



**Michigan  
Technological  
University**

Michigan Technological University  
**Digital Commons @ Michigan Tech**

---

Dissertations, Master's Theses and Master's Reports

---

2020

# IMPLEMENTATION OF A CHANNEL-AWARE ROUTING PROTOCOL IN THE NETWORK SIMULATOR FOR UNDERWATER ACOUSTIC COMMUNICATION NETWORKING

Adesoji Bello

*Michigan Technological University, bello@mtu.edu*

Copyright 2020 Adesoji Bello

---

## Recommended Citation

Bello, Adesoji, "IMPLEMENTATION OF A CHANNEL-AWARE ROUTING PROTOCOL IN THE NETWORK SIMULATOR FOR UNDERWATER ACOUSTIC COMMUNICATION NETWORKING", Open Access Master's Report, Michigan Technological University, 2020.  
<https://doi.org/10.37099/mtu.dc.etdr/1119>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etdr>



Part of the [Systems and Communications Commons](#)

# **IMPLEMENTATION OF A CHANNEL-AWARE ROUTING PROTOCOL IN THE NETWORK SIMULATOR FOR UNDERWATER ACOUSTIC COMMUNICATION NETWORKING**

By

Adesoji Bello

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical and Computer Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2020

© 2020 Adesoji Bello

This report has been approved in partial fulfillment of the requirements for the Degree of  
MASTER OF SCIENCE in Electrical and Computer Engineering.

Department of Electrical and Computer Engineering

Report Advisor: *Dr. Zhaohui Wang*

Committee Member: *Dr. Aurenice M. Oliveira*

Committee Member: *Dr. Anthony Pinar*

Department Chair: *Dr. Glen E. Archer*

# **Table of Contents**

List of figures .....	iv
Acknowledgements .....	vi
List of Abbreviations .....	vii
Abstract .....	viii
1. Introduction .....	1
1.1. Overview of UWSN .....	1
1.2. Contributions .....	2
2. Related Work .....	3
3. Overview of the CARP Protocol .....	6
3.1. Description of Channel-aware Routing Protocol .....	6
3.2. Algorithm of Protocol .....	6
3.3. Description of Aqua-Sim NG .....	7
3.4. Architecture and Implementation of Aqua-Sim NG .....	8
4. Results .....	12
4.1. Overview of Results .....	12
4.1.1 Aqua-Sim Routing Header .....	12
4.1.2 Aqua-Sim Routing Header Main .....	12
4.1.3 Aqua-Sim Carp Header .....	13
4.1.4 Aqua-Sim Carp Main .....	13
4.2. Discussion of Simulation Results .....	19
5. Future Research .....	22
6. Conclusion .....	23
7. References .....	24

## List of Figures

Figure 1	Header Information of DBR	3
Figure 2	An illustration of void in Depth Based Routing	4
Figure 3	Diagrammatic representation of VBF in Underwater Sensor Networks	5
Figure 4	Flowchart of the CARP algorithm	6
Figure 5	Architecture of Aqua-Sim Next Generation	10
Figure 6	A screenshot showing build of the CARP module in Aqua-Sim-NG simulator	15
Figure 7	A picture of the successful built modules under the Aqua-Sim-NG in NS-3	15
Figure 8	Network Topology (Single Source Node)	16
Figure 9	Packet delivery ratio with varying node density using SFAMA MAC protocol	17
Figure 10	Packet delivery ratio with varying node density using ALOHA MAC protocol	17
Figure 11	Throughput with varying node density using SFAMA MAC protocol	18
Figure 12	Throughput with varying node density using ALOHA MAC protocol	18
Figure 13	Network Topology (Multiple Source Node)	19
Figure 14	Packet delivery ratio with varying node density using SFAMA MAC protocol	19
Figure 15	Packet delivery ratio with varying node density using ALOHA MAC protocol	20
Figure 16	Throughput with varying node density using SFAMA MAC protocol	20
Figure 17	Throughput with varying node density using ALOHA MAC protocol	21

## List of Tables

Table 4.1	Simulation parameters	16
-----------	-----------------------	----

## Acknowledgements

I give special thanks to Almighty God for giving me the strength and wisdom to complete my master's degree even regardless of the daunting challenges I had to go through in the course of my sojourn. In a similar vein, I would also like to appreciate the tremendous support of my parents and family in Nigeria who made tremendous sacrifice to ensure I achieve my goals in the pursuit of advancing my education. They have truly been instrumental in my success at Michigan Technological University.

Also, I would tremendously like to appreciate the efforts of my advisor Dr. Wang for her unflinching support during the development of this project. Even in the face of the pandemic which limited face-to-face interaction, she took out time routinely to ensure every technical aspect of the project which posed a challenge to me was resolved by providing ample support. I would also like to thank my committee members Dr. Oliveira and Dr. Pinar who both meticulously reviewed my project report and provided constructive feedback to better articulate the findings of the project and the future work. Similarly, I would like to appreciate the support of Dr. Zheng Peng and his PhD student Mr. Dmitrii Dugaev at the City University of New York whose in-depth knowledge of Aqua-Sim-NG simulator played a vital role in assisting with the underlying challenges related to code development of CARP.

Moreover, I would like to thank my friends at Michigan Tech who hail from different backgrounds, cultures and countries. Further to that is the African Students Organization family and the National Society of Black Engineers, Michigan Tech family, both associations have demonstrated the value of collegiality, togetherness and providing succor to loved ones at important times.

## List of Abbreviations

UWSN:	Underwater Wireless Sensor Network
RF:	Radio Frequency
CARP:	Channel-aware Routing Protocol
MAC:	Media Access Control
RTS:	Request to Send
CTS:	Clear to Send
CSMA /CA:	Carrier Sense Multiple Access /Collision Avoidance
FBR:	Focused Beam Routing
PER:	Packet Error Rate
VBF:	Vector Based Forwarding
DBR:	Depth Based Routing
$d_c$ :	Depth of current node
$d_p$ :	Depth of previous node
NS-3:	Network Simulator (Version 3)
AoA:	Angle of Arrival
SP:	Sender Position
FP:	Forwarder Position
TP:	Target Position
HC:	Hop Count
BPSK:	Binary Phase Shift Keying
BER:	Bit Error Rate
SNR:	Signal to Noise Ratio
FAMA:	Floor Acquisition Multiple Access
SFAMA:	Slotted Floor Acquisition Multiple Access

## **Abstract**

Over two-thirds of the earth surface are occupied by water, however the level of research in the field of Underwater Wireless Sensor Networking (UWSN) does not necessarily commensurate with the size and potentials. Importantly, the underwater acoustic channel is susceptible to signal degradation as a result of the dynamic and harsh state of the environment such as high propagation delay and limited channel bandwidth.

The above challenges are the motivation behind the Implementation of The Channel-aware Routing Protocol (CARP) for the routing performance optimization in UWSN. CARP [1] is a cross-layer routing paradigm which collaboratively leverages the link quality estimation between neighboring sensor nodes and the hop counts from the sink to determine the next relay node for packet forwarding. In this work, CARP was implemented in Aqua-Sim-NG, an ns-3 based underwater wireless sensor network simulator that simulates underwater acoustic channels with high fidelity.



# 1. Introduction

## 1.1. OVERVIEW OF UWSN

Underwater wireless sensor networking (UWSN) [2] has been the enabling technology for a broad range of evolving applications such as ocean monitoring for scientific exploration and commercial exploitation, prediction of underwater seismic events, coastline monitoring, offshore exploration etc. A typical underwater environment setup requires multi-hop networks where spatially placed sensor nodes transmit data to one or more sinks located at the surface level which forwards the received information to onshore control stations usually radio frequency (RF) transmission.

According to [1], the major hindrance of deploying UWSN emanates from the specific environment they operate. Underwater communications are characterized by long propagation delay, limited bandwidth, sound speed variability, slow power signal attenuation, environmental impairments, human interference, doppler effect based on the relative movement of sensor nodes etc. The rapidly changing conditions of the acoustic channel in the UW environment may probably give rise to time-varying link reliability and asymmetric links. The former has a significant impact as it poses novel design challenges in the medium access control (MAC) and routing protocols which differs largely from its terrestrial counterpart.

Existing routing protocols for UWSN at the MAC and routing layers address the problems of channel access and multi-hop separately. Accordingly [3], it was demonstrated that cross layer techniques can impact protocol performance positively most especially with networks with limited resources or deployed in a challenging environment. A good example of a cross-layer approach is Focused Beam Routing (FBR) where control packets (RTS/CTS) as in CSMA /CA like channel access are used in channel reservation and carry node location information that is used to select next hop relay, which is the neighbor closest to the sink. The use of short packets for channel access and relay selection particularly effective for routing in challenging UWSN. They can be robust during transmission as a result of the small byte size using the low rates of acoustic modem as they are also less susceptible to noise and interference. The use might not be efficient in selecting a relay required to receive a longer data packet correctly. In solutions similar to FBR, link quality is not considered when it comes to selecting the next hop relay.

In this report, CARP is presented [1], a new distributed cross-layer for multi-hop delivery of data to the sink. The use of short packets for robust channel access and relay selection. CARP addresses the drawbacks of other solutions as link quality is taken into account in the cross-layer relay selection. Relays are selected if they have a history of successful transmissions to their neighbors in routes to the sink. CARP also combines link quality with simple topology information (hop count) for routing around connectivity holes and shadow zones. Among viable relays, preference is given to neighbors with the highest residual energy and can receive the larger number of packets. CARP is also designed to leverage modern power control by selecting transmission powers in such a way that shorter control packets experience similar packet error rate (PER) of longer data packets. Power increase is made for transmission of data packets after the handshake process which is when the relay selection has been completed to increase the delivery likelihood of data packets.

The rest of the report is organized as follows. Section II describes related work which gives an overview of other multi-hop routing protocols. In Section III, Channel-aware routing protocol is described in addition with the algorithm behind the protocol. Results via development of the codebase are presented in Section IV. Future research and conclusion are projected in Section V and Section VI.

## **1.2. CONTRIBUTIONS**

The works and results presented in [4] were very essential in creating a robust foundation which was used while working on this project. The following highlights some of the key contributions of this project in the wireless communications of data packets from the source nodes to the sink in the underwater acoustic channel as found below:

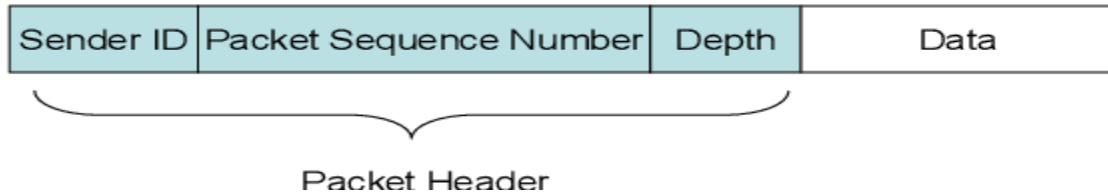
- I. Design, development and integration of the Channel-aware routing protocol in the Aqua-Sim Next Generator simulator
- II. Integration of a cross-layer routing protocol into the existing routing modules of the simulator framework
- III. Provision of a comprehensive codebase documentation on GitHub for researchers in Underwater Wireless Sensor Networking who intend to build upon the existing routing CARP routing protocol

## 2. Related Work

The underwater channel is very harsh and dynamic which depicts the unpredictability of channel status that directly affects the performance of data transmission. Existing routing protocols which are in use leverage limited metric in determining the next-hop, thus the relay node which is used to forward the packets towards the destination from the sink.

A review of **Depth Based Routing (DBR)** [5] which uses relative depth of sensor nodes from the sensor nodes to determine the forwarding path towards the destination. Shortcomings range from node might not be in the direction of the destination, does not factor the channel status to determine which node works best, on-demand routing protocol which contributes to latency and reduced throughput efficiency etc. DBR leverages the general UWSN architecture, based on the depth information of each sensor, it forwards data packets greedily towards the sink. It has a field in the packet header which records the depth of the forwarding sensor node. It is important to note that the depth information is regarded as the vertical distance from itself to the water surface.

To summarize, the major merits of DBR are as follows. 1) It does not require full-dimensional location information. 2) It handles dynamic network with proper energy optimization. 3) It capitalizes multiple-sink network architecture without introducing extra cost



**Figure 1: Header information of DBR [5]**

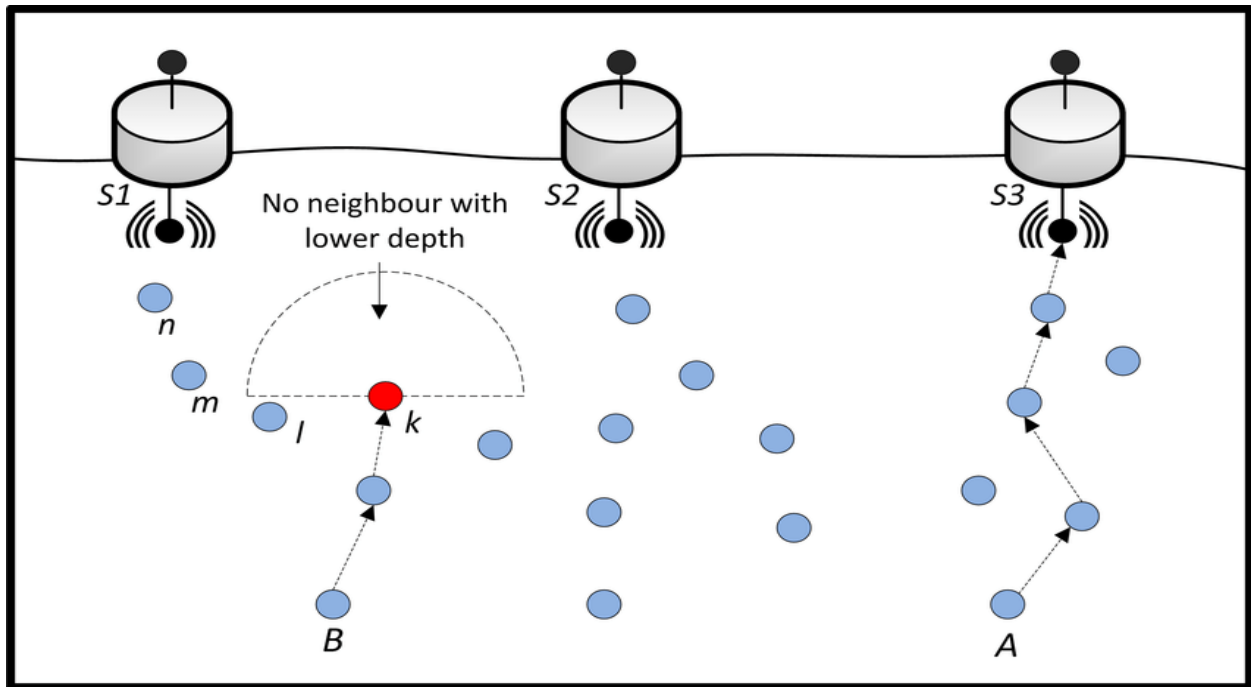
### Protocol Overview and Design

In DBR, as the packet approaches the destination, the depth of the forwarding node decreases. The depth of the forwarder or packet's previous hop is termed **dp**. The depth of the receiving node is termed **dc**, if the node is close to the surface  $dc < dp$ . If a node forwards a packet, as shown in the above header information, the packet contains the sender id, packet sequence number which is unique for every packet and the depth information of the sending node. A node which receives the packet checks if its depth information is less than that of the previous hop, if the former is true, the node is considered as a qualified candidate to forward the packet, otherwise the packet is discarded.

According to [6], DBR being a multiple path routing protocol would likely generate redundant packets during packet transmission, which is why the packet sequence number embedded in the packet header is essential to minimize broadcast and collisions. The use of holding time which is the amount of time delay before a packet is forwarded is used to suppress redundant packet transmissions when multiple nodes are considered as qualified nodes. In DBR, each node adopts a priority Queue Q1 and packet history buffer Q2 to achieve minimization in redundant packet transmission. Every node places a packet destined for forwarding in Q1, after the expiration of the holding time the packet is forwarded. Other qualified nodes with longer distances from the sink have a longer holding time. The logic is simply allowing nodes further away from the sending node to have shorter holding time in the event of having multiple qualified candidates. Once a packet is forwarded, the sender id and packet sequence number is queued in Q2. At the

point when Q2 is filled, the least recently accessed entry is dropped which depicts a node always has the most recent copy of a forwarded packet. This way, nodes participating in DBR avoid sending duplicate packets and prevent packet congestion when multiple qualified nodes are selected.

Depth threshold which is represented as **dth** is another way DBR has been able to reduce the number of nodes participating in packet forwarding. A node only forwards a packet if the value of  $(dp - dc) > dth$ . The smaller the depth threshold the larger number of nodes selected as qualified candidates, vice versa. This method is also a good tradeoff used by DBR in addressing energy efficiency and packet delivery ratio (PDR). A larger value of **dth**, results in a much more energy efficient wireless sensor but impacts adversely on the PDR as fewer nodes participate in packet forwarding. However, a small value of **dth**, allows more nodes to participate which affects the energy utilization of the nodes but improves the PDR.



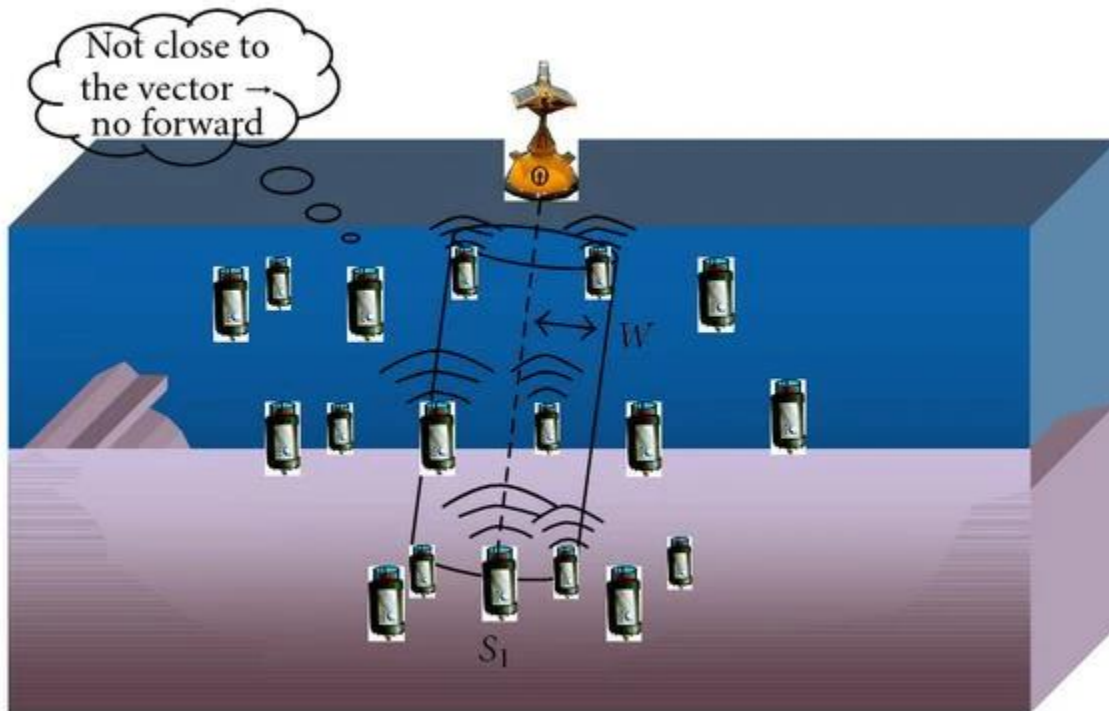
**Figure 2: An illustration of void in Depth Based Routing [7]**

Moreover, an extension of geographic based routing is **Vector Based Forwarding (VBF)** [6] which is scalable, robust and energy efficient. Packet transmission from the source to the destination is determined via a forwarding path which is similar to a vector along the sender and the receiver. In VBF, every packet contains the positions of the sender, forwarder and the target also called the destination. It is assumed that every node participating in VBF is aware of its position information which is provided by some location algorithm. In the event a node is unaware about its location information, the relative distance to the forwarder and angle of arrival (AoA) could be used in determining the position of the node.

If a node receives a packet, it compares its position with the relative position of the forwarder. Also, a certain width threshold is used which is derived from the radius of the routing pipe along the sender and target. If the position of the node exceeds this width value, the packet is discarded, otherwise, the node

appends its position as that of the forwarder and sends the packet. Only nodes within the axis of the pipe formed along the vector take part in packet forwarding which makes VBF robust to energy utilization. It is important to note that no state information is required for the operation of VBF.

According to [5], In a packet, there are three position fields, SP, TP, and FP, that is, the coordinates of the sender, the target, and the forwarder. In order to handle node mobility, each packet contains a RANGE field. When a packet reaches the area specified by its TP, this packet is flooded in an area controlled by the RANGE field. The forwarding path is specified by the routing vector from the sender to the target. Each packet also has a RADIUS field, which is a predefined threshold used by sensor nodes to determine if they are close enough to the routing vector and eligible for packet forwarding.



**Figure 3: Diagrammatic representation of VBF in Underwater Sensor Networks [8]**

### 3. Overview of the CARP Protocol

Channel-aware routing protocol (CARP) [1] is a cross-layer routing paradigm that leverages both the MAC and the routing layer to determine the relay node used for packet forwarding. The use of control packets is initially exchanged during the handshake process for channel reservation and selection of relay nodes before packet forwarding is initiated. The link quality of the sender to all the neighboring nodes are computed which have a positive advancement towards the destination. Upon responding with the link quality which is highly dependent on nodes that have exhibited a history of successful transmission, the node with the highest value is selected as the forwarding node. It is important to note that CARP is an on-demand and reactive routing protocol as this process is always initiated anytime a node decides to forward a packet. Moreover, it does not require management of state information as every attribute required during the route discovery process is only at the point of need.

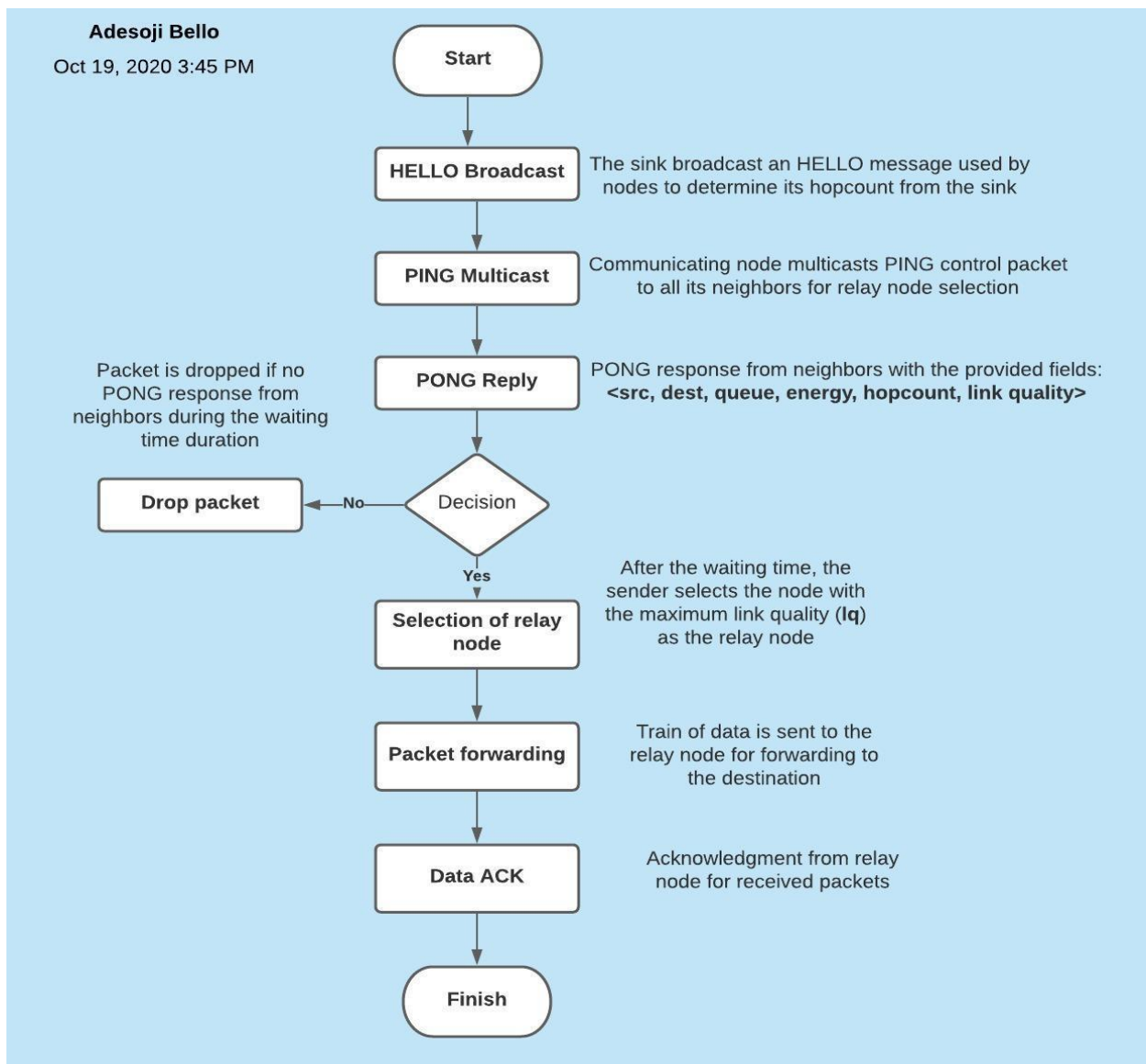


Figure 4: Flowchart of the CARP algorithm [10]

### 3.1. Description of the CARP Protocol

The following enumerates the routing procedures using CARP below:

- I. Broadcast of HELLO packet by the sink node
- II. Exchange of PING packet by the sending node during the relay selection process
- III. Reply from neighboring nodes with a PONG packet during the waiting time  $\delta$
- IV. Selecting of relay node based on the results of the PONG packet
- V. Forwarding of packets by the sender node
- VI. Data acknowledgement from the receiving node
- VII. Update of hop count information by the sender to dynamically be in sync with the unpredictable nature of the underwater environment

### 3.2. Algorithm of the Protocol

At network setup [1], HELLO packets are flooded from the sink through the network. All nodes acquire hop count which is the spatial distance from the sink. In line with the above-mentioned procedures, the algorithm behind each process would be further explained in subsequent paragraphs.

#### A. Broadcast of Hello Packet

```
<i>. Sink broadcast HELLO packet (p) to all its neighbors
<ii>. Each packet contains <src, numPkt>. Since it is emanating from the
sink HC(X) is 0
<iii>. At every node the following occurs:
If  $HC^t(x) < HC^{t-1}(x)$ 
    Hop =  $HC^t(x) + 1$ 
    forward (p)
else
    Hop =  $HC^{t-1}(x)$ 
    forward (p)
```

#### B. Relay Selection

During the relay selection process, nodes exchange **PING** and **PONG** packets which possess fields used in determining which of the neighboring nodes become the next hop used in advancing the packet towards the destination.

A source node x sends a **PING** packet which carries the following information:

```
<src, numpkt>
```

A neighboring node y responds to x with a **PONG** packet with the following fields:

<src, dst, hop, queue, energy, lq>

The **src** field represents the sender ID of the source node while the **hop** field represents the number of hops of the node from the sink, the **queue** represents how many packets the neighboring node can accommodate, the **energy** shows the residual energy of the node and **lq** is the link quality of the neighboring node to its best neighbors towards the sink. Node  $x$  awaits PONG replies from neighboring nodes for a time  $\delta$ . The waiting time is dependent on the modem nominal transmission range and acoustic signal propagation speed in water which is continuously updated by using the actual round-trip time of the PING and PONG packet exchange. After the time  $\delta$ , the link quality  $lq_y$  of all the available neighbors  $y$  of node  $x$  is used to determine the goodness of  $x$  to all neighbors of  $y$ . For each neighbor  $y$  of  $x$ , node  $x$  computes the goodness in [1] as found below:

$$\text{goodness}_y = lq_{y,z} lq_{x,y} \quad (1)$$

The node  $y$  with the highest ratio  $\text{goodness}_y / \text{HC}(y)$  is chosen as the relay node and a train of packets is sent to it. After receiving a train of data packets, node  $y$  replies with a cumulative ACK, acknowledging each single packet in the train (bit mask). Upon receiving acknowledgement from node  $y$ , node  $x$  updates its hop count  $\text{HC}(y) + 1$ , in this way the hop count information is dynamically updated based on the changes in the network. It is important to note that node  $y$  only re-transmit packets not previously forwarded so as to reduce congestion on the network and mitigate duplicate packet transmission.

### C. Link Quality Computation

The goodness factor is an estimate of the channel quality from node  $x$  to  $y$  and  $y$  to its best reachable neighbor in a route towards the sink. More precisely for each data transmissions to one of its neighbors  $z$ , node  $y$  computes the link quality in [1] as found below:

$$lq_{y,z}^t = \alpha Y_{y,z}^t + (1 - \alpha) lq_{y,z}^{t-1} \quad (2)$$

$lq_{y,z}$  is computed based on the success of past transmissions to its neighbors. The use of  $\alpha \in (0,1)$  is a smoothing factor which controls how fast the impact of older transmissions decreases. Depending on how well previous transmissions plan to be discounted, that also determines the value  $\alpha$  would be assigned. The value  $Y_{y,z}^t$  is the packet success ratio (the number of packets correctly received by  $z$ , hence, acknowledged by  $z$  to the number of packets forwarded in the train of that transmission from  $y$ ) from node  $y$  to all of its neighbors at time  $t$ , while the value  $lq_{y,z}^t$  is the value of the moving average after  $(t-1)$  transmissions.

### D. CARP and Power Control

Generally, cross-layer protocols as a result of the handshake mechanism for joint channel access and relay selection such as CARP determine the next hop relay based on the correct exchange of control packets. A channel is reserved once a neighbor has been selected as a relay and subsequently used for data transmission. A selected packet error rate (PER) for short control packets might be too high for data packets which are longer.



It is important to note that common wireless sensor modems are not flexible to select a given transmission power, however, CARP is designed to leverage on the power control to obtain similar PER for both data and control packets. The power used to send PING is computed to obtain a PER corresponding to a given channel BER, once a relay has been selected, the power is increased so the corresponding PER is similar to that of PING /PONG exchange through which the relay has been determined. Using BPSK modulation technique, the probability to transmit a packet that is  $L$  bits long is  $(1-BER)^L$ . Specifically, the power  $P$  for transmitting packets at a given PER is computed in [1] as found below:

$$BER = \frac{1}{2} \operatorname{erfc}(\sqrt{SNR}) \quad (3)$$

Where:

BER: Bit error rate

Erfc: Complementary error function

$$SNR = \frac{P / A(r,f)}{N(f) \Delta f} \quad (4)$$

Where  $P$  is the transmission power,  $A(r,f)$  is the attenuation in the underwater channel over a distance  $r$  for a signal frequency  $f$ ,  $N(f)$  is the noise spectral density and  $\Delta f$  is the receiver noise bandwidth.

### 3.3. Description of Aqua-Sim

According to [9], Aqua-Sim is an ns-3 based underwater sensor network simulator. To better advance the research of UWSN, the need to create a robust standard simulation platform to benchmark and evaluate various network designs, algorithms and protocols. Most of the existing network simulators were designed for terrestrial radio wireless or wired networks and not for UWSN which made it difficult to incorporate acoustic propagation models into the simulation of underwater acoustic networks. Aqua-Sim was previously developed on ns-2, but as a result of numerous routine releases and enhancements on ns-2, this led to the evolution of ns-3 in 2008. In order to address various constraints introduced by ns-2 which was the underlay framework for Aqua-Sim, **Aqua-Sim Next Generation** was introduced which is a network simulator based on ns-3. The objective of the revamp was to improve memory management via the use of smart pointers, adopt proper packet handling techniques and overall performance improvements.

Taking cognizance of the unfriendly nature of the UWSN, it is very essential to simulate various network topologies akin to the real-world network design to obtain important testing results required for research purposes. Consequently, the areas of improvements of Aqua-Sim NG are enumerated below.

- I. In [9], there was an improvement in the channel support which consist of specialized noise generators, multiple channel support, range-based propagation, and trace driven testing. This ensured a localized packet interference as well as variation in the range of the acoustic propagation between modems which are spatially placed in the UW environment
- II. The physical model support was also enhanced via integration of through of a signal cache, signal-to-interference-plus-noise ratio (SINR) checker and modulation support. This enabled decoding of packet on the physical layer of the modem depending on the existing UW channel condition

- III. Provision of protocol layer support which include synchronization and localization modules, security features and busy terminal model support. As a result of the high mobility and considerable transmission delays experienced in underwater, it is important to be capable of testing specialized security techniques
- IV. Integration of adapted Information-Centric Networking (ICN) techniques for UWSN. Considering the inventive approach of ICN architectures, it is tremendously prospecting to develop and integrate various modules of the framework to enable testing in the long-term

### Reasons why UWSN is not practically suitable for the current network simulators?

- I. Existing network simulators were designed for the terrestrial environment using radio signals for packet transmission in lieu of acoustic signal
- II. The speed of propagation of sound is 1500m/s compared to that of the radio signal. In a similar vein, the acoustic signal propagation model in the underwater channel is quite different from that of the radio channel propagation model in free space
- III. The simulator suited for UWSN is usually designed for a 3D space while the existing network simulator for the terrestrial environment is for a 2D space

### 3.4. Architecture and Implementation of Aqua-Sim NG

The existing application programming interfaces (API) of both ns-2 and ns-3 are significantly different, in this regard, Aqua-Sim NG was completely rebuilt to accommodate these new standards. In most cases, it involved rewriting new functions and classes to adhere to the ns-3 standard.

Aqua-Sim NG's new architecture in Figure 5 is centered around the net device of each individual node which facilitates interaction with the channel layer. Isolated modules could be used in parallel such as adopting synchronization support with the MAC protocols to better manage the way packets are handled during simulation.

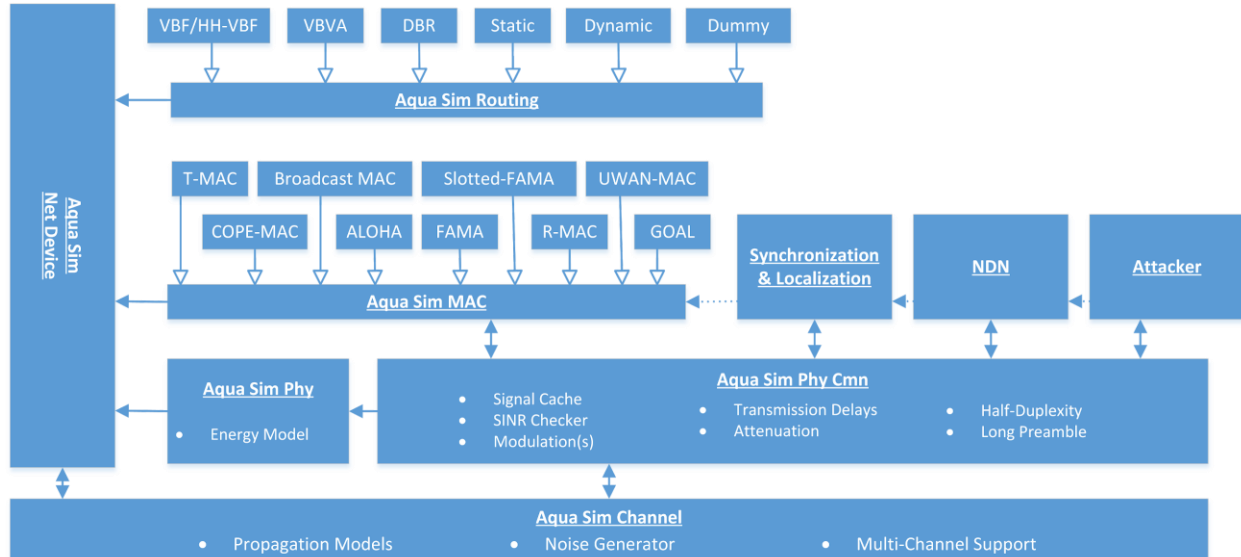


Figure 5: Architecture of Aqua-Sim Next Generation [9]

Furthermore, the inheritance of classes among the modules are well related to ensure viable use of the functions belonging to the class of each module.

### **Advantages of Aqua-Sim NG**

- I. It is a discrete-event trace driven network simulator
- II. It supports a complete protocol stack from the physical to the application layer
- III. It supports 3D networks and mobile networks
- IV. It simulates underwater acoustic channel with a significant level of accuracy

### **Implemented Protocols in Aqua-Sim NG**

1. Media Access Control (MAC) Layer
  - a. ALOHA
  - b. Broadcast MAC
  - c. GOAL: a geographical routing MAC protocol which combines VBF and handshake scheme
  - d. Slotted-FAMA: Floor Acquisition Multiple Access protocol which combines carrier sensing with handshake mechanism
2. Routing Layer
  - a. VBF: vector-based forwarding
  - b. HH-VBF: hop-by-hop VBF
  - c. Dynamic routing: distance vector which uses the hop count information
  - d. Static Routing: creates predefined routes on the routing table
  - e. Dummy Routing: suitable for testing purposes which only forwards packets to the upper /lower layers

## 4. Results

### 4.1. Overview of Results

After careful study of the underlying implementation concepts of the Aqua-Sim-NG module with special emphasis on how the application layer, routing layer, mac layer, physical layer and the underwater acoustic channel interact with the AquaSimNetDevice which is a class abstraction represented by an object that holds a unique address of every node participating in the simulation and require a way to allow the upper layers interact with the lower layers, the aqua-sim-routing-carp module was finally developed, compiled and linked with the existing modules of the Aqua-Sim-NG simulator.

According to [10], the Channel-aware routing protocol is made up of the following files as enumerated below:

1. aqua-sim-header-routing.h
2. aqua-sim-header-routing.cc
3. aqua-sim-routing-carp.h
4. aqua-sim-routing-carp.cc

#### 4.1.1 Aqua-Sim Routing Header

This is contained in the **aqua-sim-header-routing.h** file with header files of pre-existing routing protocols in Aqua-Sim-NG which include: VBF, DBR, Dynamic and Dummy routing. All header files contain declaration of methods of the specific class attached to the routing protocol in question. In order to simplify and ensure previously declared function /methods are used by other header files, the class of all the routing protocols is a derived class of the base **public Header** class which contains virtual functions for serialization of frames, deserialization of bits in the wire, obtaining the size of serialized bits and extracting the instance type of the packet header.

As regards the CARP module, a class tagged **CarpHeader** was declared as a derived class from the public Header base class. This class contain the methods used to set and retrieve the source address of the sender and destination, set and obtain the hop count of a node from the sink, set the values of the queue capacity of the node's buffer, keep track of individual's node residual energy, and also append packet tags to packet to distinguish between DATA, ACK and packets used for link quality estimation. Moreover, the **HelloHeader** which is used during neighbor discovery is also a derived class of the CarpHeader leveraging already declared methods in the base class. Similar class declaration logic is also adopted by the **PingHeader** and the **PongHeader** used to obtain the link quality estimate and selection of the relay node.

#### 4.1.2. Aqua-Sim Routing Header (Main Code File)

This is the main file of the aqua-sim-header-routing contained in the **aqua-sim-header-routing.cc** is a mirror of the header file in terms of the number of functions available in the file. The only difference is that functions are defined in this case, by so doing, the logic behind the operation of all the methods required to ensure the CARP headers perform as desired is implemented.

### 4.1.3 Aqua-Sim Carp Header

This is the proprietary header file of the CARP protocol in the **aqua-sim-routing-carp.h** file which contains the declaration of all the states of the protocol from neighbor discovery to packet forwarding. Similarly, the **AquaSimCarp** is a derived class of the **AquaSimRouting** the same way the CARP headers were derived classes of the Header class to mitigate redundancy in function definition and utilizing the polymorphic attribute of classes in C++. In this file, a constructor and destructor of the class was declared so as to initialize all class variables to null. A map container was used to create a logic between the address of every node and its neighbors. A typeid library was used to keep track of the class attributes during runtime, methods for the HELLO broadcast for both the sending and receiving node were declared, same way for the PING control packet and the PONG packet. Also, a set of functions were declared to enable the neighbors to send an acknowledgement for received PONG packet used for link quality estimation coupled with a method to determine the relay node of the sender using the provided smoothing factor and an ACK count from its respective neighbors.

Moreover, the base AquaSimRouting class has three inherent functions to enable the routing layer to interact with the lower and upper layer of the protocol stack. The send down method was declared to help transmit the packet to the mac layer for onward transmission to the channel, the send up layer is being used by the routing layer to send DATA packets specifically meant for the current node to the application layer while the receive method is being used to interface with both the mac and application layer depending on the destination address encapsulated in the packet.

### 4.1.4 Aqua-Sim Carp Main

This file is contained in the **aqua-sim-routing-carp.cc** file. Virtually all the methods and variables declared in the header file of the CARP routing module were utilized in the main file coupled with the aqua-sim routing header files which is jointly shared among other routing protocols in the simulator. In consistent with the CARP protocol description in the above paragraph, the methods of the routing module followed the same concept with a much more granular level of implementation. The main protocol states in this file are the HELLO broadcast, PING multicast, PONG unicast and DATA forwarding.

#### HELLO Broadcast

The SendHello and RecvHello methods of the class handle this process. At the initialization phase of the CARP algorithm, the sink uses the SendHello to broadcast HELLO packets to all its neighbors, the RecvHello module is used by the neighbors to receive the broadcast and update its hop count information from the sink before invoking the SendHello module for subsequent broadcast of HELLO packets. This process is repeated until the time duration for the neighbor discovery elapses. A map container is used by the receiving neighbor node to store the neighbor address information of every unique neighbor that sends a HELLO broadcast message to it. This is used by the PING module to send PING packets. As previously mentioned, the objective of this method is to ensure every node in the network topology has its hop count information from the sink and also an array of its neighbors.

#### PING Multicast

The SendPing and RecvPing methods are used to send a PING packet to all the neighbors of the sender node who intends to forward a packet. From the neighbor information obtained during the HELLO broadcast phase, a sender calls the SendPing function to multicast PING packets to all its neighbors, every node receiving a PING packet uses the RecvPing to handle this process. The RecvPing has no major

implementation but only calls the SendPong module. This set of methods are used at the initial stage for the link quality estimate of the sender to all potential relay nodes.

### **PONG Unicast**

As rightly mentioned in chapter 3 of this report specifically while discussing the PONG packet, the sender node waits for a time duration which is determined by the speed of sound in water and the euclidean distance between the two farthest nodes for PONG packets. A neighbor uses the SendPong method to send the PONG packet which has in its header the source address, destination address, residual energy, buffer, hop count and the link quality estimate of its best neighbor towards the destination. As a result of the complexity of the latter, only the link quality estimate from the sender to all the neighboring nodes are implemented in this scope. The sender uses the RecvPong to receive the PONG packets from its neighbors during the wait time and also invoke the SetNextHop which returns the address of the neighbor node with the highest link quality estimate using a given smoothing factor which discounts past transmissions at the current time of transmission. It is important to note that the PONG methods are used to select the next relay node based on the node with the maximum packet to success ratio based on correctly delivered packets at the time of transmission

### **DATA Forwarding**

Using an already existing Recv method inherited from the AquaSimRouting class, once a node selects a relay node, the packet is encapsulated with the required information and sent to the mac layer using the Senddown function. In the event whereby a node receives a packet, it cross checks if the packet is a DATA packet and verifies if the destination address is the same as its own address. If the former is valid, the Sendup method is used to send a packet to the application layer, otherwise the packet is encapsulated and destined for forwarding to the mac layer. The use of a boolean return type is used by this method to terminate the process after forwarding down and moving the packet to the upper layer to prevent a loop in the implementation.

After an extensive code development of the above-mentioned files and painstakingly debugging the codebase, Figures 5 and 6 show a successful build of the CARP protocol in Aqua-Sim-NG simulator and the available modules co-existing with the default modules on ns-3 respectively.

```
sojiadvanced@labdevice:/home/Store/Desktop/ns-allinone-3.27/ns-3.27$ ./waf
Waf: Entering directory '/home/Store/Desktop/ns-allinone-3.27/ns-3.27/build'
[ 59/2718] Compiling install-ns3-header: ns3/aqua-sim-routing-carp.h
[1643/2718] Compiling src/aqua-sim-ng/examples/floodMac.cc
[1644/2718] Compiling src/aqua-sim-ng/examples/ddos.cc
[1645/2718] Compiling src/aqua-sim-ng/examples/GOAL_string.cc
[1646/2718] Compiling src/aqua-sim-ng/examples/broadcastMAC_example.cc
[1740/2718] Compiling src/aqua-sim-ng/model/aqua-sim-routing-vbva.cc
[1741/2718] Compiling src/aqua-sim-ng/model/aqua-sim-routing-carp.cc
[1754/2718] Compiling src/aqua-sim-ng/examples/floodtest.cc
[1755/2718] Compiling src/aqua-sim-ng/examples/bMAC.cc
[2361/2718] Compiling src/aqua-sim-ng/examples/named_data_example.cc
[2376/2718] Compiling src/aqua-sim-ng/examples/VBF.cc
[2410/2718] Compiling src/aqua-sim-ng/examples/onandoffapp.cc
[2415/2718] Compiling src/aqua-sim-ng/examples/onandoffapp_dbr.cc
[2579/2718] Linking build/libns3.27-aqua-sim-ng-debug.so
[2651/2718] Linking build/utls/ns3.27-print-introspected-doxygen-debug
[2652/2718] Linking build/scratch/myfirst
[2653/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-named_data_example-debug
[2654/2718] Linking build/scratch/scratch-simulator
[2657/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-broadcastMAC_example-debug
[2656/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-GOAL_string-debug
[2658/2718] Linking build/libns3.27-aqua-sim-ng-test-debug.so
[2658/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-FloodingMac-debug
[2700/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-VBF-debug
[2701/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-bMAC-debug
[2702/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-ddos-debug
[2703/2718] Linking build/scratch/subdir/subdir
[2704/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-floodtest-debug
[2705/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-onandoffapp-debug
[2706/2718] Linking build/src/aqua-sim-ng/examples/ns3.27-onandoffapp_dbr-debug
[2707/2718] Linking build/utls/ns3.27-test-runner-debug
Waf: Leaving directory '/home/Store/Desktop/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (33.122s)

Modules built:
antenna                aadv                applications
```

Figure 6: A screenshot showing build of the CARP module in Aqua-Sim-NG simulator [10]

```
Build commands will be stored in build/compile_commands.json
'build' finished successfully (33.122s)

Modules built:
antenna                aadv                applications
aqua-sim-ng            bridge             buildings
config-store           core               csma
csma-layout            dsdv              dsr
energy                 fd-net-device      flow-monitor
internet               internet-apps      lr-wpan
lte                    mesh              mobility
mpi                    netanim (no Python) network
nix-vector-routing     olsr              point-to-point
point-to-point-layout  propagation        sixlowpan
spectrum               stats              tap-bridge
test (no Python)       topology-read      traffic-control
uan                    virtual-net-device wave
wifi                   wimax

Modules not built (see ns-3 tutorial for explanation):
brite                  click              openflow
visualizer

sojiadvanced@labdevice:/home/Store/Desktop/ns-allinone-3.27/ns-3.27$
```

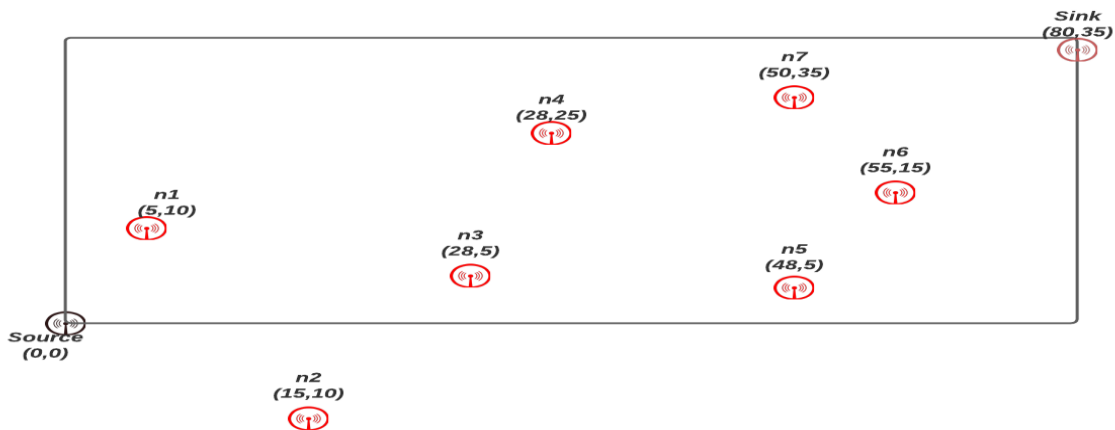
Figure 7: A picture of the successful built modules under the Aqua-Sim-NG in NS-3 [10]

Importantly, to ensure researchers who intend to build upon the existing results of the CARP framework already developed in Aqua-Sim-NG, in [10], a comprehensive ReadMe documentation is provided which thoroughly explains the use of class dependencies, arguments and return parameters of methods declared in the CARP class. This would go a long way to ensure ease with reusability of the codebase which is in line with standard coding practice and best standards for software development.

**Table 4.1. Simulation parameters**

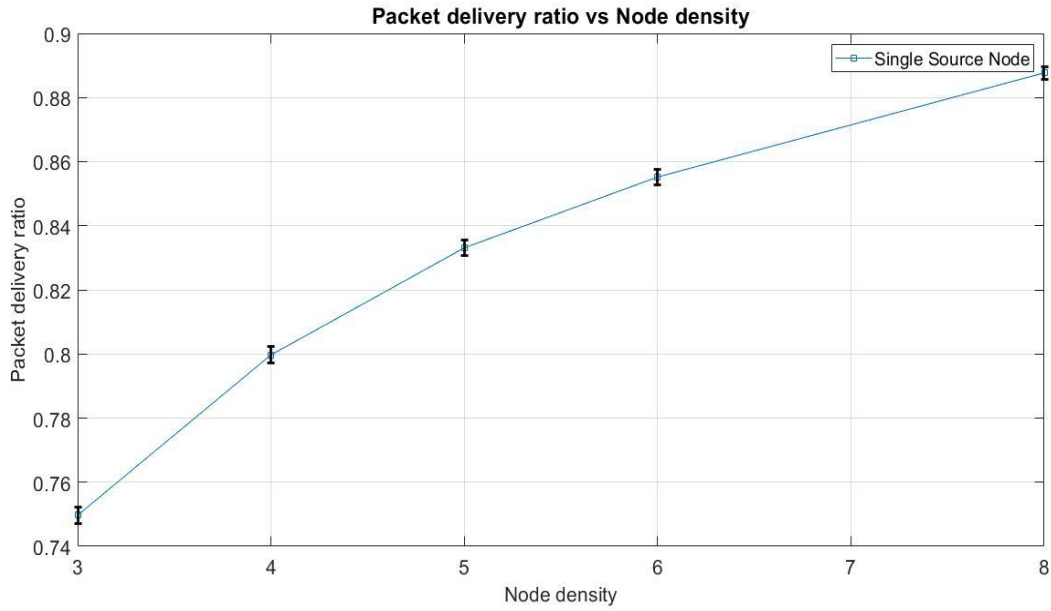
Packet Size	100 bytes
Data Rate	16 kbps
Frequency	24 kHz
Bandwidth	4 kHz
Node Density	3,4,5,6,8 nodes /grid size
No of Sinks	1
Simulation Time	100 seconds
MAC Protocol(s)	Slotted FAMA, ALOHA
Number of Iterations	10

To further evaluate the performance of the CARP protocol, a network topology of distributed sensor nodes ranging from three to eight with a single sink was used to obtain the packet delivery ratio and throughput for a multi-source node setup and a single-source node setup as demonstrated in the topology below. In a similar vein, results obtained from the simulations were polled after running ten iterations and averaging the results.

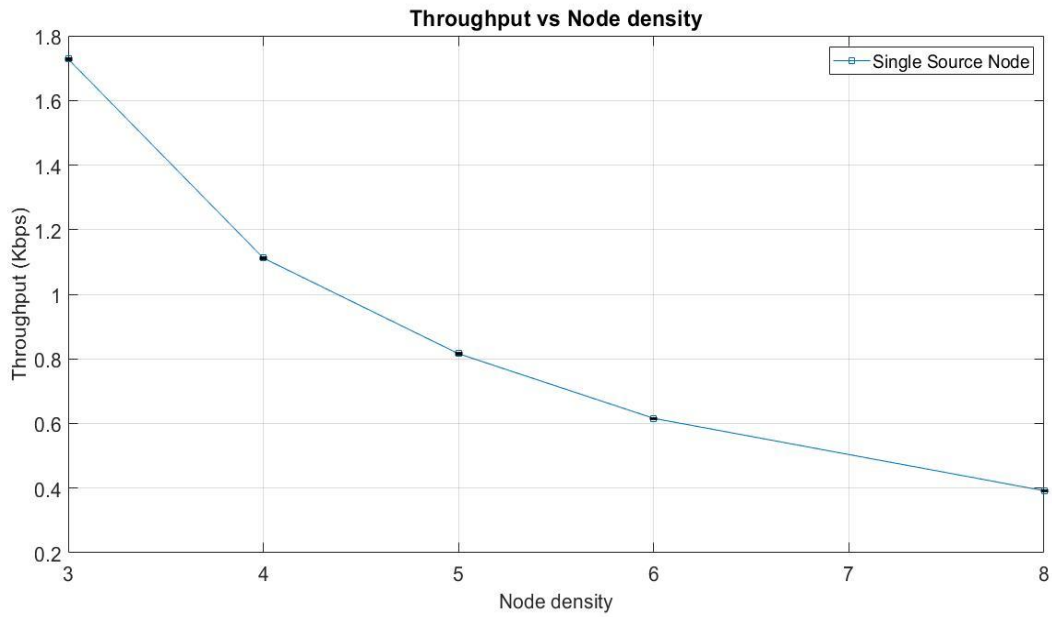


**Figure 8: Network Topology (Single Source Node) [10]**

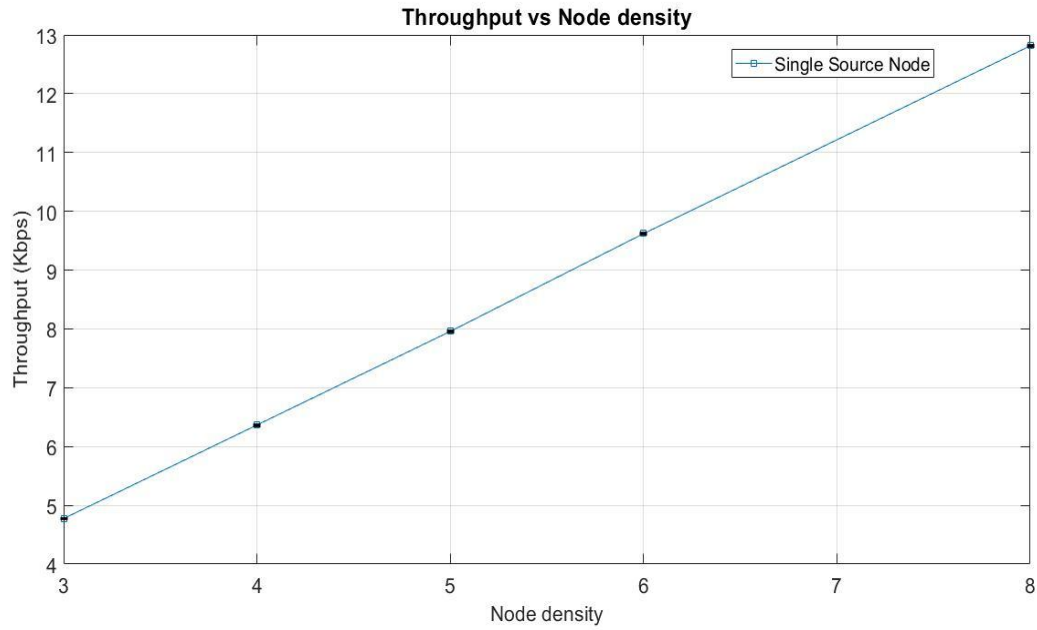




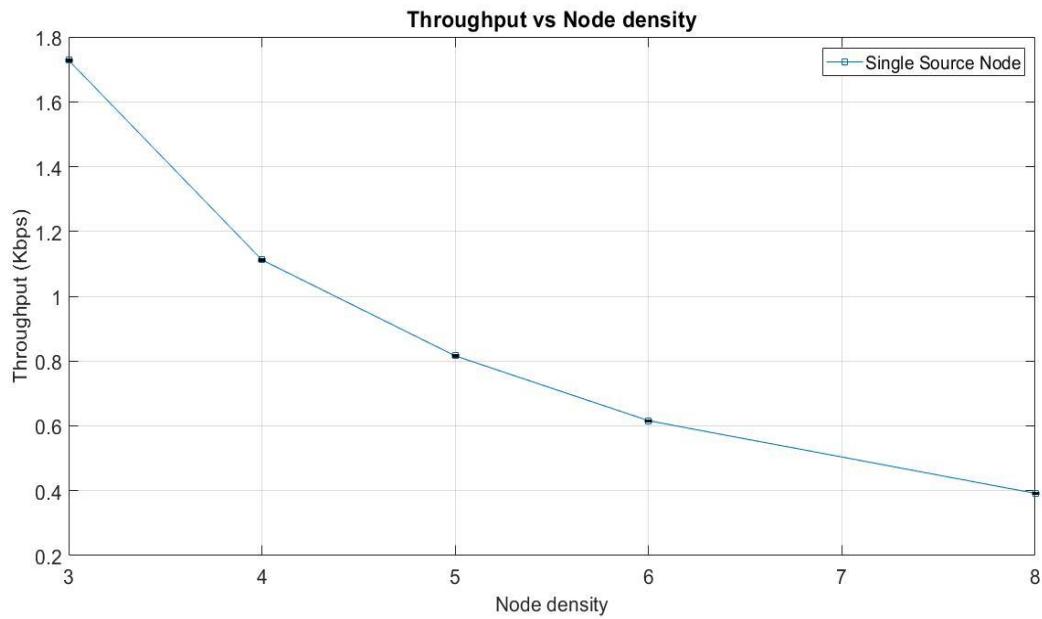
**Figure 9: Packet delivery ratio with varying node density using SFAMA MAC protocol [10]**



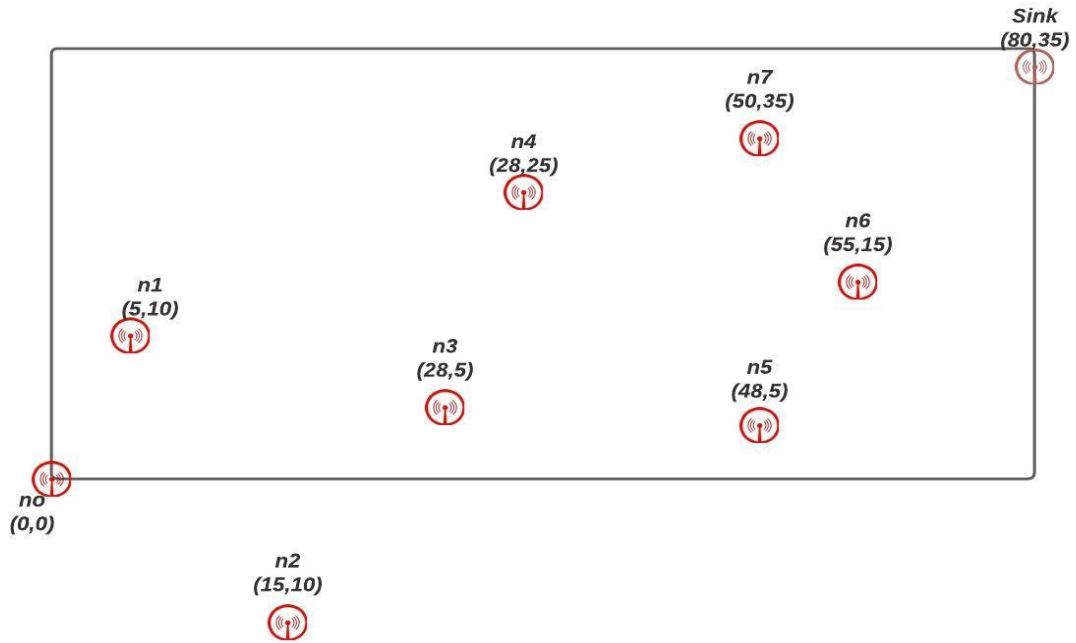
**Figure 10: Packet delivery ratio with varying node density using ALOHA MAC protocol [10]**



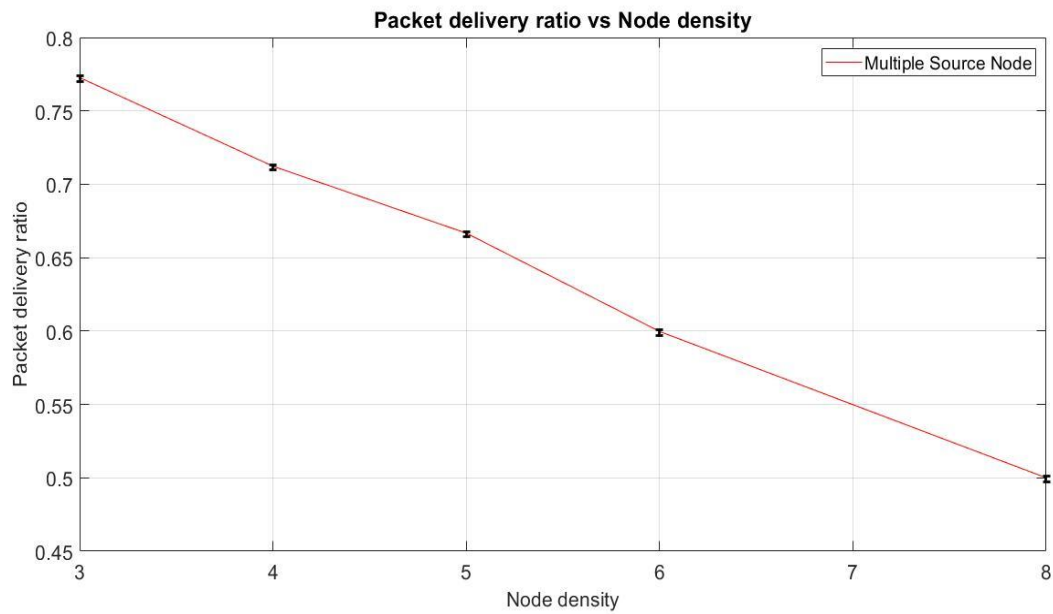
**Figure 11: Throughput with varying node density using SFAMA MAC protocol [10]**



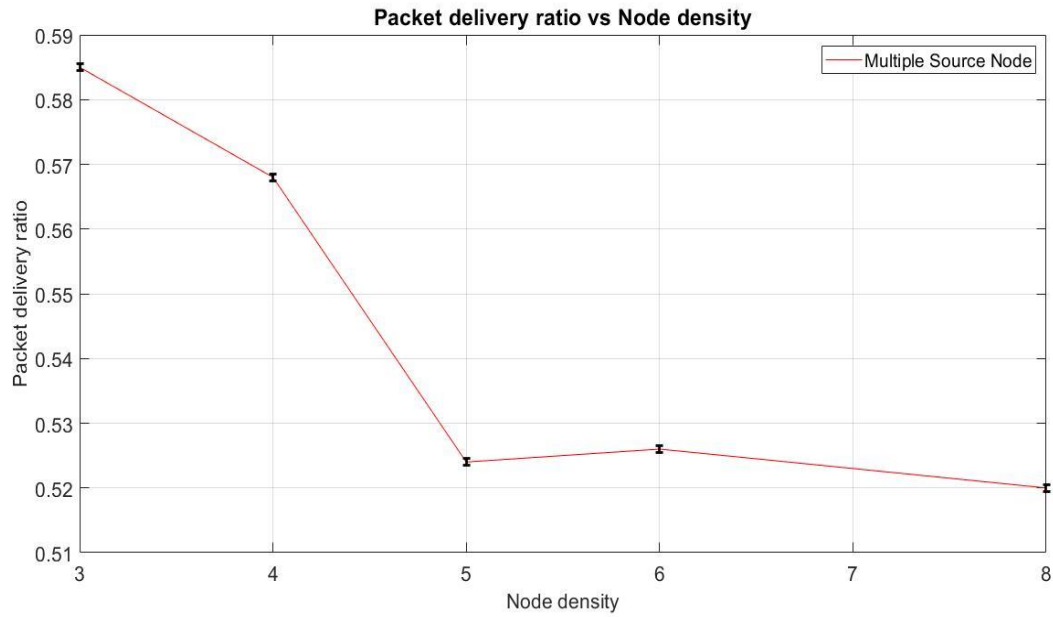
**Figure 12: Throughput with varying node density using ALOHA MAC protocol [10]**



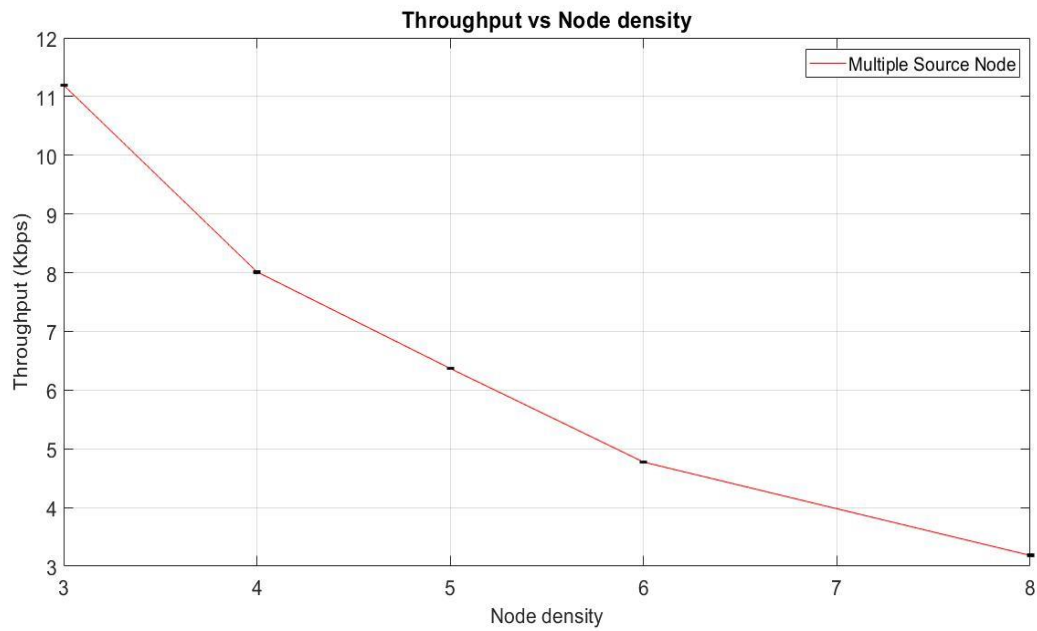
**Figure 13: Network Topology (Multiple Source Node) [10]**



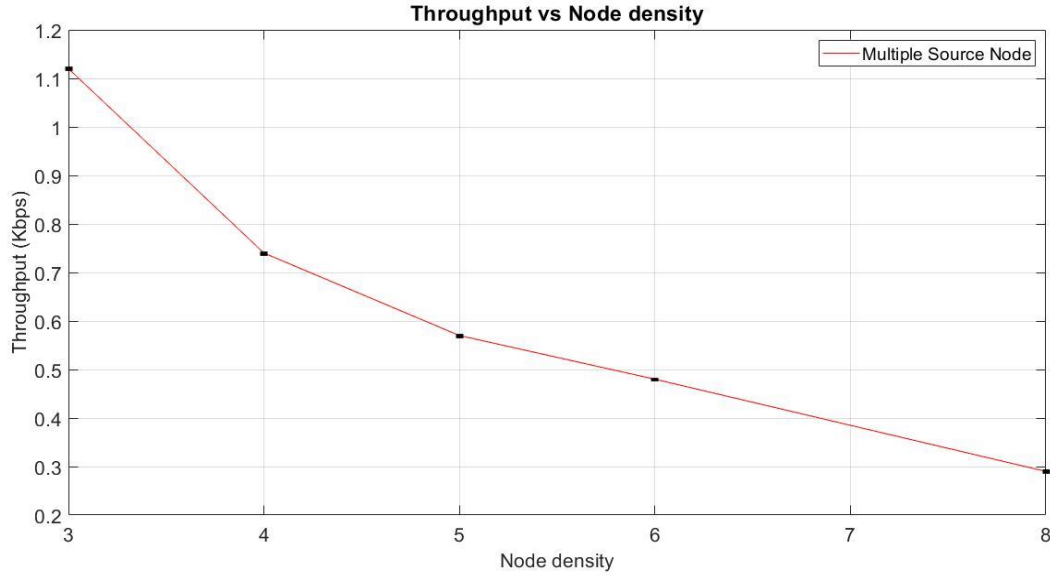
**Figure 14: Packet delivery ratio with varying node density using SFAMA MAC protocol [10]**



**Figure 15: Packet delivery ratio with varying node density using ALOHA MAC protocol [10]**



**Figure 16: Throughput with varying node density using SFAMA MAC protocol [10]**



**Figure 17: Throughput with varying node density using ALOHA MAC protocol [10]**

#### 4.2. Discussion of Simulation Results

- I. The use of Slotted Floor Acquisition Multiple Access protocol (SFAMA) which is a MAC protocol that leverages handshake mechanism for channel reservation further reduced the packet collisions generated during the simulations
- II. According to Figure 9, in the single source network topology using the SFAMA MAC protocol, increasing the node density further bridged the path between the source and the destination and increased the packet delivery ratio. Conversely, Figure 10 shows a downward trend in the packet delivery ratio with the use of the ALOHA MAC protocol. This is due to the frequent packet collisions resulting from poor allocation of the shared media and the likelihood of several nodes transmitting packets simultaneously.
- III. In a similar vein, more bytes were delivered at the destination (sink) using SFAMA MAC protocol which had a considerable impact on the throughput according to Figure 11. However, Figure 12 shows gradual reduction in the throughput with fewer bytes received at the destination using the ALOHA MAC protocol.
- IV. The standard deviation of the values obtained after ten iterations while obtaining the packet delivery ratio for each node density value was very minimal
- V. In the multi-source node network topology, rapid transmissions from the source nodes increased the packet collisions in the network by adversely affecting the packet delivery ratio as experienced in Figure 14 and Figure 15. However, the adoption of SFAMA MAC protocol in Figure 14 was able to cushion the huge collisions that would have been experienced while increasing the node density with the use of its handshake mechanism among nodes to better reserve the channel and also employing time slots. This is in contrast with the use of ALOHA in Figure 15
- VI. The throughput of the network was also severed in the multi-source network topology owing to the reduced successfully delivered packets to the sink as shown in Figure 16 and Figure 17. The use of SFAMA in Figure 16 yielded better throughput values in contrast to ALOHA in Figure 17.

## 5. Future Research

Based on the work that has been carried out and the results achieved, a number of assumptions were made during the development of CARP in Aqua-Sim-NG which introduced some constraints in the computation of the goodness factor via the link quality estimation. Further studies could be conducted through extension of the CARP module to develop a model that better captures the variation of the simple topology information and further computes the  $lq$  values of the sender's neighbors. These are elucidated below.

- I. Gradual variation of the smoothing factor used during the link quality estimation to accurately predict the value that closely relates with the dynamic nature of the underwater environment
- II. Extension of the link quality estimation to further obtain the favorable value among the  $lq_{y,z}$  of its neighbors  $z$  with positive advancement towards the destination
- III. Performance evaluation of the CARP routing protocol benchmarked with existing routing protocols in Aqua-Sim NG. E.g VBF, DBR

## 6. Conclusions

In this project, the implementation of Channel-aware Routing Protocol (CARP) was achieved after careful study of the Aqua-Sim-NG which is an ns-3 based network simulator which is famous for an accurate model of attenuation and packet collisions in the underwater channel according to [9]. The importance of this milestone cannot be overemphasized as it not only extends the available routing protocols in Aqua-Sim-NG simulator but also provides the first cross layer routing protocol in an ns-3 based simulator which takes cognizant of the link quality between the sender and its neighbor nodes and also some attributes in the routing layer.

Moreover, the CARP protocol which was built and linked successfully with existing modules of the simulator provides a baseline for researchers to build upon the cross-layer design by varying key parameters which are sensitive to the link layer for gradual optimization of the routing performance in the highly dynamic underwater environment. Similarly, the simulations conducted with the use of CARP by varying the node density in a single and multi-source network topology further demonstrated promising results of the packet delivery ratio and the throughput. Future studies could also explore the performance evaluation of CARP benchmarked with existing routing protocols in Aqua-Sim-NG. The throughput efficiency, end-to-end latency, packet success ratio and other essential performance metrics could be used to evaluate the network performance across various network topologies of the nodes, be it static or the incorporation of a mobility model.

Lastly, a comprehensive documentation was provided in the code repository [10] for researchers who are actively contributing to the development of high-performance, low cost, effective and robust routing protocols in underwater wireless sensor networking.

## 7. References

- [1] S. Basagni, C. Petrioli, R. Petrocchia and D. Spaccini, "Channel-aware routing for underwater wireless networks," 2012 Oceans - Yeosu, Yeosu, 2012, pp. 1-9, doi: 10.1109/OCEANS-Yeosu.2012.6263538.
- [2] Zhou, Shengli, and Zhaohui Wang. *OFDM for underwater acoustic communications*. John Wiley & Sons, 2014.
- [3] Akyildiz, Ian F., Dario Pompili, and Tommaso Melodia. "Underwater acoustic sensor networks: research challenges." *Ad hoc networks* 3, no. 3 (2005): 257-279.
- [4] Chitre, Mandar, Shiraz Shahabudeen, and Milica Stojanovic. "Underwater acoustic communications and networking: Recent advances and future challenges." *Marine technology society journal* 42, no. 1 (2008): 103-116.
- [5] Xie, P., Zhou, Z., Nicolaou, N. *et al.* Efficient Vector-Based Forwarding for Underwater Sensor Networks. *J Wireless Com Network* 2010, 195910 (2010). <https://doi.org/10.1155/2010/195910>.
- [6] Hai Yan, Zhijie Jerry Shi, and Jun-Hong Cui, "Depth-Based Routing for Underwater Sensor Networks," Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269-2155.
- [7] Ghoreyshi, Seyed Mohammad & Shahrabi, Alireza & Boutaleb, Tuleen. (2017). Void-Handling Techniques for Routing Protocols in Underwater Sensor Networks: Survey and Challenges. *IEEE Communications Surveys & Tutorials*. PP. 1-1. 10.1109/COMST.2017.2657881.
- [8] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J. Cui, Z. Shi, Y. Fei, S. Zhou, "Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks", Underwater Sensor Networks Lab University of Connecticut, Storrs, CT 06269-2155.
- [9] [http://134.74.112.6/mediawiki/index.php/Aqua-Sim\\_NG](http://134.74.112.6/mediawiki/index.php/Aqua-Sim_NG)
- [10] <https://github.com/sojiadvanced/aqua-sim-carp/tree/aqua-sim-carp-dev>