are myopic and use merely the last feedback from the environment instead of considering the whole history of the feedback. Experimental results show the performance of our algorithm and its ability to find the optimal sensor position.

Chapter 4

A Multi-Agent Diversity-based Gradient Approach for Quality of Monitoring Optimization in Underwater Sensor Networks

Contents

4.1 Introduction

Quality of Monitoring of Information is one of the most subject that attracts researchers in the field of Wireless sensors and connected objects. In recent years, many different types of Underwater wireless Sensor Networks (UWSNs) have been implemented for monitoring and scanning environments. UWSNs are made up of several number of autonomous sensor nodes. These sensor nodes are scattered in underwater to carry out detection tasks in order to collect different properties related to water (113). Collecting data is the main purpose of many explorations after detecting some specific data in order to make intelligent decisions (114). For example, monitoring the living conditions of fish, such as measuring temperature, humidity, pH and CO2 concentrations in order to associate them with the amount of fish produced under these conditions and relative to a given time. Previous solutions in Terrestrial Wireless Sensor Networks (TWSNs) are not well applied in UWSNs. As we know, RF signal and optical signal are not suitable for UWSNs due to underwater characteristics. Only the acoustic signal is suitable for the transmission of data collected in a marine environment (115).

The majority of these solutions even with traditional Wireless Sensor Network, neglects the problem of Redundancy correlation with target, because it is well known that the cost for each node to send the data collected is very high, therefore, it is essential to filter these data using techniques to reduce the amount of unnecessary traffic and subsequently minimize redundancy (116). In this work, we are using a simple function to quantify the quality of monitoring in Underwater wireless Sensor Network, by applying the gradient and covariance function to reduce redundancy and find the optimum coverage to place sensors to better cover the area and minimize energy consumption and extend lifetime of network.

The remainder of the chapter is as follows. Next Section, presents some general related work proposed in the literature and the type of underwater sensors used in this work. After that, the proposed solution is defined and at the end of this study, a series of experiments and some simulation results are presented to show the validity and relevance of the proposed approaches. finally, a conclusion of this Chapter will be presented.

4.2 Related Work

Monitoring of the marine environment has gained increasing attention in recent years with the development of various monitoring systems, due to growing concerns about climate change and also to be used in several other areas of different activities (117). The use of fixed sensors in buoys on water surface does not provide enough geographical coverage and also takes time with many drawbacks in certain applications with very sensitive

purposes such as environmental and tactical surveillance. As a result, recourse to the use of sensors and underwater robots helps to better resolve these problems(118). In particular, the application of advanced information and communication technology such as the Internet of Things (IoT) and the various learning methods to better manage the behavior of underwater vehicles (AUV) for water quality monitoring purpose. For example, the processing and visualization of water quality data can be done remotely and in real time using these underwater vehicles (AUV).
An example of this applications is presented in(119) where an underwater environment monitoring system based on UWSNs is introduced.This system has been conceived to be able to perform a large quantity of uninterrupted collected data. Further work is presented in (120) which introduces advanced wireless protocols developed for the IoT in order to highlight their adaptability for the WSN application used in water quality monitoring.

Having good localization coverage with low localization error is another problem that the authors in (115) try to overstate, for this they have proposed a top-down TPS positioning scheme for acoustic UWSNs while ensuring the quality of the new reference nodes during the definition of well-located nodes based on the gradient method.
In addition, this work presents a new method of estimating the 3D Euclidean distance to facilitate non-localized nodes to find more reference nodes in order to get localized. Many other spatial coverage algorithms are surveyed in (92) with a very detailed comparison. For collaborating mobile sensor networks, the distributed coverage control scheme is described In (121), where a density function of frequency random events with mobile sensors operating within a restricted range specified by a probabilistic model. The algorithm used in this work is based on the gradient which needs local information on each sensor and maximizes the probabilities of detection of common random events.For a coverage control problem, costs of communication are calculated according to two scenarios of data collection: the first takes the network as a network which collects data from single source and the second one identifies the network with multi-source. To model the cost of communication authors use the same form of energy consumption.
Authors in (122) deployed a same scheme in underwater environment, which is a gradient-based decentralized controller that dynamically adjusts the depth of an submarine sensor network to optimize detection for the calculation of highly detailed volumetric models. the study proved that the controller converges to a local minimum. This controller is adapted to a network of submarine sensors capable of adjusting their depths. The results of simulations and experiments have verified the functionality and performance of this system and the algorithm presented. This solution was implementing to solve the problem of monitoring chromophoric dissolved organic matter (CDOM) in the Neponset River that feeds into Boston harbor. An other research project, the SALMON (Sea Water Quality Monitoring and Management) presented (123) a concept of a guidance system

using autonomous underwater vehicles to detect and perform automated analysis of several water quality parameters.
In order to model the quality of monitoring (QoM), (124) focuses on the theoretical study of spatial and temporal correlations due to the various physical phenomena of Wireless sensor deployment in nature. Two schemes were proposed to reveal the time and space dependent under centralized and distributed setting to maximize the overall QoM based on sensing scheduling.The same authors proposed another study in (125) using the non-decreasing sub-modular function to measure the QoM but this time they took into account the correlation in the detected data in order to define distributed scheduling schemes which is used to determine a high QoM in a ring cycle sensor array.
In (126) authors proposed (RDBF) which is a relative remote routing protocol that takes into consideration energy saving while minimizing delays in transmission. This work is based on the use of an aptitude factor to determine the degree of relevance of a node to participate in transmitting packets, this aptitude test helps reduce the needless transfer by the nodes, which helps reduce power consumption and end-to-end delay,in addition to reduce redundancy by controlling transfer time of multiple sender.

However, none of the existing studies use these functions and algorithm to deal with issues of quality of monitoring in the underwater environment

How our underwater sensors AUV move horizontally and vertically

In recent years, the evolution to control robots has experienced a strong demand, especially based on learning instead of programming.Several methods have addressed this demand using genetic algorithms, neural networks and other Artificial Intelligent (AI) or machine learning methods to control some functionality of robots(127). The majority of mutlirobot systems rely on a default programmed algorithm, something that cannot be applied in a dynamic environment characterized by unpredictable change, therefore the robot system has to adapt with the environmental changes and take into account the local perception of the robot. In a changing environement,(128) authors proposed a Hierarchical Gene Regulatory Network (H-GRNe) for Adaptive Multirobot Pattern Formation, which is a two-layer gene regulatory network (GRN) model to adapt generation and formation of multirobot pattern. In this model the adaptation part of pattern generation is conducted in the first layer and then, these generated patterns will drives the robots in the second layer with a decentralized control mechanism. Authors accompanied their study with simulations in a changing environment that prove the efficiency of H-GRNe to form the desired pattern, and also a strong adaptation to robot failure.

In this study we used Autonomous underwater vehicle (AUV)robots because of their ability to move underwater without needing any external intervention. AUVs as Under-

water robots have great ability to be used in the areas of mining, agriculture and so forth (129). They are one of the most significant tools for the exploration and application of marine resources(130)(131). An autonomous underwater vehicle (AUV) is a self-piloting vehicle carrying its own power. For the realization of a task, the AUVs depend on an on-board artificial intelligence system with a set of programmed commands, which can be modified remotely by data or information broadcast by the vehicle's sensors (132).
In our network, we consider that the AUVs move in 2-D direction;horizontally and vertically by implementing Gene Regulatory Networks (GRNs) which is one of a widely used method in different field such as swarm robotics (133),(134),(135).
The used AUVs in this network apply the cellular adhesion molecules (CAM) combined with GRN controllers proposed by (136), this model is based on the control of GRN-CAM hydrons which refer in our case to a group of AUVs. Our network of AUVs is characterized by its ability to move horizontally by ejecting water and to move vertically, AUVs use a buoyancy control system.

4.3 An optimization function for water quality which minimizes sensor redundancy and maximizes diversity

This section provides the details of our solution, with the additional aim of highlighting the characteristics of the proposed architecture and how it is implemented.

We consider N AUVs at locations $P_i(x_i, y_i, z_i)$ with $i = 1, \ldots, N$. We assume that the sensors move in a two-dimensional plane defined by the x and z axes, with a fixed y coordinate, as seen in Figure 4.1, reducing the three-dimensional positioning to $p_i(x_i, z_i)$.

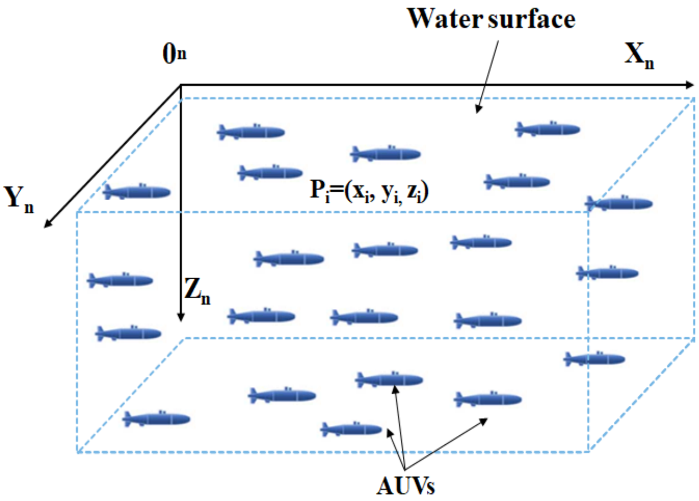


Figure 4.1 – *AUV system coordinates*

We will assume that the correlation between pairs of sensors decreases, not necessarily isotropically, with their distance as a Gaussian function. Consequently, we can postulate that the covariance between two sensors i and j is given by

$$\mathrm{Cov}(p_i, p_j) = \exp\left(-\frac{(x_i - x_j)^2}{2\sigma_x^2} - \frac{(z_i - z_j)^2}{2\sigma_z^2}\right) \tag{4.1}$$

where σ_x and σ_z have the meaning of (spatial) correlation decreasing rates in the x and z directions respectively.

Since we want to minimize redundancy among the sensors, we need to minimize the overall pairwise correlation between sensors. In other words, we minimize the following function:

$$H(p_1, \ldots, p_N) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \mathrm{Cov}(p_i, p_j)\,. \tag{4.2}$$

The minimum of $H(p_1, \ldots, p_N)$ fulfills the equations

$$\nabla_{x_i, z_i} H(p_1, \ldots, p_N) \quad \equiv \quad \left(\frac{\partial H}{\partial x_i}, \frac{\partial H}{\partial z_i}\right) = 0\,, \tag{4.3}$$

for $i = 1, \ldots, N$, yielding

$$\left(\sum_{j=i+1}^{N} \mathrm{Cov}(p_i, p_j)\frac{(x_i - x_j)}{\sigma_x^2}, \sum_{j=i+1}^{N} \mathrm{Cov}(p_i, p_j)\frac{(z_i - z_j)}{\sigma_z^2}\right) = 0\,. \tag{4.4}$$

Minimizing the function $H(p_1, \ldots, p_N)$ alone leads to a solution which indeed minimizes redundancy, but does not guarantee that one covers the maximum amount of information in the system. In other words, one also needs to take into account the diversity covered by the set of sensors. Assuming all the information of the system can be encoded in the linear correlations observed in the systems, the determinant of the covariance matrix L between pairs of sensors is a proper measure of such a total amount of information, since it reflects the total variance of the data collected by the set of sensors. The idea of using the determinant as a measure of diversity is found also in the the theory of determinantal point processes (137).

We therefore consider the covariance matrix L with elements $L_{ij} = \mathrm{Cov}(p_i, p_j)$ as defined in Eq. (4.1) and seek its maximum, which is a solution of

$$\nabla_{x_i, z_i} \det(L) \quad \equiv \quad \left(\frac{\partial \det(L)}{\partial x_i}, \frac{\partial \det(L)}{\partial z_i}\right) = 0\,, \tag{4.5}$$

which can be written as

$$\left(\frac{\det(L)}{\sigma_x^2}\mathrm{tr}(L \odot L^{-T}\frac{dG}{dx_i}), \frac{\det(L)}{\sigma_z^2}\mathrm{tr}(L \odot L^{-T}\frac{dG}{dz_i})\right) \quad = \quad 0\,, \tag{4.6}$$

where G is a matrix with elements $G_{ij} = G_{ji} = -(x_i - x_j)^2 - (z_i - z_j)^2$ and $\odot$ denotes the Hadamard product. For the full derivation of Eq. (4.6) see Append. C

We now combine both the redundancy H and diversity L in the same weighted objective function F, defined as

$$F = w\frac{H - H_{\text{min}}}{H_{\text{max}} - H_{\text{min}}} - (1-w)\frac{\det(L) - \det(L)_{\text{min}}}{\det(L)_{\text{max}} - \det(L)_{\text{max}}}\,, \tag{4.7}$$

where we consider the normalization of both function H and L to have values between 0 and 1, and introduce a parameter w which tunes how much the function H dominates over the function L.

For simplicity, we define

$$\begin{aligned} N_H &= H_{\text{max}} - H_{\text{min}} \\ N_L &= \det(L)_{\text{max}} - \det(L)_{\text{min}} \\ \zeta &= \frac{1-\omega}{\omega}\frac{N_H}{N_L} \end{aligned}$$

reducing the minimization problem

$$\nabla_{x_i,z_i}F \quad \equiv \quad 0 \tag{4.8}$$

to

$$\left(\frac{\partial H}{\partial x_i} - \zeta\frac{\partial \det(L)}{\partial x_i}, \frac{\partial H}{\partial z_i} - \zeta\frac{\partial \det(L)}{\partial z_i}\right) \quad = \quad 0\,. \tag{4.9}$$

Equation (4.9) together with equations (4.4) and (4.6) close the optimization problem for extracting the set of locations (x_i, z_i) of the N sensors which optimizes the redundancy and diversity together. Note that, as proven in Append. D, the gradient controller in Equation (4.9) converges to a critical point of F.

At this juncture, we are ready to present our multi-agent algorithm for optimizing the above objective function. From Equation (4.9), the numerical implementation of the optimization problem can be done through a simple Newton-Raphson scheme. Namely, let t denote a discrete time instant. We shall update the positions of sensor i recursively.

The position at time $t+1$ is given by:

$$\begin{aligned} x_i(t+1) &= x_i(t) - \lambda \frac{\partial F_\zeta}{\partial x_i}, & (4.10a) \\ z_i(t+1) &= z_i(t) - \lambda \frac{\partial F_\zeta}{\partial z_i}. & (4.10b) \end{aligned}$$

where λ is a learning parameter.

4.4 Numerical implementation and experiments

To test the performance of our algorithm, we adopt the same environment parameters describing the concentration of CDOM specific to the depth of the Neponset River caused by the tide found in (126). Each underwater environment is characterized by σ_s and σ_d. Although those parameters were not explicitly given by the authors in their studies (122, 126), we resort to a separability in the exponential function describing the covariance in order to extract them directly from Figure 4 in (122) via curve fitting.

For this first environment used in (122), we have: $\sigma_s = 2.074$ as covariance according to X and $\sigma_d = 0.917$ as covariance according to Z.

We use a grid size of length 8 km along the X direction and 3 m along the Z direction. Furthermore, we use a learning rate $\lambda = 0.1$. Choosing an excessively large value of the learning parameter λ gives wrong convergence and can make the system oscillate. However, choosing a too small value λ makes the convergence sluggish.

Now, we shall report the experimental results for different number of sensors. Our second environment is characterized by $\sigma_s = 1.977$ as covariance according to X and $\sigma_d = 1.198$ as covariance according to Z. We obtain similar results to environment 1. For the sake of brevity, we merely report the results for the second environment in D.1. Although we have conducted a large set experiments for different sets of sensors and different parameters of the multi-objective function, we merely report a few representative results for the sake of brevity as the conclusions are similar for the different experiments. When it comes to the objective function, we report results for two representative cases: $\omega = 0.8$ which describes a case where the multi-objective function weights the covariance minimization term more and $\omega = 0.2$ which describes a case where the multi-objective function favors more the diversity maximization term.

4.4.1 Case of 10 sensors

In this scenario, we deploy 10 sensors initially at uniformly random positions and we run our scheme using $\omega = 0.8$ and $\omega = 0.2$. Note that according to the multi-objective function, $\omega = 0.8$ puts more weight on the covariance while $\omega = 0.2$ puts more weight on the diversity.

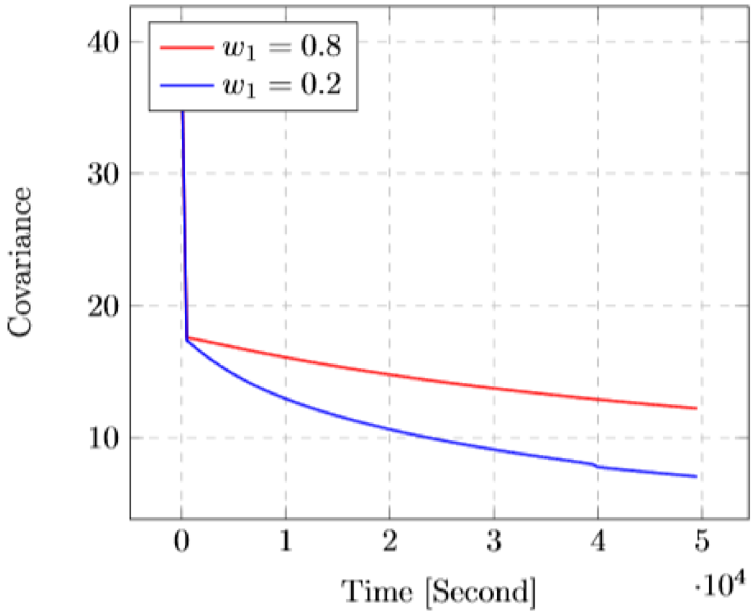


Figure 4.2 – *Covariance for* $w_1 = 0.2$ *and* $w_1 = 0.8$

Figure 4.2 shows the covariance after running our algorithm for 10^4 iterations. We can clearly see that in the case of $\omega = 0.2$ we obtain the minimal covariance and fastest convergence rate.

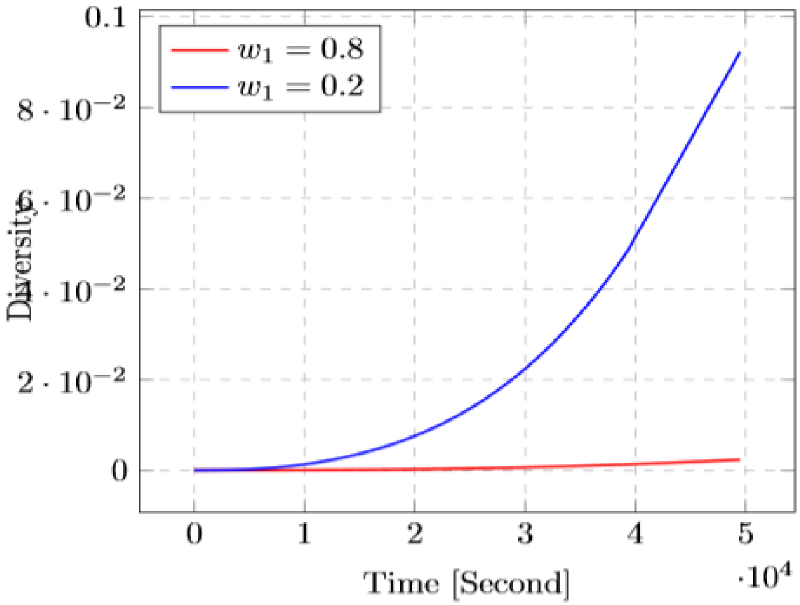


Figure 4.3 – *Diversity for $w_1 = 0.2$ and $w_1 = 0.8$*

Figure 4.3 shows that the corresponding diversity is largest for $\omega = 0.2$. We therefore conclude that by choosing $\omega = 0.2$, we obtain both lower covariance and higher diversity. In other terms, introducing the diversity term permits also to reduce the covariance as it seems that the diversity permits the optimization system to avoid some local minima.

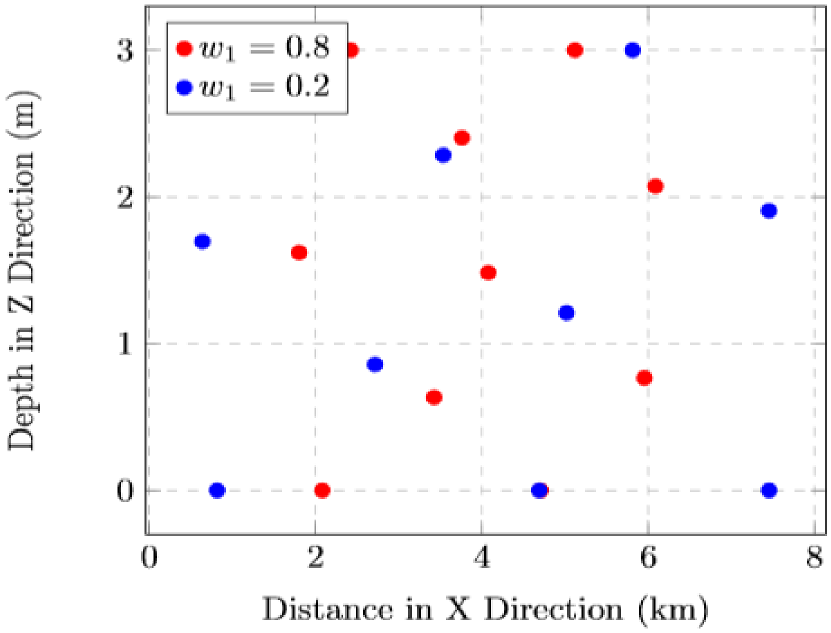


Figure 4.4 – *Final Positions for* $w_1 = 0.2$ *and* $w_1 = 0.8$

The final positions are depicted in Figure 4.4. We visually observe that the positions with $\omega = 0.2$ give a total coverage of the sensors while with $\omega = 0.8$ the sensors are positioned only in the middle and at the top of the network.

4.4.2 Case of 20 sensors

Now, we describe the experiment for 20 sensors. We use the same values of $\omega = 0.2$ and $\omega = 0.8$ and show the graphs for covariance, diversity and final positions.

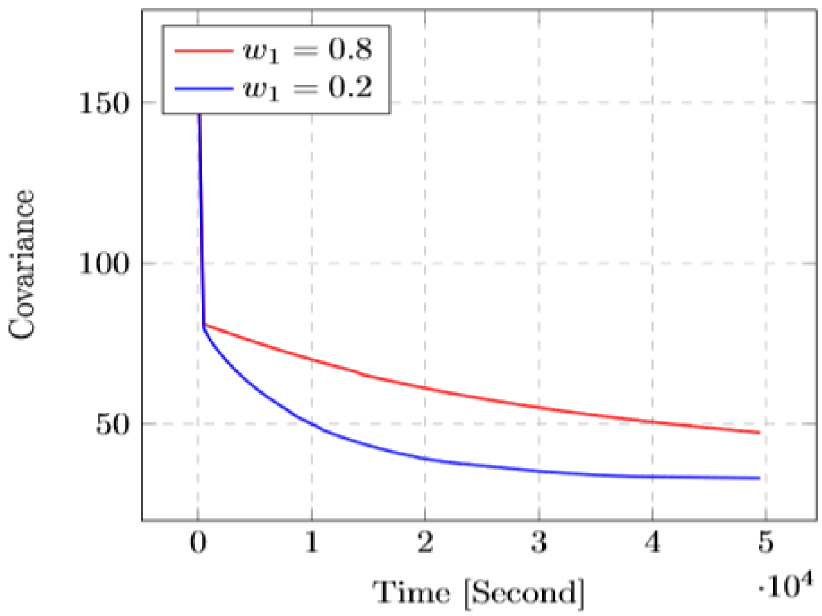


Figure 4.5 – *Covariance for* $w_1 = 0.2$ *and* $w_1 = 0.8$

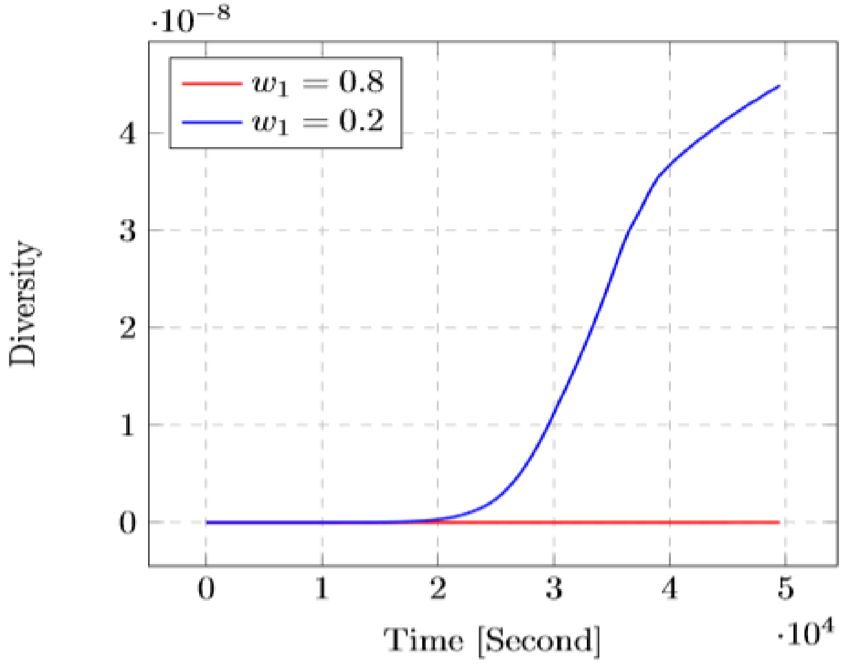


Figure 4.6 – *Diversity for* $w_1 = 0.2$ *and* $w_1 = 0.8$

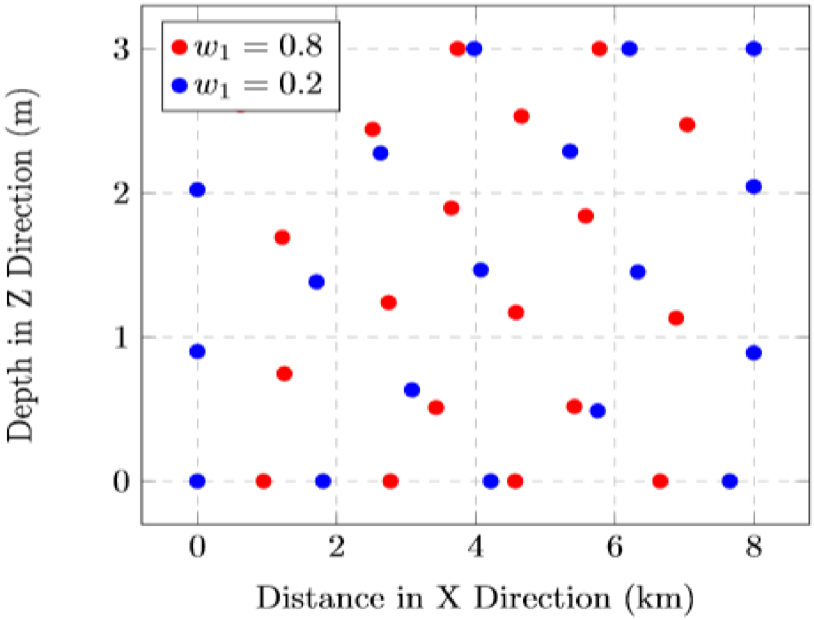


Figure 4.7 – *Final Positions for* $w_1 = 0.2$ *and* $w_1 = 0.8$

The covariance is depicted in Figure 4.5 where the convergence speed seems faster for $\omega = 0.2$ compared to $\omega = 0.8$. The rate of diversity is depicted in Figure 4.6 for both values $\omega = 0.2$ and $\omega = 0.8$. We can see that $\omega = 0.2$ gives a higher value for the diversity. The final position of this case study is presented in Figure 4.7 and we can verify visually the adequate positioning of the sensors with $\omega = 0.2$, despite the increase of the number of sensors.

4.5 Conclusion

In this Work, a new optimization algorithm based on covariance and diversity is presented to eliminate redundancy correlation with sensors in order to have a better positioning of sensor nodes in a marine environment.

Chapter 5

Energy performances of a routing protocol based on fuzzy logic approach in an underwater wireless sensor networks

Contents

5.1 Introduction

Ocean monitoring, disaster prevention, navigation and underwater communication are new application areas that need an emerging wireless technology with sensor network. This new technology is designed to detect events or phenomena, collect and process data, and transmit sensed information to interested users (138) for establishing an underwater network.

Using the mobile underwater sensors ends up with the use of acoustic signals instead of radio signals used in wireless sensor networks (WSNs). In fact, at very low frequency, radio signals cannot propagate to long distances and require high transmission power and large antennas.

Contrary to WSNs deploying an UWSNs is very expensive. The batteries used have a very limited memory and because of the adverse environment conditions. Due to this issue, the management of the energy of batteries is a problem that arises and has a high priority in UWSNs.

Energy consumption in the underwater wireless networks depends on the life time of the network. Increasing the network life time consists on decreasing the use of energy consumption. However, a big amount of energy consumption constraints exists in these networks in order to ensure the continuity of the nodes operations .Unfortunately, these nodes are powered by small batteries, which cannot be recharged or replaced, once exhausted (139).In this context, many research studies (140) have been done to reduce the energy consumption by modifying and creating new protocols.

In terrestrial wireless sensor networks, many routing protocols are proposed (141, 142) but still not adapted to the underwater areas because of the different environment characteristics. Yet, designing energy-efficient, robust and scalable routing protocols in underwater wireless sensor networks still very challenging (143).In order, to optimize the routing protocols different techniques have been proposed (144, 145, 146, 147, 148). Fuzzy logic is one of the techniques that researchers use in routing protocols to select the best forwarder node based in specific characters while saving energy.

Energy consumption is one of the major problems of the current generation. The contributions developed in this chapter aims to perform many simulations to evaluate the benefit of using Fuzzy Inference System (FIS) that provides the process of the mapping from a given input to output using Fuzzy Logic in routing protocol by studying the evolution of the energy metrics in order to save energy consumed and improve the life time of the network. The results of these simulations will be compared to those with the original routing protocol without fuzzy logic approach VBF(61)

The rest of the chapter is organized as follows. Firstly, in Section II, we present the VBF routing Protocol and we review the related work of Routing protocol with fuzzy logic approach already in existence for UWSN. After that, we we present a review of the VBF routing protocol and the Fuzzy Logic Optimized VBF Protocol (FLOVP) in section III, and we evaluate the performances the Fuzzy Logic Optimized VBF Protocol (FLOVP) with VBF protocol on Ns2 modeled in section IV. Finally a conclusion is presented in Section V to summarize the article basic points.

5.2 Background

There are a lot of researches and articles on UWSNs which used FIS for optimizing Cluster Head election and best route selection and localization. Furthermore, there are a few articles which introduce Fuzzy Logic for selecting best forwarder node with the impact on energy consumption. Some these Algorithms are introduced in this section.

The Gupta protocol (149) uses a Fuzzy Logic approach to select CHs. The FIS designer considered three descriptors: energy level, concentration, and centrality, each divided into three levels, and one output which is chance, divided into seven levels. The system also uses 27 IF-THEN rules. The CHEF protocol (Cluster Head Election mechanism using Fuzzy Logic in Wireless Sensor Networks)c(150), uses a Fuzzy Logic approach to maximize the lifetime of WSNs. It is similar to the Gupta protocol but it does not need the BS to collect information from all nodes.

A power-efficient routing 'PER Fuzzy' protocol for UWSNs is proposed in (151). The design of this protocol focus on enhancing performance metrics to fit the dynamic nature of underwater environments, especially the need for ease of deployment and the severe power constraints of the nodes are considered. Both fuzzy logic inference systems and decision trees are adopted as the candidates for a forwarding node selector to determine the suitable sensor(s) for forwarding packets, and a tree trimming mechanism is developed to prevent the growing of the packet for-warding tree so as to effectively reduce the power consumption of the sensor nodes.

In (143) a Fuzzy Depth Based Routing Protocol is proposed to improve the DBR (62) protocol by making routing decisions depend on fuzzy cost based on the residual energy of receiver node in conjunction with the depth difference of receiver node and previous forwarder node and the number of hops traveled by the received packet.

More in (152) Yahya Kord et .al propose a new routing protocol based on fuzzy logic with an algorithm that uses a two-step fuzzy logic system to select the appropriate CHs. The selection of CHs is based on six descriptors; residual energy, density, distances to base station, vulnerability index, centrality and distance between CHs. The selection of CHs is based on six descriptors; residual energy, density, distances to base station, vulnerability index, centrality and distance between CHs. The evaluations of the proposed algorithm show that it performs better in case of fair distribution and balancing of the overall energy consumption.

Another approach proposed by Nitin et al. (153) where they use the fuzzy logic to determine the CH selection and CS, to provide an optimal selection for CH as well as optimal intra and inter cluster communications based on energy and multiple paths. Through this approach the network can have an energy efficient transmission between the nodes and the clusters and also this approach prolongs the network life time.

5.2.1 Vector-Based Forwarding (VBF) Routing Protocol

In this section, we review the VBF (61)Routing Protocol and some related work on Fuzzy Logic Inference systems (FIS).

Vector Based Forwarder VBF is one of routing protocols that is essentially a location-based routing approach (61) in underwater sensor networks. Its main idea is the use of routing pipe, centered on the vector from the source to the sink (154). The sensor nodes in this pipe are eligible for packet forwarding, and those which are not close to the routing vector (i.e., the axis of the pipe) do not forward, they simply discard the received packets, as Peng Xie et .al explain in their technical report (61).

VBF introduce a self-adaptation algorithm based on a desirableness factor to measure the 'suitableness' of a node to forward packets. In this algorithm, when a node receives a packet, it first determines if it is close enough to the routing vector as it depicted in Figure 5.1. If yes, the node then holds the packet for a time period related to its desirableness factor (61). Although two nodes in the same level they will receive and send data, which doubles the cost of the lost energy. To summarise: - VBF main idea is the use of routing pipe, centered around a vector from the source to the sink. - VBF works just with one factor, called Desirableness factor decides whether the node can forward packet or must discard it.

5.2.2 FLOVP Vector Fuzzy Logic Optimised Routing Protocol

To improve the VBF protocol, authors in [3] used a Fuzzy Inference System (FIS), which provides the process of the mapping from a given input(s) to output using Fuzzy Logic in order to prolong the lifetime of UWSNs, and optimizing the energy consumption.

5.2.2.1 Fuzzy Logic Architecture Model

The model architecture is depicted in the Figure 5.2, where the Mamdani FIS system has 4 parts: a Fuziffier, fuzzy rules, fuzzy inference engine, and a deffuzifier are the main components of our model of fuzzy logic control. The model is based on the so-called Mamdani (155) method which is the most commonly used in fuzzy inference technique.

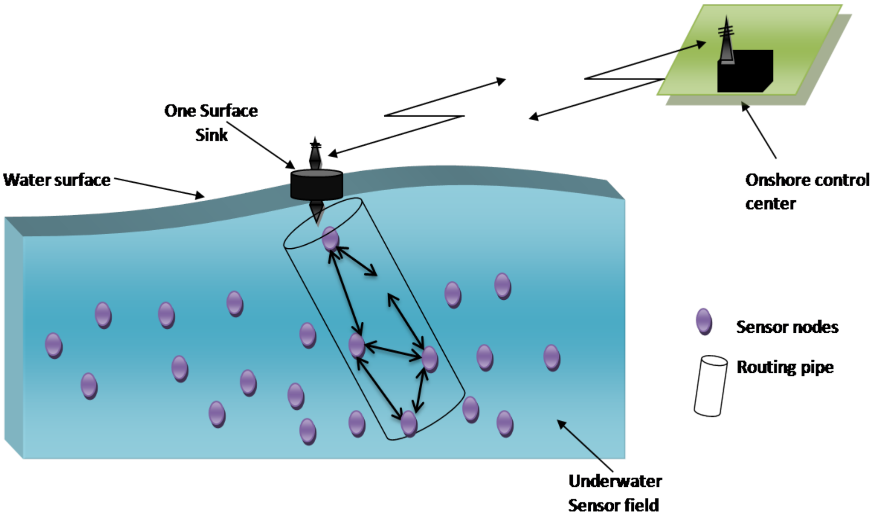


Figure 5.1 – *VBF mechanism for UWSNs.*

Thus, the way our system performs as inference techniques and our Fuzzy Logic system is described in a simplified way in Figure 5.3.

5.2.2.2 Step1: Input of Crisp Values and Fuzzification

Fuzzification of the input variables Weight, Closeness and Depth - taking the crisp inputs from each of these and determining the degree to which these inputs belong to each of the appropriate fuzzy sets. For using these three different parameters, in order to make them comparative with each other, we use normalization. For this reason parameters must divided by their maximum number. In Following we can see how the maximum number is obtained.

The outcome to represent the forwarder selection chance was divided into nine levels: very weak, weak, little weak, medium, low medium, medium, and high medium, little strong, strong and very strong. The fuzzy rule base currently includes rules like the following: if the closeness is close and the weight is high and the depth is high then the forwarder node selection chance is very strong.
Thus, we used $3^3 = 27$ rules for the fuzzy rule base. And we used triangle membership functions to represent the fuzzy sets medium and adequate and trapezoid membership functions to represent low, high, close and far fuzzy sets. The Triangle and Trapezoidal membership functions are more useful than the other types because their degree is more easily determined.

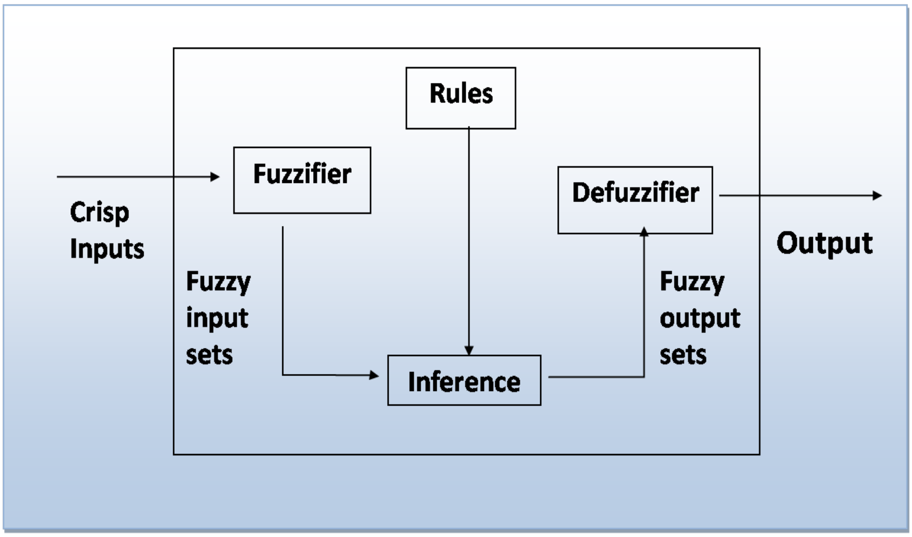


Figure 5.2 – *Fuzzy logic system*

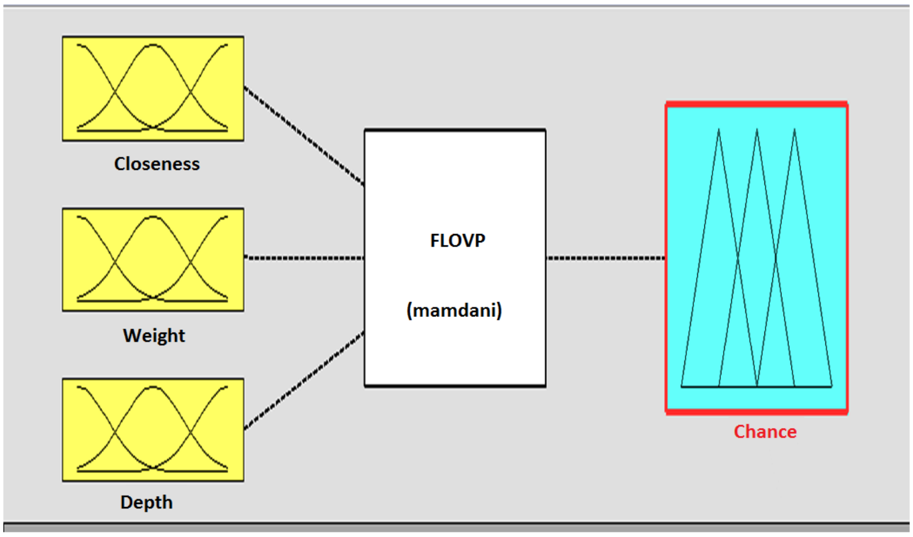


Figure 5.3 – *FIS FLOVP Model*

Table 5.1 – *Description of the Input of Crisp Values*

Input of Crisp Values	**Description**
Closeness	• The first parameter is computed from the following expression: **Closeness = p/w + (R - d x c) / R** P: refers to the distance between selected nodes to routing vector. W: presents the width of routing vector and shows the closeness of selected node to the routing vector and the second proportion.
Weight	• This parameter has a direct relation with energy consumption. • This parameter indicates that if the node is selected as a forwarder before or not. If yes, its weight increases. • This is used to determine the energy level of the node. • The node which works as a forwarder its energy will degrade. • The initial energy level and in other words the maximum energy level for each node are 10000. • Each forwarder consumes 10 J per packet. As a result, each forwarder can be selected 1000 times.
Depth	• The maximum of depth is determined in the simulation phase. • The depth of the dimension of our simulation is the maximum depth.

In the Following equations 5.1,5.2 we can see the formulas for Triangle and Trapezoidal membership functions. We present the medium level (medium) with triangle membership, while we present both sides levels (low, high, close, and far) by a trapezoidal membership function.

The triangle Membership Function is presented as follow:

$$\mu_A(x) = \begin{cases} 0, & x < a, \\ \dfrac{x-a}{b-a}, & a \leq x \leq b, \\ \dfrac{c-x}{c-b}, & b \leq x \leq c, \\ 0, & c \leq x \end{cases} \tag{5.1}$$

However, we calculate the Trapezoidal Membership Function as follow:

$$\mu_A(x) = \begin{cases} 0, & x < a, \\ \dfrac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ \dfrac{d-x}{d-b}, & c \leq x \leq d, \\ 0, & d \leq x \end{cases} \tag{5.2}$$

The membership functions developed and their corresponding linguistic states are represented in Table 5.2.2.2 and from Figure 5.4 to Figure 5.7.

Table 5.2 – *Linguistic sets for membership functions*

	Weight	Closeness	Depth
1	Low	Far	Low
2	Low	Far	Medium
3	Low	Far	High
4	Low	Medium	Low
5	Low	Medium	Medium
6	Low	Medium	High
7	Low	Close	Low
8	Low	Close	Medium
9	Low	Close	High
10	Medium	Far	Low
11	Medium	Far	Medium
12	Medium	Far	High
13	Medium	Medium	Low
14	Medium	Medium	Medium
15	Medium	Medium	High
16	Medium	Far	Low
17	Medium	Far	Medium
18	Medium	Far	High
19	High	Close	Low
20	High	Close	Medium
21	High	Close	High
22	High	Medium	Low
23	High	Medium	Medium
24	High	Medium	High
25	High	Far	Low
26	High	Far	Medium
27	High	Far	High

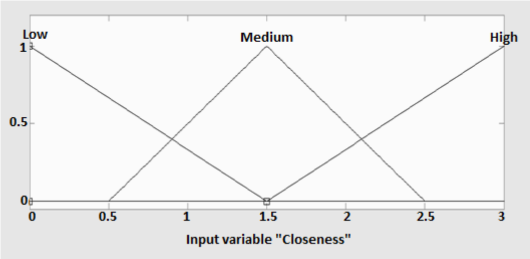


Figure 5.4 – *Fuzzy Sets for Fuzzy Variable Closeness*

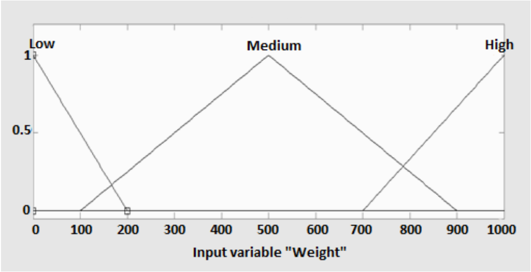


Figure 5.5 – *Fuzzy Sets for Fuzzy Variable Weight*

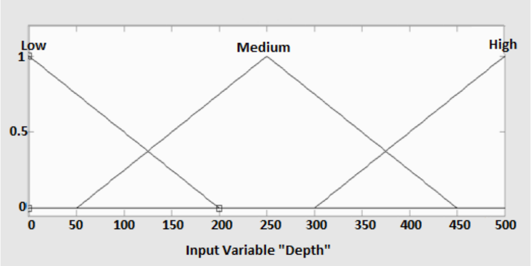


Figure 5.6 – *Fuzzy Sets for Fuzzy Variable Depth*

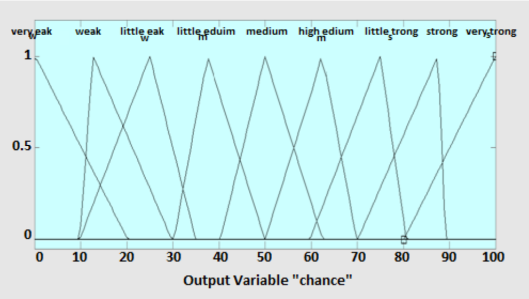


Figure 5.7 – *Fuzzy Set for Fuzzy Output Chance*

5.2.2.3 Step 2: Rule Evaluation

To determine our new fuzzy output set, we supply/feed fuzzified inputs to our IF-THEN rules we use SUM-PROD inference; which utilize multiply for AND operand and sum for OR operand. There are some approaches to combine expressions for the result of rules. Finally the output decision here Chance is made based of the intersection of the corresponding members of the sets input values: the Weight, the Closeness and the Depth. Table 5.2.2.3 presents the results of the conditions making in fuzzy logic for each input.

Table 5.3 – *Fuzzy Rule Evaluation*

	Weight	Closeness	Depth	Chance
1	Low	Far	Low	Very weak
2	Low	Far	Medium	Weak
3	Low	Far	High	Low Weak
4	Low	Medium	Low	Weak
5	Low	Medium	Low	Low weak
6	Low	Medium	High	Low medium
7	Low	Close	Low	Low weak
8	Low	Close	Medium	Low medium
9	Low	Close	High	medium
10	Medium	Far	Low	Low weak
11	Medium	Far	Medium	Medium
12	Medium	Far	High	Low medium
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Low medium
15	Medium	Medium	High	Medium
16	Medium	Far	Low	High medium
17	Medium	Far	Medium	Medium
18	Medium	Far	High	High medium
19	High	Close	Low	Low strong
20	High	Close	Medium	Medium
21	High	Close	High	High medium
22	High	Medium	Low	Low strong
23	High	Medium	Medium	High medium
24	High	Medium	High	Low strong
25	High	Far	Low	Strong
26	High	Far	Medium	Strong
27	High	Far	High	Very strong

5.2.2.4 Step 3: Aggregation of the rule outputs

The aggregation of the rule outputs is described in the Figure 5.8, where the OR operator simply selects the maximum of the 27 rule evaluation values, to generate the new aggregate fuzzy set that we will use in next step.

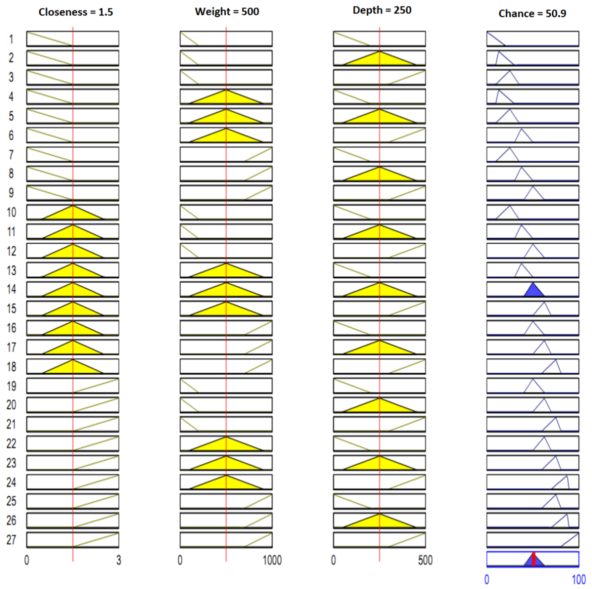


Figure 5.8 – *Aggregation of the rule*

Step 4: Defuzzification

We cannot have our chance value without the last step of defuzzification, where we define the process of transforming a fuzzy output of a fuzzy inference system into a crisp output. Firstly with the use of Mamdani technique we calculate the implication value. Therefore, the Centroid defuzzification method helps us to find the final crisp number to represent the CH election chance value to form the cluster formation. So that, the equation 5.3 uses the Center Of Area (COA) to compute the centroid defuzzification.

$$\mathbf{R} = \sum_{i=0}^{n} W_i \mu_A(W_i) / \sum_{i=0}^{n} \mu_A(W_i) \tag{5.3}$$

With, W_i *is the domain value corresponding to rule.*
i, n *are the number of rules triggered in the fuzzy inference engine.*
$\mu_A(W_i)$ *is the predicate truth for that domain value.*

5.3 Performance Evaluation

5.3.1 Energy analysis parameters

In this section we present an analysing description of the considered metrics with which we will evaluate the two routing protocol in the simulations part.
The first considered metrics is the Total Energy Consumption (ETC) which can be com-

puted as follows:

$$\textbf{ETC = Er + Th} \tag{5.4}$$

Where: **Th**: is the throughput of the network and can be expressed as:

$$\textbf{Th} = \sum_{i=1}^{n} \Big(\prod_{j=1}^{h(i)} p(h(j)) \Big) \lambda \tag{5.5}$$

$$\begin{aligned} \textit{With,} \quad \lambda \quad & \textit{is the total number of generated packets at each node} \\ n \quad & \textit{is the number of nodes..} \\ h(i) \quad & \textit{is the number of hops of ith node.} \\ p(h(j)) \quad & \textit{is the number of hops of jth node.} \end{aligned}$$

Er: is the total energy consumption of the network in one data gathering round without contention and it is given as follows:

$$E_r = \sum_{i=1}^{n} \Big(e_g + \sum_{j=1}^{h(i)} (e_t + N_j \times e_r) \Big) \tag{5.6}$$

With,

N_j *is the number of neighbor nodes at the jth hop*

e_g *is the average energy required to generate one data packet.*

e_t *is the average energy required to transmit one data packet from source node to destination/relay node*

e_r *is the average energy required to receive one data packet from the source node.*

So, we can find the final expression of our total energy consumption 5.4 as follow:

$$\textbf{ETC} = \sum_{i=1}^{n} \Big(e_g + \sum_{j=1}^{h(i)} (e_t + N_j \times e_r) \Big) \lambda \tag{5.7}$$

And then, the second metric the Average Energy Consumption (AEC) per successful packet can be calculated as follow:

$$\textbf{AEC= ETC/Th} \tag{5.8}$$

Then, we found the final expression of the equation 5.9 by using the equations 5.7 and 5.5 as follow:

$$\textbf{AEC} = \frac{\sum_{i=1}^{n}\left(e_g + \sum_{j=1}^{h(i)}(e_t + N_j \times e_r)\right)\lambda}{\sum_{i=1}^{n}\left(\prod_{j=1}^{h(i)} p(h(j))\right)\lambda} \tag{5.9}$$

5.3.2 Simulation analysis and Results

The architecture of our submarine sensor network takes into consideration the general parameters described in the Table 5.4 to evaluate the considered metrics in Table 5.5.

To implement our simulations we used Aqua-sim [8] simulator which is based on NS2 tool, where we find all the parameters that characterizes the underwater environment. Two scenarios are considered for the performance analysis of our proposed protocol. The first scenario studied the effect of varying de number of nodes in the network which start from 500 nodes to 3000 nodes. In the other hand, the second scenario is interested to assess the behavior of the network with different value of the speed of the node in the network

In the first scenario we consider the impact of Network Size to evaluate our metrics.

Table 5.4 – *Configuration Parameters*

• N sensor nodes which are randomly distributed in 3D field of 1000m x 1000m x 500 .
• One source fixed at location (900, 900, 500) near one corner of the field at the floor
• One sink located in (100, 100, 0) near the opposite corner at the surface
• all other nodes are mobile
• Each node randomly selects a destination and moves toward that destination. Once the node arrives at the destination, it randomly selects a new destination and moves in a new direction.

The TEC i.e. total energy consumption is illustrated in Figure 5.9. It can be seen that the energy consumption of FLOVP is slightly increased when the number of sensor nodes gets larger. And it is more significant as the network gets denser. This is reasonable as our

Table 5.5 – *The considered metrics*

TEC	Total Energy Consumption
AEC	Average Energy Consumption

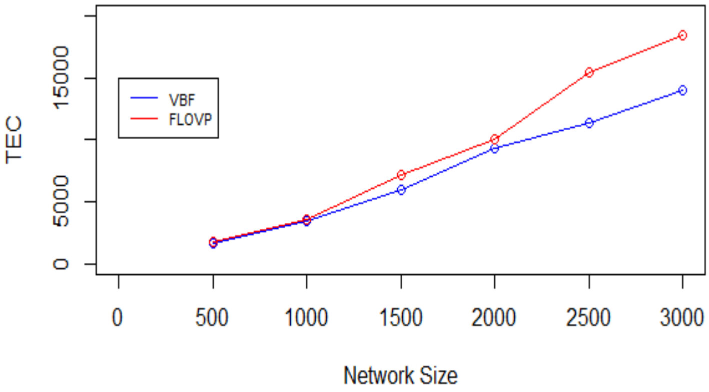


Figure 5.9 – *TEC vs. Network Size*

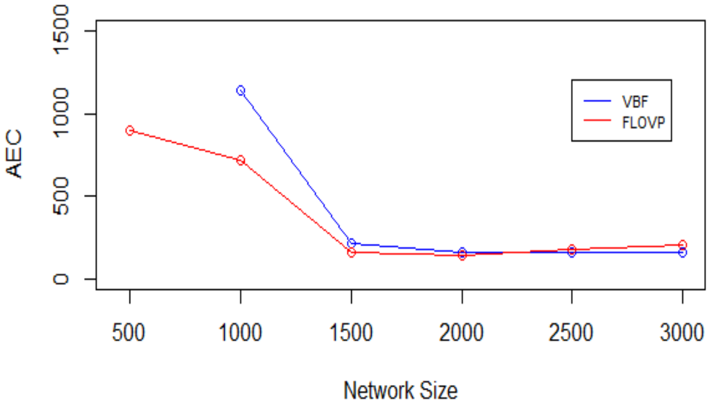


Figure 5.10 – *AEC vs. Network Size*

proposed Fuzzy algorithm uses three parameters for decision making. So, all nodes in the neighborhood compute their fuzzy to select the next node as a forwarder .As a result the total energy of network increases. It can be one of the issues of our future work. However, when we compute the energy consumption of each node the Figure 10 is obtained.

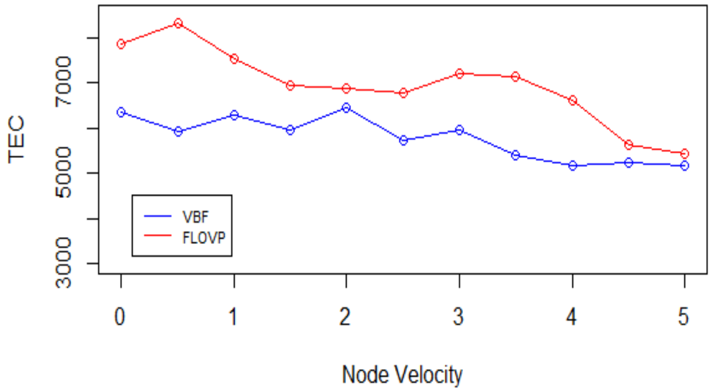


Figure 5.11 – *TEC vs. Node Velocity*

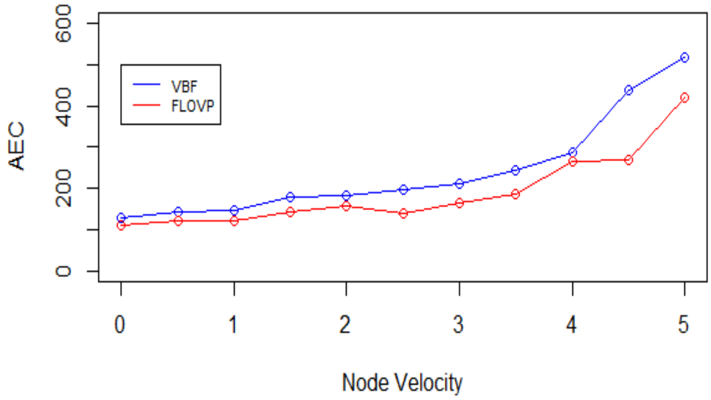


Figure 5.12 – *AEC vs. Node Velocity*

As we can see in Figure 5.10, with increasing the number of nodes to 1000 nodes in sparse network the total energy rises. But as the network becomes denser, the average energy consumption decreases. Furthermore, the trend of Energy consumption in FLOVP is at the lower level than that of VBF, except in network size higher than approximately 2300 nodes. The impact of Node Velocity was the second Scenario where, the proposed Protocol was evaluated in comparison with the VBF routing protocol. In this scenario the network size was fixed at 1500, and the node speed is varied from 0 to 5 m/s. The result is showing in the following.

The impact of the variation in node velocity under the water on the total energy con-

sumption is presented in Figure 5.11. As it mentioned before, in both algorithm with increasing node velocity, the number of nodes which is exist in the routing pipe degrades and as a result lower nodes needs to calculate their fuzzy output for selecting forwarder node. As a result, the total energy consumption trend for whole network decreases. But in FLOVP protocol each node provided with a more complex computation so need more energy. Therefore, consume more energy in contrast to VBF.
In the last Figure 5.12, the average energy consumption for each packet in terms of node velocity is illustrated. As it can be seen, with increasing in nodes velocity in VBF protocol, more energy is consumed for forwarding packet. On the other hand, in FLOVP when the amount of packet delivery is higher, the total energy consumption will decrease; consequently the average energy consumption for forwarding each packet is lower than that of in VBF.

5.4 Conclusion

In this chapter, we have evaluated an improving routing protocols which use the Mamdani technique based on Fuzzy Logic approach in underwater environment. Through two analysing scenarios, the effect of the number of nodes and node's velocity on energy consumption were tested with FLVOP protocol and compared to the VBF protocol.

The results show that the performance obtained through using fuzzy logic protocol is satisfactory compared to the VBF protocol but the trend of Energy consumption in FLOVP is at the least level than that of VBF, except in a network size higher than a threshold. This study enabled us to understand how we can improve energy consumption through implementing a new algorithm based on the fuzzy logic approach.

Conclusions

Summary

70,71% is the proportion of water on the earth as against of only 29,29% of the ground, a fairly appealing percentage that makes sea space a vast natural field to explore. Several scientists from different fields have proved that the seabed and all other marine sources give rise to many natural phenomena. This led to several searches in this natural environment, such as the development of underwater navigation during the Second World War and also in 1945 the researchers began to deploy the underwater phone, which produced a wide range of applications.

Building IoUT has become a very attractive area for researcher and industry due to the ability of this type of networks to provide data in different fields, using sensor nodes that are battery powered and placed on remote locations same as in the WSNs, as it has been presented namely surveillance, study and monitoring of underwater events for security and offshore exploration. Thus, the Underwater environment is a network that is characterized by its unique conditions, which affect its performance in general, which make its construction difficult and makes it a big challenge to overcome.

Communication between underwater devices is the key that makes all these applications viable, including unmanned vehicles in environments that are restrictive to humans. In addition, the use of small batteries and for long duration imposes a strict consumption of energy to extend the life of the network. In the other hand, the study of the sensor's positions is very important to better facilitate communication and subsequently to guarantee better network coverage and also to save more energy which will extend the life's network.

This thesis has investigated a study of Underwater Wireless Sensor Networks in term of Transport Communication Protocol, energy consumption and position of sensors and data monitoring. The result of our contributions are as follows:

The first Chapter brings a general introduction that presents the Wireless Sensor and Underwater Wireless sensor in terms of definition, components, architectures, domain of applications challenges and a comparison between this two kind of wireless environments.

In the second chapter which refers to our first contribution, we start by presetting a comparative study of two traditional TCP mechanisms in underwater environment. After that, we present two different amelioration of NewReno TCP in UWSNs mainly the adjustment of the RTT and the adaptation of the windows. Finally, based on this analysis and results We conclude this chapter by presenting a new TCP version on New Reno specially adapted to Underwater environment named U-NewReno. Simulation results affirms the large advantage of U-NewReno compared to its based TCP NewReno.

The third chapter presents our contribution related to the control of the thermocline sensors mobility, this work aims to build an adaptive mecanisme in order to improve the link stability in underwater networks, and we showed that using a pursuit LA algorithm gives a great results to find an appropriate positions to the sensors.

Chapter four, introduces an approach based on a multi-objective function to improve the quality of surveillance information in order to minimize the covariance between all the sensors, in other words, reduce redundancy and at the same time maximize diversity. The gradient was used to iteratively adjust the positions of the sensor nodes. Simulation results based on realistic environmental conditions are produced, which conquer with the theoretical results, and illustrate the performance of our approach.

Chapter six, has investigated the reduction and the evaluation of energy consumption of VBF routing protocol by implementing the fuzzy logic theory. The simulation results showed a significant performances compared to the normal VBF Routing protocol.

Future research directions

While this thesis presented, a lot of perspectives remain open for large domain and applications. In this subsection we discuss directions for future research in the field of context and prediction of future context in underwater application systems for IoUT.

For our first contribution concerning the TCP un Underwater in IoUTs, the first perspective is to investigate the study how other parameters in different constraint in diverse phase of transmission can influence the performance of TCP U-New Reno in the underwater environment. We also plan to study the use of this protocol in different real applications and compare its performance with other variants of TCP. Furthermore, it will be relevant to perform the same study on other traditional TCP in order to adapt its parameters to get better performance in this kind of environment.

For the second contribution, this research work could lead to the development of small autonomous robots for more precise and extended underwater search and rescue operations in IoUT.

For the third contribution, as future work, this contribution opens up an opportunity for us to discover more parameters that influence the quality on monitoring in UWSNs for example, we could try to jointly optimize the communication cost and quality of monitoring.

For the last contribution, interesting perspectives open up and are built on the basis of the results of our work, since saving energy is one of the most challenged point in IoUTs and which go beyond the scope of this thesis. As discussed above, there remains more work to be carried out for covering all the aspects of energy saving and adaptation conditions so that the saving energy produces quality results. Our work could be treated by integrating different methods such as, the use of Machine Learning algorithms and optimization techniques in order to lower and optimize the cost of energy consumption over routing protocol.

Ph.D. Publications

International Journals (SCOPUS)

1. **Bennouri Hajar**, Berqia Amine,Yazidi Anis"*A Diversity based Gradient Approach for Quality of Monitoring in Underwater Sensor Networks*", **Submitted** to the ACM Journal.
2. **Bennouri Hajar**, Berqia Amine, Patrick N.Koffi, "*TCP U-NewReno: a Transmission Control Protocol to Enhance transmission communication in Submarine Wireless Sensor*", **Accepted** in Journal of King Saud University - Computer and Information Sciences
3. **Bennouri Hajar**, Berqia Amine,"*Assessing the performance of different TCP congestion mechanisms in underwater wireless sensor networks*", International Journal of Vehicle Information and Communication Systems, 2020
4. **Bennouri Hajar**, Berqia Amine, Patrick N.Koffi, "*Adapting the appropriate RTT timeout of TCP newreno in submarine communication networks*", Journal of Communications, 2019

International Conferences(SCOPUS)

1. **Bennouri Hajar**, Berqia Amine," *Energy performances of a routing protocol based on fuzzy logic approach in an underwater wireless sensor networks*", 2019 International Conference on High Performance Computing & Simulation (HPCS), Dublin, Ireland, 2019, pp. 990-994, doi: 10.1109/HPCS48598.2019.9188061.
2. **Bennouri Hajar**, Berqia Amine, Patrick N.Koffi,"*Controlling Maximum Window of TCP NewReno in Underwater Wireless Sensor Network*", International Symposium on Advanced Electrical and Communication Technologies, ISAECT 2018 - Proceedings.
3. **Hajar Bennouri**, Anis Yazidi, Amine Berqia, " *A pursuit learning solution to underwater communications with limited mobility agents*", RACS 2018: 112-117
4. **Bennouri Hajar**, Berqia Amine, "*The Impact of TCP Packet Size and Number of TCP Connections in Underwater Wireless Sensor Networks*", Proceedings on 2018 International Conference on Advances in Computing and Communication Engineering, ICACCE 2018
5. **Bennouri Hajar**, Berqia Amine, "*TCP Throughput and Packet Delivery Ratio performances in an Underwater Wireless Sensor Network*", at IEEE International Conference on Communications for Connecting Humanity -Joint 7th N2Women and WICE: Professional Development Workshop. 20-24, Mai Kansas City, MO, USA (POSTER)

Appendix A

Network Simulator 2: NS2

A.1 Introduction

Network Simulator v2 (NS2) is a discrete event simulator targeted at networking research. NS began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years.
In 1995, NS development was supported by DARPA (Defense Advanced Research Projects Agency) through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI(156).
Currently, NS development is supported through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI.

A.2 Structure

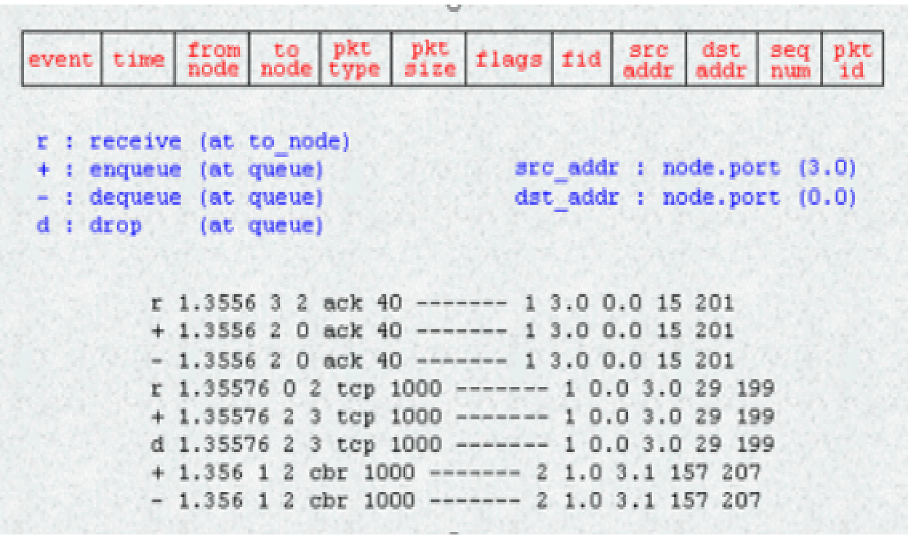
The simulator is written in C/C++ and is interfaced with TCL (Tool Command Language) /OTCL (Object-oriented extension of Tcl)(69) (157). The distribution between the two languages is that the simulator kernel and the network modules are written in C/C++.
Interfacing with the simulator is done with TCL/OTCL where the network will be initiated, the topology built and the different events in the simulation configured. With this distribution of languages, the compiled modules perform well, but there is no need to recompile whenever a change occurs in the topology(69). However, NS2 cannot handle a simulation in the underwater environment because this environment requires specific components that are not built-in.

A.3 Tool Command Language and Object-oriented TCL

Tool Command Language (TCL) is most probably used for writing simulation code. An example in(69) is provided to show how to program in TCL. OTCL is an extension of Tcl with object-oriented programming. It is dynamically extensible and built on Tcl syntax and concepts. The network simulator NS is actually a special version of an OTcl programming language interpreter (69). This work provide a basis is provided for learning about OTCL.

A.4 Trace file

A trace file (cf. figure A.1) is used to collect data and processed to evaluate simulation results. With the aid of NS2 trace file, drop or arrival of packets that occurs in the queue or in a link are recorded (157).



event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

```
r : receive (at to_node)
+ : enqueue (at queue)                src_addr : node.port (3.0)
- : dequeue (at queue)                dst_addr : node.port (0.0)
d : drop    (at queue)

         r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
         + 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
         - 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
         r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
         + 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
         d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
         + 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
         - 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Figure A.1 – *Trace format example*

Appendix B

AQUASIM

B.1 Definition

Aquasim is a simulator for underwater sensor networks which is built on top of NS2 (158). The effectiveness of Aqua-sim to simulate acoustic signal attenuation, packet collisions in underwater sensor networks, support three-dimensional deployment is discussed in(81). The advantage with Aquasim is that it is a simulation package which runs in parallel with the CMU wireless package while relying on NS2 so as to be independent of the wireless package. Figure B.1 displays the relationship between Aquasim, CMU wireless package, and NS2. Currently organized into four folders (uw common, uw mac, uw routing and uwtcl), Aqua-Sim follows the object-oriented design style of NS-2, and all network entities are implemented as classes in C++ (81).

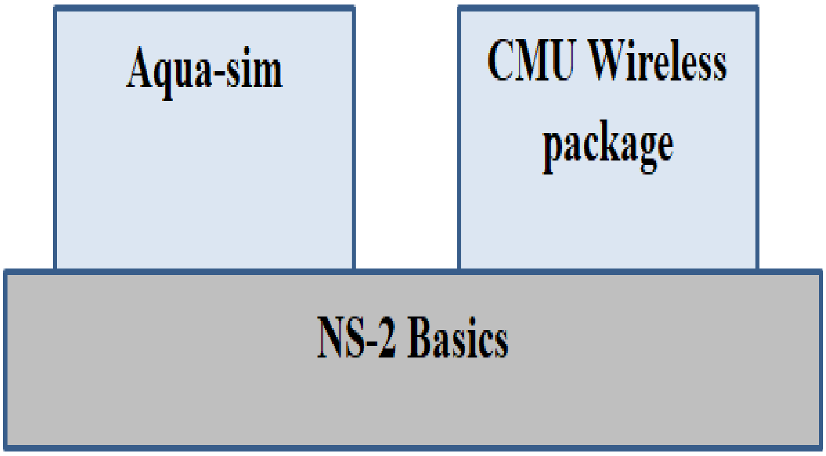


Figure B.1 – *Relationship between Aqua-sim, CMU wireless package and NS-2*

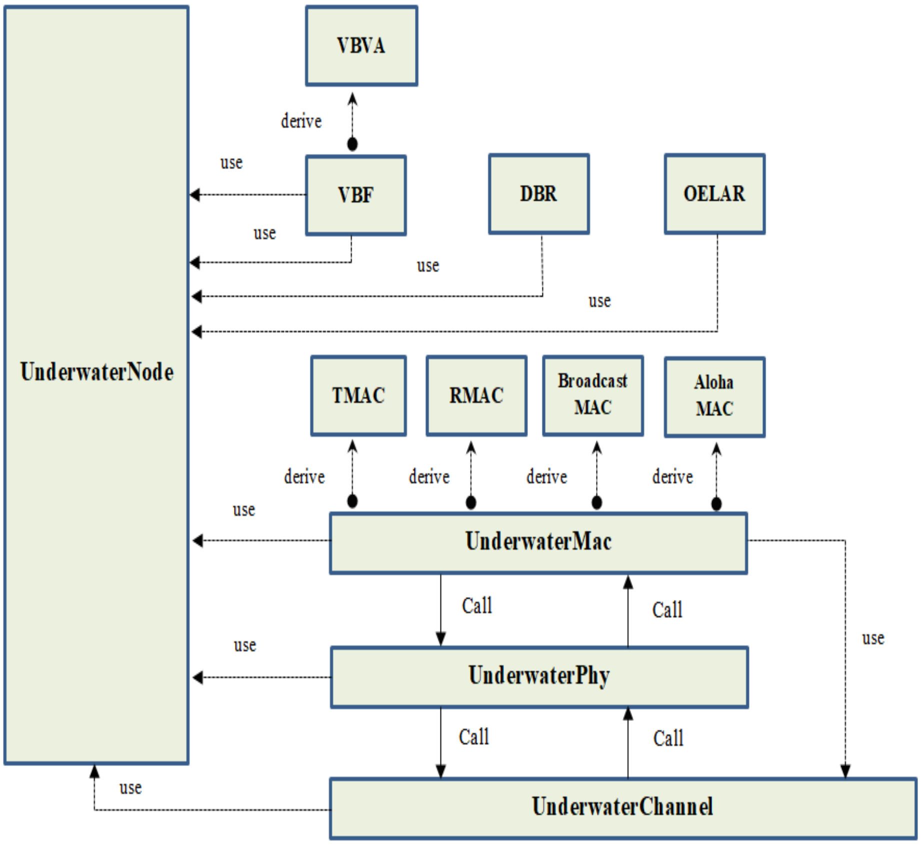


Figure B.2 – *Class diagram of Aqua-sim*

Figure B.2 shows a design of Aquasim architecture. The implementation of Aqua-Sim is briefly provided in (81) and it includes: - Physical layer models: underwaterChannel, underwaterPhy - MAC layer classes: Broadcast MAC, Aloha, Tu-MAC, and R-MAC. All these MAC protocols are derived from the same abstract base class âĂIJUnderwaterMacâĂİ - Routing layer classes: Vector based routing protocol (VBF)(61), Depth-base Routing (DBR)(62) and Q-learning-based Routing (QELAR) - Other classes: UnderwaterNode, GOD, UnderwaterSink Among all those simulation tools, Aquasim is the one that has been chosen to go further in our work.

B.2 Example of TCL code on Aquasim

The Tcl code of the simulation

This script describes a 100 nodes topology of which 12 transmitting and one TCPSink

==== General setting====

set opt(chan) Channel/UnderwaterChannel
set opt(prop) Propagation/UnderwaterPropagation
set opt(netif) Phy/UnderwaterPhy
set opt(mac) Mac/UnderwaterMac/BroadcastMac
set opt(ifq) Queue/DropTail/PriQueue
set opt(ll) LL
set opt(energy) EnergyModel
set opt(txpower) 2.0
set opt(rxpower) 1.0
set opt(initialenergy) 10000
set opt(idlepower) 0.008
set opt(ant) Antenna/OmniAntenna
set opt(ifqlen) 50 ;# max queue length in if
set opt(x) 2000 ;# X dimension of the topography
set opt(y) 2000 ;# Y dimension of the topography
set opt(z) 100
set opt(adhocRouting) DSDV

=====LINK LAYER SETTING=====
LL set mindelay$_{5}0us$
$LLsetdelay_{2}5us$
$LLsetbandwidth_{0};notused$

=====QUEUE SETTING=====
Queue/DropTail/PriQueue set Prefer$_{R}outing_{P}rotocols1$

=====ANTENNA SETTING=====
set up the antennas to be centered in the node and 1.5 meters above it
Antenna/OmniAntenna set X$_{0}$
$Antenna/OmniAntennasetY_{0}Antenna/OmniAntennasetZ_{1}.5$
$Antenna/OmniAntennasetZ_{0}.05$
$Antenna/OmniAntennasetGt_{1}.0$
$Antenna/OmniAntennasetGr_{1}.0$

=====MAC LAYER SETTING=====
Mac/UnderwaterMac set bit$_{r}ate_{1}.0e4;10kbps$
$Mac/UnderwaterMacsetencoding_{e}fficiency_{1}$

$Mac/UnderwaterMac/BroadcastMacsetpacketheader_{s}ize_{0};ofbytes$

=====PHYSICAL LAYER SETTING=====
Initialize the SharedMedia interface with parameters to make
it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/UnderwaterPhy set CPThresh$_{1}0; 10.0$
$Phy/UnderwaterPhysetCSThresh_{0};sensingrangeof200minWSN$
$Phy/UnderwaterPhysetRXThresh_{0};communicationrangeof200inWSN$
$Phy/WirelessPhysetRb_{2}*1e6$
$Phy/UnderwaterPhysetPt_{0}.2818$
$Phy/UnderwaterPhysetfreq_{2}5;25kHz$
$Phy/UnderwaterPhysetK_{2}.0;sphericalspreading$

=====TRANSPORT LAYER SETTING=====
Agent/TCP set packetSize$_{1}500$
$Agent/TCPsetwindowInit_{3}$
$Agent/TCPsetwindow_{1}2$
$Agent/TCPsetrtxcur_{i}nit_{7}$

set ns [new Simulator]
set topo [new Topography]
$topoload_{c}ubicgrid$opt(x) $opt(y)$opt(z)

Create a nam trace datafile.
set nf [open dsdv$_{n}reno12_{d}efault.namw$]

—– Setup wireless environment. —-
set tracefd [open dsdv$_{n}reno12_{d}efault.trw$]
ns trace-all $tracefd$
ns namtrace-all-wireless nfopt(x) $opt(y)$

global TN
set TN 100
set num$_{n}ode[expr$TN+1]
set god$_{[}create-god$num$_{n}ode]$
$setchan_{1[}new$opt(chan)]

global defaultRNG
$defaultRNGseed$opt(seed)
global node setting
$nsnode-config-adhocRouting$opt(adhocRouting)
-llType $opt(ll)$
$-macType$opt(mac)
-ifqType $opt(ifq)$
$-ifqLen$opt(ifqlen)
-antType $opt(ant)$
$-propType$opt(prop)
-phyType $opt(netif)$
$-channelType$opt(chan)
-agentTrace ON
-routerTrace OFF
-macTrace OFF -topoInstance $topo$
$-energyModel$opt(energy)
-txPower $opt(txpower)$
$-rxPower$opt(rxpower)
-initialEnergy $opt(initialenergy)$
$-idlePower$opt(idlepower)
-channel $chan_1$

Create underwater nodes and set position.
Setting node position block
for set i 0 i <TN incr i
set node(i)[ns node]
$node$(i) random-motion 0
$nsinitial_node_pos$node(i)50.000000

$forsetj$10j1 $< TNincrj$110$incrx$10$incry$175node$(j1)setX$x1
$node$(j1) set Zz1
$node$(j1) set Yy1
$nsat$0.000000″node$(j1)setdest$x1 y10.0″

$forsetj$21j2 $< TNincrj$210$incrx$20$incry$275node$(j2)setX$x2
$node$(j2) set Yy2
$node$(j2) set Zz2
$nsat$0.000000″node$(j2)setdest$x2 y20.0″
.

for set j10 9 $j10 <$TN incr j10 10
incr x10 0
incr y10 75
$node$(j10) set Xx10
$node$(j10) set Yy10
$node$(j10) set Zz10
$nsat0.000000$"node$(j10)setdest$x10 $y100.0$"

Set the node 100 as sink
set node(100) [$nsnode$]
node(100) random-motion 0
$nsinitial_{n}ode_{p}os$node(100) 50.000000
$node(100)setX_{4}75$
node(100) set Y_{4}62
node(100) set Z_{0}
ns at 0.000000 "$node(100)setdest475.0462.00.0$"

Set TCP agent and the application
for set j 0 $j <$TN incr j
set tcp$(j)[newAgent/TCP/Newreno]$
ns attach-agent $node$(j) tcp(j)
tcp(j) set fidj
set cbr$(j)[newApplication/Traffic/CBR]$
cbr$(j)attach - agent$tcp(j)
cbr$(j)settype_{C}BR$
cbr$(j)setpacket_{s}ize_{6}5536$
cbr$(j)setrate64kb$
cbr$(j)setrandom_{f}alse$

Set TCPSink
for set k 0 $k <$TN incr k
set tcpSink$(k)[newAgent/TCPSink]$
ns attach-agent $node(100)$tcpSink(k)

#Connect TCP traffic to the sink
$nsconnect$tcp(0) $tcpSink(0)$
ns connect $tcp(90)$tcpSink(90)
$nsconnect$tcp(9) $tcpSink(9)$
ns connect $tcp(99)$tcpSink(99)
$nsconnect$tcp(11) $tcpSink(11)$
ns connect $tcp(71)$tcpSink(71)
$nsconnect$tcp(88) $tcpSink(88)$
ns connect $tcp(28)$tcpSink(28)
$nsconnect$tcp(32) $tcpSink(32)$
ns connect $tcp(62)$tcpSink(62)
$nsconnect$tcp(67) $tcpSink(67)$
ns connect $tcp(37)$tcpSink(37)

#Schedule the simulation
$nsat$0.5000000″cbr(0) start"
$nsat$0.5000000″cbr(90) start"
$nsat$0.5000000″cbr(9) start"
$nsat$0.5000000″cbr(99) start"
$nsat$0.5000000″cbr(11) start"
$nsat$0.5000000″cbr(71) start"
$nsat$0.5000000″cbr(88) start"
$nsat$0.5000000″cbr(28) start"
$nsat$0.5000000″cbr(32) start"
$nsat$0.5000000″cbr(62) start"
$nsat$0.5000000″cbr(67) start"
$nsat$0.5000000″cbr(37) start"
$nsat$295.000000″cbr(0) stop"
$nsat$295.000000″cbr(90) stop"
$nsat$295.000000″cbr(9) stop"
$nsat$295.000000″cbr(99) stop"
$nsat$295.000000″cbr(11) stop"
$nsat$295.000000″cbr(71) stop"
$nsat$295.000000″cbr(88) stop"
$nsat$295.000000″cbr(28) stop"
$nsat$295.000000″cbr(32) stop"
$nsat$295.000000″cbr(62) stop"
$nsat$295.000000″cbr(67) stop"

$nsat$295.000000″cbr(37) stop"

# Run the simulation

proc finish

global ns nf

$nsflush - trace$

$close$nf

exec nam dsdv$_{n}reno12_{d}efault.nam$

$exit$0

#$callthefinishprocedureafter5miniutesofsimulatedtime$

Appendix C

Appendix: derivation of the gradient of diversity

From equation 4.1, it is easy to obtain

$$\frac{\partial L_{i,j}}{\partial x_i} = -\frac{1}{2\sigma_x^2}(x_i - x_j)L_{i,j}$$

Therefore

$$\frac{\partial L}{\partial x_i} = \frac{1}{\sigma_x^2} L \odot \frac{dG}{dx_i}$$

We apply the Jacob formula:

$$\begin{aligned}\frac{\partial \det(L)}{\partial x_i} &= det(L)tr(L^{-1}\frac{\partial L}{\partial x_i}) \\ &= det(L)tr(L^{-1}\frac{1}{\sigma_s^2}L \odot \frac{dG}{dx_i}) \\ &= \frac{det(L)}{\sigma_s^2}tr(L^{-1}L \odot \frac{dG}{dx_i}) \\ &= \frac{det(L)}{\sigma_s^2}tr(L^{-1}L \odot \frac{dG}{dx_i})\end{aligned}$$

and we got:

$$\frac{\partial \det(L)}{\partial x_i} = \frac{det(L)}{\sigma_s^2}tr(L \odot L^{-T}\frac{dG}{dx_i}) \tag{C.1}$$

We used that the diagonal entries of $(A \circ B)C^T$ and $(A \circ C)B^T$ coincide (159) and we used too the fact that $\frac{dG}{dx_i}^T = \frac{dG}{dx_i}$ because of symmetric.

The matrix $R = \left(L \odot L^{-T}\right)$ is commonly known the field of control theory as the relative gain array and admits many applications in the latter field .

Similarly:

$$\frac{\partial \det(L)}{\partial z_i} = \frac{det(L)}{\sigma_d^2} tr(L \odot L^{-T} \frac{dG}{dz_i}) \tag{C.2}$$

Appendix D

Appendix: Proof of the convergence of the gradient controller

We define the gradient controller as:

$$F_\zeta = H - \zeta det(L) \tag{D.1}$$

To prove that our gradient controller (equation D.1) converges to a critical point of F_ζ, we must verify the following 4 properties:

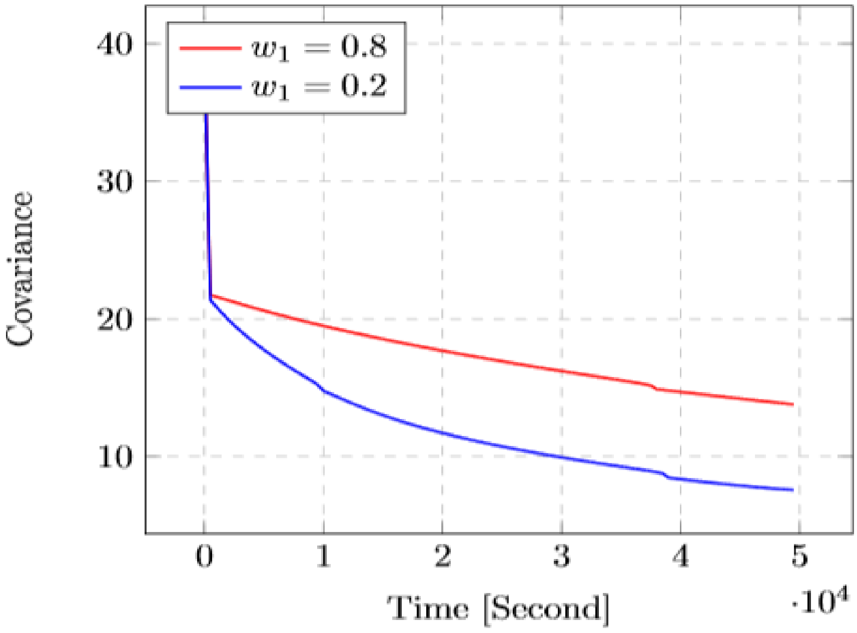
1. Must be differential;
2. Must be locally Lipschitz;
3. Must have a lower bound;
4. Must be radially unbounded or the trajectories of the system must be bounded.

While this assures convergence to a critical point of F_ζ , small perturbations to the system will cause the gradient controller to converge to a local minimum and not a local maximum or saddle point of the cost function (160).

H and $det(L)$ verify properties 1, 2, 3 and 4.
We use also the sum of two Lipschitz functions is Lipschitz.
Therefore F_ζ verify all the 4 properties (121).

D.1 Environment 2

Our second environment is obtained from (122) and is characterised by $\sigma_s = 1.97786755059$ as covariance according to X and $\sigma_d = 1.19859333137$ as covariance according to Z. We obtain similar results to environment 1.



vcaptionCovariance for $w_1 = 0.2$ and $w_1 = 0.8$

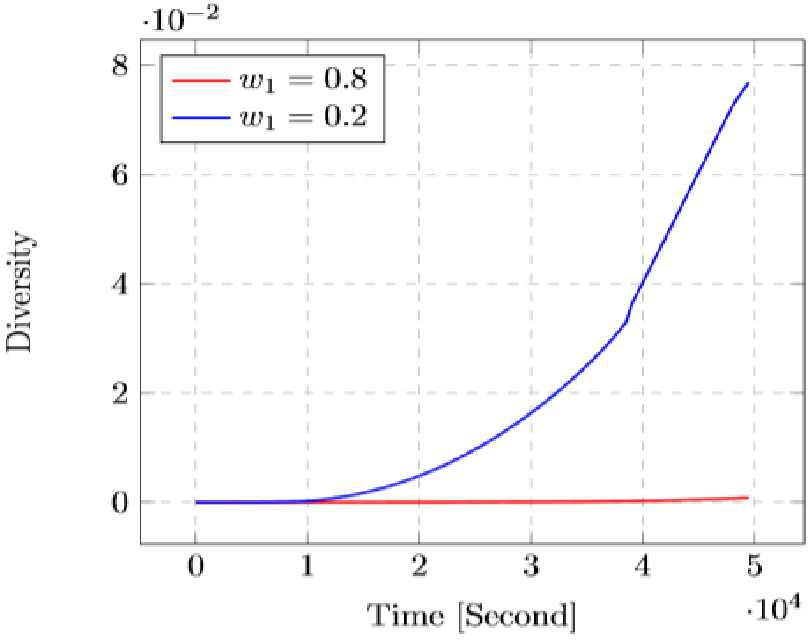


Figure D.1 – *Diversity for* $w_1 = 0.2$ *and* $w_1 = 0.8$
Figure D.2 – *Diversity for* $\omega = 0.8$ *and* $\omega = 0.2$ *and 10 sensors*

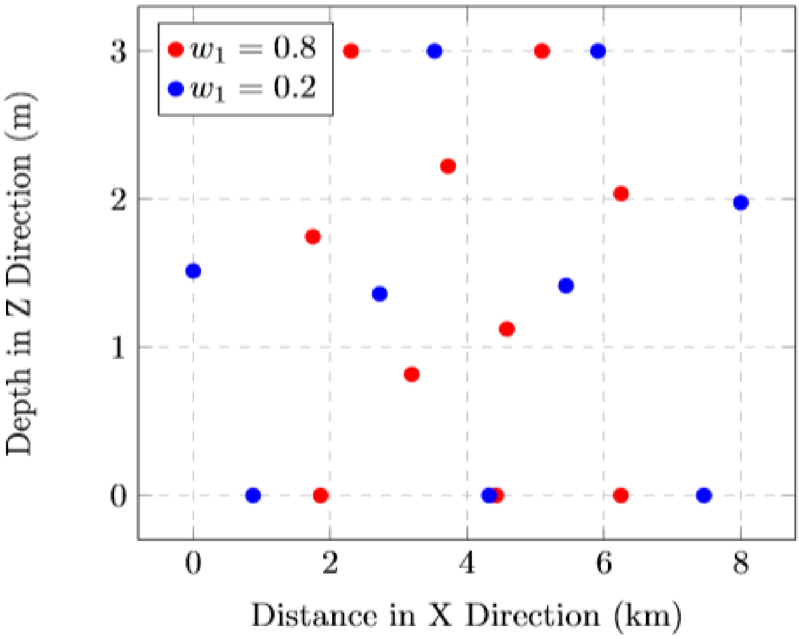


Figure D.3 – *Final Positions for* $w_1 = 0.2$ *and* $w_1 = 0.8$

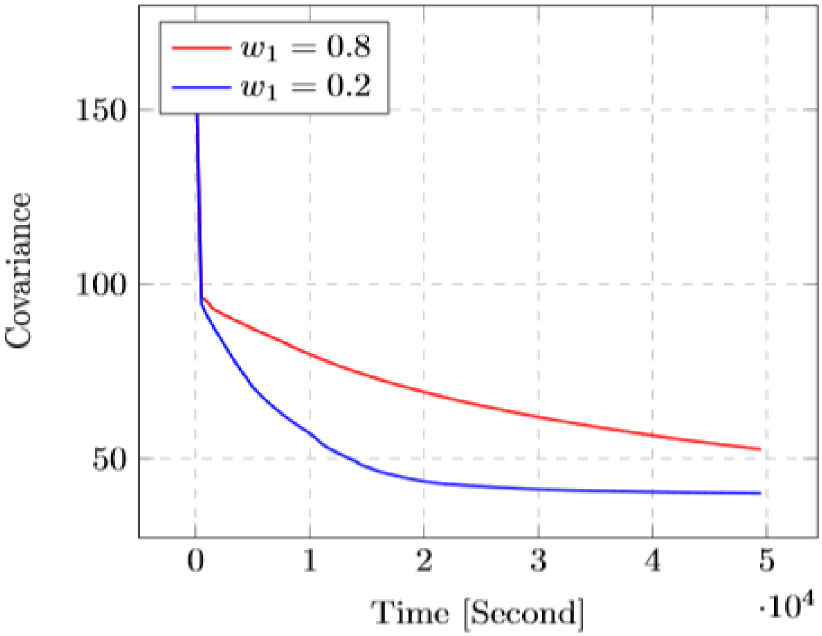


Figure D.4 – *Covariance for* $w_1 = 0.2$ *and* $w_1 = 0.8$

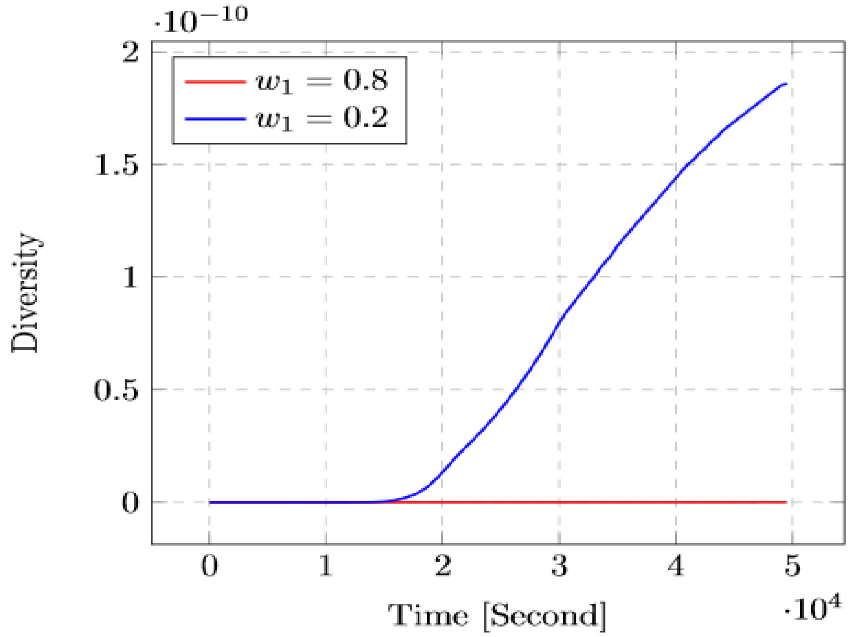


Figure D.5 – *Diversity for $\omega = 0.8$ and $\omega = 0.2$ and 20 sensors*

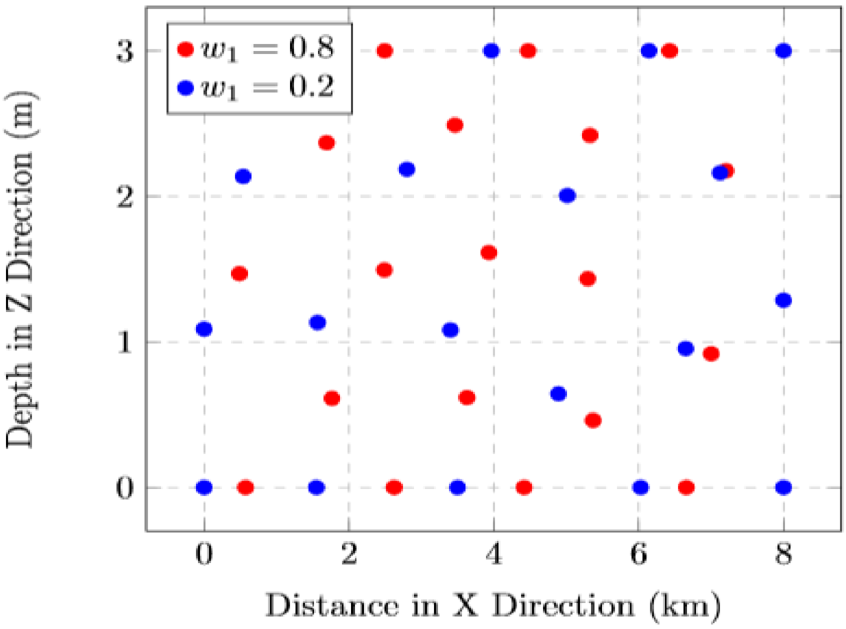


Figure D.6 – *Final Positions for $w_1 = 0.2$ and $w_1 = 0.8$*
Figure D.7 – *Final Positions for $\omega = 0.8$ and $\omega = 0.2$ and 20 sensors*

Bibliography

[1] B. Hyland, "Extending internet of things (iot) to subsea operations," 2018, noia Conference Presentation.

[2] P. Budny *et al.*, "Análise da aplicação de veículos subaquáticos não tripulados em atividades de inspeção marítima," 2018.

[3] H. Bennouri and A. Berqia, "The impact of tcp packet size and number of tcp connections in underwater wireless sensor networks," in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. IEEE, 2018, pp. 121–126.

[4] I. F. Akyildiz, D. Pompili, and T. Melodia, "State-of-the-art in protocol research for underwater acoustic sensor networks," in *Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 7–16.

[5] E. Danson, "The economics of scale: Using autonomous underwater vehicles (auvs) for wide-area hydrographic survey and ocean data acquisition," in *FIG XXII International Congress, Washington, Dc USA*, 2002.

[6] M. C. Domingo, "An overview of the internet of underwater things," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, 2012.

[7] G. Barrenetxea, F. Ingelrest, Y. M. Lu, and M. Vetterli, "Assessing the challenges of environmental signal processing through the sensorscope project," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 5149–5152.

[8] A. BANANA, M. Kaddi *et al.*, "Vers une plate-forme efficace en énergie pour les réseaux de capteurs sans fil," Ph.D. dissertation, Université Ahmed Draia-Adrar, 2018.

[9] K. Yang, "Wireless sensor networks," *Principles, Design and Applications*, 2014.

[10] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, S. Son, and G. J. Pappas, "Atpc: adaptive transmission power control for wireless sensor net-works," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 1, p. 6, 2016.

[11] X. M. Zhang, Y. Zhang, F. Yan, and A.-n. V. Vasilakos, "Interference-based topology control algorithm for delay-constrained mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 742–754, 2015.

[12] S. P. Singh and S. Sharma, "A survey on cluster based routing protocols in wireless sensor networks," *Procedia computer science*, vol. 45, pp. 687–695, 2015.

[13] e. Antonopoulos, "Integrated toolset for wsn application planning, development, commissioning and maintenance: The wsn-dpcm artemis-ju project," *Sensors*, vol. 16, no. 6, p. 804, 2016.

[14] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: A survey," *Computers & Electrical Engineering*, vol. 41, pp. 159–176, 2015.

[15] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computers and Electronics in Agriculture*, vol. 118, pp. 66–84, 2015.

[16] S. Gajjar, N. Choksi, M. Sarkar, and K. Das-gupta, "Comparative analysis of wireless sensor network motes," in *Signal Processing and Integrated Networks (SPIN), 2014 Inter-national Conference on*. IEEE, 2014, pp. 426–431.

[17] Á. F. García-Fernández, L. Svensson, and S. Särkkä, "Cooperative localization using posterior linearization belief propagation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 832–836, 2018.

[18] F. Palumbo, P. Barsocchi, S. Chessa, and J. C. Augusto, "A stigmergic approach to indoor localization using bluetooth low energy beacons," in *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2015, pp. 1–6.

[19] T. A. Ngo, M. Tummala, and J. C. McEachen, "Wireless signal localization and collection from an airborne sym-metric line array network," Oct. 23 2018, uS Patent App. 10/107,891.

[20] R. W. Coutinho, A. Boukerche, L. F. Vieira, and A. A. Loureiro, "A novel void node recovery paradigm for long-term underwater sen-sor networks," *Ad Hoc Networks*, vol. 34, pp. 144–156, 2015.

[21] S. K. Sarkar, T. G. Basavaraju, and C. Puttama-dappa, *Ad hoc mobile wireless networks: principles, protocols, and appli-cations*. CRC Press, 2016.

[22] E. Fadel, V. C. Gungor, L. Nassef, N. Akkari, M. A. Malik, S. Almasri, and I. F. Akyildiz, "A survey on wireless sensor networks for smart grid," *Computer Communications*, vol. 71, pp. 22–33, 2015.

[23] M. Hamdi, N. Boudriga, and M. S. Obaidat, "Whomoves: An optimized broadband sensor network for military vehicle tracking," *International Journal of Communication Systems*, vol. 21, no. 3, pp. 277–300, 2008.

[24] T. I. Nagy and J. Tick, "Intelligent sensor networks in the military and civil sectors," in *2009 5th International Symposium on Applied Computational Intelligence and Informatics*. IEEE, 2009, pp. 471–474.

[25] D. M. Davenport, B. Deb, and F. J. Ross, "Wireless propagation and coexistence of medical body sensor networks for ambulatory patient monitoring," in *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*. IEEE, 2009, pp. 41–45.

[26] M. R. Yuce, P. C. Ng, and J. Y. Khan, "Monitoring of physiological parameters from multiple patients using wireless sensor network," *Journal of medical systems*, vol. 32, no. 5, pp. 433–441, 2008.

[27] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–32, 2010.

[28] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Out-of-the-box environmental monitoring," in *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*. IEEE, 2008, pp. 332–343.

[29] V. Tsetsos, G. Alyfantis, T. Hasiotis, O. Sekkas, and S. Hadjiefthymiades, "Commercial wireless sensor networks: technical and business issues," in *Second Annual Conference on Wireless On-demand Network Systems and Services*. IEEE, 2005, pp. 166–173.

[30] E. M. Petriu, N. D. Georganas, D. C. Petriu, D. Makrakis, and V. Z. Groza, "Sensor-based information appliances," *IEEE Instrumentation & Measurement Magazine*, vol. 3, no. 4, pp. 31–35, 2000.

[31] L. Lionel, "Underwater robots part i: Current systems and problem pose," in *Mobile Robots: towards New Applications*. IntechOpen, 2006.

[32] J.-M. Spiewak, "Contribution à la coordination de flottille de véhicules sous-marins autonomes," Theses, Université Montpellier II - Sciences et Techniques du Languedoc, Sep. 2007. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00195079

[33] J. Michel and R. Duranton, "Etat de l'art des véhicules sous-marins autonomes, verrous technologiques," in *Intervention Sous-Marine ISM 90, Toulon (France), 3-5 Dec 1990*, 1990.

[34] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," *IEEE journal of oceanic engineering*, vol. 25, no. 1, pp. 72–83, 2000.

[35] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[36] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater survelliance systems," *Ad Hoc Networks*, vol. 4, no. 4, pp. 431–446, 2006.

[37] T. K. Podder, M. Sibenac, and J. G. Bellingham, "Applications and challenges of auv docking systems deployed for long-term science missions," *Monterey Bay Aquarium Research Institute*, 2019.

[38] X. Chen, J. Chase, and Y. Chen, *Mobiles Robots-Past Present and Future*. INTECH Open Access Publisher, 2009.

[39] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "The challenges of building mobile underwater wireless networks for aquatic applications," *Ieee Network*, vol. 20, no. 3, pp. 12–18, 2006.

[40] A. A. Nimbalkar and D. Pompili, "Reliability in underwater inter-vehicle communications," in *Proceedings of the third ACM international workshop on Underwater Networks*, 2008, pp. 19–26.

[41] D. O. Jones, A. R. Gates, V. A. Huvenne, A. B. Phillips, and B. J. Bett, "Autonomous marine environmental monitoring: Application in decommissioned oil fields," *Science of the total environment*, vol. 668, pp. 835–853, 2019.

[42] N. Soreide, C. Woody, and S. Holt, "Overview of ocean based buoys and drifters: present applications and future needs," in *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, vol. 4. IEEE, 2001, pp. 2470–2472.

[43] A. Celik, N. Saeed, T. Y. Al-Naffouri, and M.-S. Alouini, "Modeling and performance analysis of multihop underwater optical wireless sensor networks," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[44] H. Bennouri, A. Yazidi, and A. Berqia, "A pursuit learning solution to underwater communications with limited mobility agents," in *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*. ACM, 2018, pp. 112–117.

[45] L. G. Stolarczyk, G. L. Stolarczyk, and I. Bausov, "Underground radio communications and personnel tracking system," Feb. 14 2012, uS Patent 8,115,622.

[46] F. Schill, U. R. Zimmer, and J. Trumpf, "Visible spectrum optical communication and distance sensing for underwater applications," in *Proceedings of ACRA*, vol. 2004, 2004, pp. 1–8.

[47] T. Melodia, H. Kulhandjian, L.-C. Kuo, and E. Demirors, "Advances in underwater acoustic networking," *Mobile ad hoc networking: Cutting edge directions*, pp. 804–852, 2013.

[48] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks," *ACM Sigbed Review*, vol. 1, no. 2, pp. 3–8, 2004.

[49] H. W. Austad, "Simulation of subsea communication network," Master's thesis, 2014.

[50] R. Manjula and S. S. Manvi, "Issues in underwater acoustic sensor networks," *International Journal of Computer and Electrical Engineering*, vol. 3, no. 1, p. 101, 2011.

[51] S. Porretta, M. Barbeau, J. Garcia-Alfaro, and E. Kranakis, "Learning to communicate underwater: An exploration of limited mobility agents," in *Proceedings of the International Conference on Underwater Networks & Systems*. ACM, 2017, p. 2.

[52] A. Mateen, M. Awais, N. Javaid, F. Ishmanov, M. K. Afzal, and S. Kazmi, "Geographic and opportunistic recovery with depth and power transmission adjustment for energy-efficiency and void hole alleviation in uwsns," *Sensors*, vol. 19, no. 3, p. 709, 2019.

[53] S. Floyd, T. Henderson, and A. Gurtov, "The newreno modification to tcp's fast recovery algorithm," Tech. Rep., 2004.

[54] A. Chavan, D. Kurule, and P. Dere, "Performance analysis of aodv and dsdv routing protocol in manet and modifications in aodv against black hole attack," *Procedia Computer Science*, vol. 79, pp. 835–844, 2016.

[55] A. Sharma and R. Kumar, "Performance comparison and detailed study of aodv, dsdv, dsr, tora and olsr routing protocols in ad hoc networks," in *2016 Fourth Inter-*

national Conference on Parallel, Distributed and Grid Computing (PDGC). IEEE, 2016, pp. 732–736.

[56] A. Daas, K. Mofleh, E. Jabr, and S. Hamad, "Comparison between aodv and dsdv routing protocols in mobile ad-hoc network (manet)," in *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*. IEEE, 2015, pp. 1–5.

[57] C. Perkins, E. Belding-Royer, and S. Das, "Rfc3561: Ad hoc on-demand distance vector (aodv) routing," 2003.

[58] G. He, "Destination-sequenced distance vector (dsdv) protocol," *Networking Laboratory, Helsinki University of Technology*, pp. 1–9, 2002.

[59] M. U. Chowdhury, D. Perera, and T. Pham, "A performance comparison of three wireless multi hop ad-hoc network routing protocols when streaming mpeg4 traffic," in *8th International Multitopic Conference, 2004. Proceedings of INMIC 2004.* IEEE, 2004, pp. 516–521.

[60] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol (olsr)," 2003.

[61] P. Xie, J.-H. Cui, and L. Lao, "Vbf: vector-based forwarding protocol for underwater sensor networks," in *International conference on research in networking*. Springer, 2006, pp. 1216–1221.

[62] H. Yan, Z. J. Shi, and J.-H. Cui, "Dbr: depth-based routing for underwater sensor networks," in *International conference on research in networking*. Springer, 2008, pp. 72–86.

[63] S. Zhang, D. Li, and J. Chen, "A link-state based adaptive feedback routing for underwater acoustic sensor networks," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4402–4412, 2013.

[64] M. Tariq, M. S. Latiff, M. Ayaz, Y. Coulibaly, and N. Al-Areqi, "Distance based reliable and energy efficient (dree) routing protocol for underwater acoustic sensor networks," *Journal of Networks*, vol. 10, no. 5, p. 311, 2015.

[65] A. Wahid, S. Lee, and D. Kim, "A reliable and energy-efficient routing protocol for underwater wireless sensor networks," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2048–2062, 2014.

[66] N. Javaid, N. Ilyas, A. Ahmad, N. Alrajeh, U. Qasim, Z. A. Khan, T. Liaqat, and M. I. Khan, "An efficient data-gathering routing protocol for underwater wireless sensor networks," *Sensors*, vol. 15, no. 11, pp. 29 149–29 181, 2015.

[67] T. Ali, L. T. Jung, and I. Faye, "End-to-end delay and energy efficient routing protocol for underwater wireless sensor networks," *Wireless Personal Communications*, vol. 79, no. 1, pp. 339–361, 2014.

[68] M. Faheem, G. Tuna, and V. C. Gungor, "Qerp: Quality-of-service (qos) aware evolutionary routing protocol for underwater wireless sensor networks," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2066–2073, 2017.

[69] H. Jamal and K. Sultan, "Performance analysis of tcp congestion control algorithms," *International journal of computers and communications*, vol. 2, no. 1, pp. 18–24, 2008.

[70] S. Floyd, T. Henderson, and A. Gurtov, "Rfc3782: The newreno modification to tcp's fast recovery algorithm," 2004.

[71] R. A. Hamamreh and M. J. Bawatna, "Protocol for dynamic avoiding end-to-end congestion in manets," *Journal of Wireless Networking and Communications*, vol. 4, no. 3, pp. 67–75, 2014.

[72] C.-Y. Huang, P. Ramanathan, and K. Saluja, "Routing tcp flows in underwater mesh networks," *IEEE journal on selected areas in communications*, vol. 29, no. 10, pp. 2022–2032, 2011.

[73] S. Jiang, "On reliable data transfer in underwater acoustic networks: A survey from networking perspective," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1036–1055, 2018.

[74] M. Jafari, A. Alsadoon, C. Withana, S. Ali, and A. Elchouemic, "Segment based model for tcp protocol optimization: Enhancing the bandwidth and congestion," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018, pp. 918–924.

[75] M. Albuquerque, J. H. Kim, and S. Roy, "Effect of packet size on tcp-reno performance over lossy, congested links," in *2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No. 01CH37277)*, vol. 1. IEEE, 2001, pp. 705–710.

[76] R.-Q. Wu, H. Jie, and N. Ding, "An improved tcp congestion control algorithm of based on bandwidth estimation in heterogeneous networks.[j]," *Journal of Communications*, vol. 9, no. 10, pp. 792–797, 2014.

[77] E. Vieira and M. Bauer, "Round-trip time variation in smoothtcp in the face of spurious errors," *Journal of Communications*, vol. 35, no. 2, pp. 48–56, 2006.

[78] N. Iikubo, M. Nakatsuka, J. Katto, and H. Kondo, "Improving tcp performance over underwater sensor networks," in *Proceedings of the International Workshop on UnderWater Networks*, 2008.

[79] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "Tcp vegas: New techniques for congestion detection and avoidance," in *Proceedings of the conference on Communications architectures, protocols and applications*, 1994, pp. 24–35.

[80] M. Singh, S.-G. Lee, D. Singh, and H. J. Lee, "Impact and performance of mobility models in wireless ad-hoc networks," in *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*. IEEE, 2009, pp. 139–143.

[81] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, and S. Zhou, "Aquasim: An ns-2 based simulator for underwater sensor networks," in *OCEANS 2009, MTS/IEEE biloxi-marine technology for our future: global and local challenges*. IEEE, 2009, pp. 1–7.

[82] T. Kuang, F. Xiao, and C. Williamson, "Diagnosing wireless tcp performance problems: A case study," in *In Proc. of SPECTS*. Citeseer, 2003.

[83] M. Allman, V. Paxson, and E. Blanton, "Tcp congestion control," Tech. Rep., 2009.

[84] H. Bennouri, A. Berqia, and N. K. Patrick, "Controlling maximum window of tcp newreno in underwater wireless sensor networks," in *2018 International Symposium on Advanced Electrical and Communication Technologies(ISAECT)*. IEEE, 2018.

[85] S. MCanne, "ns-lbl network simulator, 1997," *Obtain via http://www-nrg.ee.lbnl.gov/ns*.

[86] Y. Tian, K. Xu, and N. Ansari, "Tcp in wireless environments: problems and solutions," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S27–S32, 2005.

[87] H. Bennouri, A. Berqia, and N. K. Patrick, "Adapting the appropriate RTT timeout of TCP newreno in submarine communication networks," *JCM*, vol. 14, no. 12, pp. 1191–1197, 2019. [Online]. Available: https://doi.org/10.12720/jcm.14.12.1191-1197

[88] Y. Zhu, X. Lu, L. Pu, Y. Su, R. Martin, M. Zuba, Z. Peng, and J.-H. Cui, "Aquasim: an ns-2 based simulator for underwater sensor networks," *Proceedings of ACM WUWNet*, 2013.

[89] X. Liu, P. Liu, T. Long, Z. Lv, and R. Tang, "An efficient depth-based forwarding protocol for underwater wireless sensor networks," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, 2018, pp. 467–475.

[90] R. Otnes and S. Haavik, "Duplicate reduction with adaptive backoff for a flooding-based underwater network protocol," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–6.

[91] J. U. Robert, *Principles Of Underwater Sound.* Peninsula Publishing, 2013.

[92] J. Luo, L. Fan, S. Wu, and X. Yan, "Research on localization algorithms based on acoustic communication for underwater sensor networks," *Sensors*, vol. 18, no. 1, p. 67, 2018.

[93] B. J. Oommen, "Stochastic searching on the line and its applications to parameter learning in nonlinear optimization," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-27B, pp. 733–739, 1997.

[94] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007.

[95] A. H. Bass and C. W. Clark, "The physical acoustics of underwater sound communication," in *Acoustic communication*. Springer, 2003, pp. 15–64.

[96] J. A. Catipovic, "Performance limitations in underwater acoustic telemetry," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 3, pp. 205–216, 1990.

[97] K. S. Narendra and P. Mars, "The use of learning algorithms in telephone traffic routing: A methodology," *Automatica*, vol. 19, no. 5, pp. 495–502, 1983.

[98] K. S. Narendra and M. A. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 5, pp. 262–269, 1980.

[99] K. S. Narendra, E. A. Wright, and L. G. Mason, "Application of learning automata to telephone traffic routing and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 11, pp. 785–792, 1977.

[100] P. C. Etter, *Underwater acoustic modeling and simulation.* CRC Press, 2018.

[101] A. Yazidi, O.-C. Granmo, B. J. Oommen, and M. Goodwin, "A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme," *IEEE transactions on cybernetics*, vol. 44, no. 11, pp. 2202–2220, 2014.

[102] A. Yazidi and B. J. Oommen, "Novel discretized weak estimators based on the principles of the stochastic search on the line problem," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 2732–2744, 2016.

[103] A. Yazidi and H. Hammer, "Multiplicative update methods for incremental quantile estimation," *IEEE Transactions on Cybernetics*, 2017.

[104] A. Yazidi, H. Hammer, and B. J. Oommen, "Higher-fidelity frugal and accurate quantile estimation using a novel incremental discretized paradigm," *IEEE Access*, vol. 6, pp. 24 362–24 374, 2018.

[105] A. Yazidi and B. J. Oommen, "A novel technique for stochastic root-finding: Enhancing the search with adaptive d-ary search," *Information Sciences*, vol. 393, pp. 108–129, 2017.

[106] A. Yazidi and H. L. Hammer, "Solving stochastic nonlinear resource allocation problems using continuous learning automata," *Applied Intelligence*, pp. 1–20, 2018.

[107] O.-C. Granmo and B. J. Oommen, "Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata," *IEEE Transactions on Computers*, vol. 59, no. 4, pp. 545–560, 2010.

[108] B. J. Oommen and J. K. Lanctôt, "Discretized pursuit learning automata," *IEEE Transactions on systems, man, and cybernetics*, vol. 20, no. 4, pp. 931–938, 1990.

[109] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: new families of continuous and discretized learning automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 738–749, 2002.

[110] X. Zhang, B. J. Oommen, O.-C. Granmo, and L. Jiao, "A formal proof of the ε-optimality of discretized pursuit algorithms," *Applied Intelligence*, vol. 44, no. 2, pp. 282–294, 2016.

[111] L. Mason, "An optimal learning algorithm for s-model environments," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 493–496, 1973.

[112] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. New Jersey: Prentice-Hall, 1989.

[113] E. Felemban, F. K. Shaikh, U. M. Qureshi, A. A. Sheikh, and S. B. Qaisar, "Underwater sensor network applications: A comprehensive survey," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 896832, 2015.

[114] X. Gong, J. Guo, A. Wang, D. Xu, N. An, X. Chen, D. Fang, X. Zheng *et al.*, "Dedv: A data collection method for mobile sink based on dynamic estimation of data value in wsn," in *2016 International Conference on Networking and Network Applications (NaNA)*. IEEE, 2016, pp. 77–83.

[115] S. Zhang, Q. Zhang, M. Liu, and Z. Fan, "A top-down positioning scheme for underwater wireless sensor networks," *Science China Information Sciences*, vol. 57, no. 3, pp. 1–10, 2014.

[116] D. Wang, J. Liu, J. Xu, H. Jiang, and C. Wang, "Data sweeper: A proactive filtering framework for error-bounded sensor data collection," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 4, pp. 487–501, 2015.

[117] Z. Zhang, S. Qi, and S. Li, "Marine observation beacon clustering and recycling technology based on wireless sensor networks," *Sensors*, vol. 19, no. 17, p. 3726, 2019.

[118] M. Erol, L. F. Vieira, and M. Gerla, "Localization with dive'n'rise (dnr) beacons for underwater acoustic sensor networks," in *Proceedings of the second workshop on Underwater networks*, 2007, pp. 97–100.

[119] M. Karpagam and D. Prabha, "Underwater wireless sensor network based marine environment monitoring system," *International Journal of Oceans and Oceanography*, vol. 13, no. 2, pp. 269–276, 2019.

[120] Y. Chen and D. Han, "Water quality monitoring in smart city: A pilot project," *Automation in Construction*, vol. 89, pp. 307–316, 2018.

[121] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 2542–2547.

[122] C. Detweiler, M. Doniec, M. Jiang, M. Schwager, R. Chen, and D. Rus, "Adaptive decentralized control of underwater sensor networks for modeling underwater phenomena," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 253–266.

[123] D. Karimanzira, M. Jacobi, T. Pfützenreuter, T. Rauschenbach, M. Eichhorn, R. Taubert, and C. Ament, "First testing of an auv mission planning and guidance system for water quality monitoring and fish behavior observation in net cage fish farming," *Information Processing in Agriculture*, vol. 1, no. 2, pp. 131–140, 2014.

[124] S. Tang and L. Yang, "Morello: A quality-of-monitoring oriented sensing scheduling protocol in sensor networks," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2676–2680.

[125] S. Tang and J. Yuan, "Damson: on distributed sensing scheduling to achieve high quality of monitoring," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 155–159.

[126] C. Detweiler, S. Banerjee, M. Doniec, M. Jiang, F. Peri, R. Chen, and D. Rus, "Adaptive decentralized control of mobile underwater sensor networks and robots for modeling underwater phenomena," *Journal of Sensor and Actuator Networks*, vol. 3, no. 2, pp. 113–149, 2014.

[127] S. A. Ali, E. Hall, M. Ghaffari, and X. Liao, *Mobile robotics, moving intelligence*. INTECH Open Access Publisher, 2006.

[128] Y. Jin, H. Guo, and Y. Meng, "A hierarchical gene regulatory network for adaptive multirobot pattern formation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 805–816, 2012.

[129] Y. He, L. Zhu, G. Sun, J. Qiao, and S. Guo, "Underwater motion characteristics evaluation of multi amphibious spherical robots," *Microsystem Technologies*, vol. 25, no. 2, pp. 499–508, 2019.

[130] C. D. Makavita, H. D. Nguyen, S. G. Jayasinghe, and D. Ranmuthugala, "Predictor-based model reference adaptive control of an unmanned underwater vehicle," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2016, pp. 1–7.

[131] F. Eren, S. PeâĂŹeri, M.-W. Thein, Y. Rzhanov, B. Celikkol, and M. R. Swift, "Position, orientation and velocity detection of unmanned underwater vehicles (uuvs) using an optical detector array," *Sensors*, vol. 17, no. 8, p. 1741, 2017.

[132] M. Anirban and H. Asada, "Control-configured design of spheroidal, appendage-free, underwater vehicle," *IEEE Trans Robot*, vol. 30, no. 2, pp. 448–460, 2014.

[133] H. Guo, Y. Meng, and Y. Jin, "Self-adaptive multi-robot construction using gene regulatory networks," in *2009 IEEE Symposium on Artificial Life*. IEEE, 2009, pp. 53–60.

[134] ——, "Analysis of local communication load in shape formation of a distributed morphogenetic swarm robotic system," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.

[135] H. Guo, Y. Jin, and Y. Meng, "A morphogenetic framework for self-organized multi-robot pattern formation and boundary coverage," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7, no. 1, pp. 1–23, 2012.

[136] T. Taylor, P. Ottery, and J. Hallam, "Pattern formation for multi-robot applications: Robust, self-repairing systems inspired by genetic regulatory networks and cellular self-organisation," *University of Edinburgh., Tech. Rep. EDI-INF-RR-0971*, 2007.

[137] J. Gillenwater, A. Kulesza, and B. Taskar, "Near-optimal map inference for determinantal point processes," in *Advances in Neural Information Processing Systems*. Citeseer, 2012, pp. 2744–2752.

[138] M. Ilyas and I. Mahgoub, *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press, 2004.

[139] S. Misra and D. Mohanta, "Adaptive listen for energy-efficient medium access control in wireless sensor networks," *Multimedia Tools and Applications*, vol. 47, no. 1, pp. 121–145, 2010.

[140] A. Majid and et al., "An energy efficient and balanced energy consumption cluster based routing protocol for underwater wireless sensor networks," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016, pp. 324–333.

[141] C. Ramachandran, S. Misra, and M. S. Obaidat, "A probabilistic zonal approach for swarm-inspired wildfire detection using sensor networks," *International Journal of Communication Systems*, vol. 21, no. 10, pp. 1047–1073, 2008.

[142] Y. Yao, Q. Cao, and A. V. Vasilakos, "Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks," *IEEE/ACM transactions on networking*, vol. 23, no. 3, pp. 810–823, 2014.

[143] R. Mohammadi, R. Javidan, and A. Jalili, "Fuzzy depth based routing protocol for underwater acoustic wireless sensor networks," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 7, no. 1, pp. 81–86, 2015.

[144] F. Zabin, S. Misra, I. Woungang, H. F. Rashvand, N.-W. Ma, and M. A. Ali, "Reep: data-centric, energy-efficient and reliable routing protocol for wireless sensor networks," *IET communications*, vol. 2, no. 8, pp. 995–1008, 2008.

[145] ur Rahman and et al., "On utilizing static courier nodes to achieve energy efficiency with depth based routing for underwater wireless sensor networks," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016, pp. 1184–1191.

[146] S. Dhurandher, S. Misra, M. Obaidat, and S. Khairwal, “Uwsim: an underwater sensor network simulator,” *SIMULATION: Transactions of the Society for Modeling and Simulation International*, vol. 84, no. 7, pp. 327–338, 2008.

[147] K. M. Pouryazdanpanah, M. Anjomshoa, S. A. Salehi, A. Afroozeh, and G. M. Moshfegh, “Ds-vbf: Dual sink vector-based routing protocol for underwater wireless sensor network,” in *2014 IEEE 5th Control and System Graduate Research Colloquium*. IEEE, 2014, pp. 227–232.

[148] A. Wahid, S. Lee, H.-J. Jeong, and D. Kim, “Eedbr: Energy-efficient depth-based routing protocol for underwater wireless sensor networks,” in *International Conference on Advanced Computer Science and Information Technology*. Springer, 2011, pp. 223–234.

[149] I. Gupta, D. Riordan, and S. Sampalli, “Cluster-head election using fuzzy logic for wireless sensor networks,” in *3rd Annual communication networks and services research conference (CNSR’05)*. IEEE, 2005, pp. 255–260.

[150] J.-M. Kim, S.-H. Park, Y.-J. Han, and T.-M. Chung, “Chef: cluster head election mechanism using fuzzy logic in wireless sensor networks,” in *2008 10th International Conference on Advanced Communication Technology*, vol. 1. IEEE, 2008, pp. 654–659.

[151] C.-J. Huang and et al., “A power-efficient routing protocol for underwater wireless sensor networks,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2348–2355, 2011.

[152] Y. K. Tamandani, M. U. Bokhari, and Q. M. Shallal, “Two-step fuzzy logic system to achieve energy efficiency and prolonging the lifetime of wsns,” *Wireless Networks*, vol. 23, no. 6, pp. 1889–1899, 2017.

[153] N. Goyal, M. Dave, and A. K. Verma, “Energy efficient architecture for intra and inter cluster communication for underwater wireless sensor networks,” *Wireless Personal Communications*, vol. 89, no. 2, pp. 687–707, 2016.

[154] Z. Guo, B. Wang, P. Xie, W. Zeng, and J.-H. Cui, “Efficient error recovery with network coding in underwater sensor networks,” *Ad Hoc Networks*, vol. 7, no. 4, pp. 791–802, 2009.

[155] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

[156] K. Fall and K. Varadhan, “ns notes and documentation. the vint project, uc berkeley, lbl, usc/isi, and xerox parc, november 1997,” 1997.

[157] Y. M. HASAN, “Communication and computer networks simulator (ns2).”

[158] T. King, “A simulator for marine wireless sensor networks,” *Graduate Thesis for the degree of Master of Science in Computer Science*, 2011.

[159] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[160] M. Schwager, “A gradient optimization approach to adaptive multi-robot control,” MASSACHUSETTS INST OF TECH CAMBRIDGE, Tech. Rep., 2009.