Further simulation testing in CoopMAC-U for underwater acoustic sensor networks

Muhammad Syirajuddin S.[1], Nurul Hidayati[2], Rieke Adriati Wijayanti[3], Ahmad Wilda Yulianto[4], Dianthy Marya[5]
[1,2,] Telecommunication Engineering, [3,4,5] Digital Telecommunication Network
Electrical Engineering, State Polytechnic of Malang, Indonesia
[1]syirajuddin@polinema.ac.id, [2]nurulhid8@polinema.ac.id, [3]riekeaw@polinema.ac.id,
[4]ahmadwildan@polinema.ac.id, [5]dianthy@polinema.ac.id

Abstract— **In order for underwater wireless sensor networks to communicate more efficiently, MAC protocols are needed to control the use of acoustic channels. With the high propagation delay and the limited bandwidth available on the acoustic channel, a specially designed MAC protocol is needed for UWASN (Underwater Acoustic Sensor Networks). In this research, the adaptation of Cooperative MAC for underwater (CoopMAC-U) will be further studied to test the protocol performance. In the previous research, CoopMAC-U was simulated yet the fairness of the transmission was not simulated and tested. In this research, CoopMAC-U will be studied further and improved. The simulation result shows that the Improved CoopMAC-U protocol produces better-normalized throughput than the initial version of CoopMAC-U. The protocol is also proven that it is backward compatible between conventional mode and cooperative mode. For offered load greater than 0.2, both the initial version of CoopMAC-U and the Improved CoopMAC-U result in stagnant normalized throughputs but the improved ones double the value of the initial version.**

***Keywords— Acoustic Network, Cooperative, Media Access Control Protocol, Underwater, Wireless Sensor Network*.**

I. INTRODUCTION

Earth's surface covered by water is wider than the land so there are many undiscovered and exploited from the sea. It encouraged researchers to install underwater wireless sensor networks for various purposes such as environmental monitoring, underwater exploration, disaster early warning system, navigation, tactical surveillance, or mine detection for military purposes. [1], [2], [3].

Instead of using radio frequency (RF) or optical waves, acoustic waves are considered more suitable for underwater communications for wireless sensor networks. This is because radio frequency waves experience very high attenuation due to the high permittivity and conductivity of seawater and the application of optical communication which generally has a narrow beam angle is very difficult in an underwater environment [1], [2], [4].

However, the use of acoustic waves for underwater environments is not problem free. Acoustic waves propagate at varying speeds depending on temperature, salinity, and pressure or depth of seawater. The propagation speed of acoustic waves is around 1500m/s which is relatively slow when compared to the propagation speed of RF waves or optical waves in the air. This low propagation speed results high propagation delay. [5], [6].

Another problem in underwater acoustic communication systems is that the available bandwidth and the optimal frequency that can be used vary depending on the transmission distance. The farther the transmission distance, the narrower the available bandwidth and the lower the optimal frequency. Therefore, increasing the available bandwidth, it can be done by dividing the long transmission distance into shorter ones by adding a relay in the middle.

For wireless sensors to communicate using acoustic waves, a specially designed media access control (MAC) protocol that considers the unique nature of underwater acoustic waves is needed. An adaptation of Cooperative MAC (CoopMAC) protocol has been simulated by Syirajuddin et.al [7] for an underwater environment (CoopMAC-U) and it showed that the protocol outperformed the MACA-U [8] and ALOHA. However, this research only tested the performance of the CoopMAC-U protocol with a concentration on getting a higher normalized throughput and did not carry out a comprehensive simulation test regarding whether or not cooperative communication can increase channel utility. Therefore, in this article, a more detailed simulation test will be carried out on the success rate of sending data from each sensor node.

II. METHOD

A. Cooperative Media Access Control for Underwater

Cooperative communication is a communication system that utilizes assistance from other nodes to transmit data. As illustrated in Figure 1, node D (destination) receives the same signal twice, the first signal is received directly from node S (source) and the second signal is a signal that is retransmitted by node R (relay) [9]. Node R does not have to be a dedicated relay that is specially prepared for that task, but it could be other sensor nodes that are not currently transmitting.

To make cooperative communication work, a media access control protocol is needed to arrange which node and when allowed to transmit its data. The proposed CooMAC-U is a contention-based MAC Protocol that utilizes several control packets so nodes can contend the channel among them. The control packets used on CoopMAC-U are RTS (request to send), HTS (helper ready to send), and ACK (acknowledged).

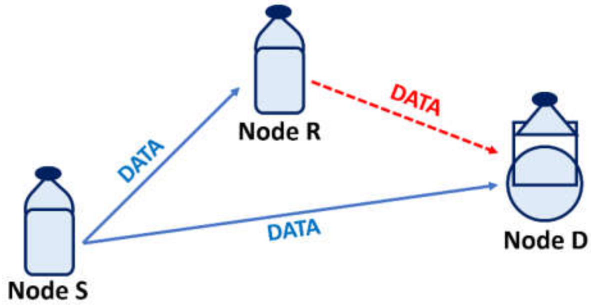

Figure 1. Illustration of a One Relay Cooperative Communication System

In this research, the improved CoopMAC-U will be discussed instead of the initial version of CoopMAC-U. The improved CoopMAC-U algorithm as illustrated in Figure 2 is started with node S in the IDLE state. When node S has a data queue in its buffer, node S will change its state to CONTEND and pick a random number to count down. When the countdown end and node S still detect the media is not occupied, it will send an RTS packet to node D. Time needed for contending follows Equation (1).

$$T_{bk} = Uniform\{0, B_{cnt}\} \times (T_{rts} + \tau_{max}) \quad (1)$$

where T_{bk} is the backoff duration in seconds, B_{cnt} is the backoff counter, T_{rts} is RTS packet duration in seconds, and τ_{max} is the maximum propagation delay in seconds.

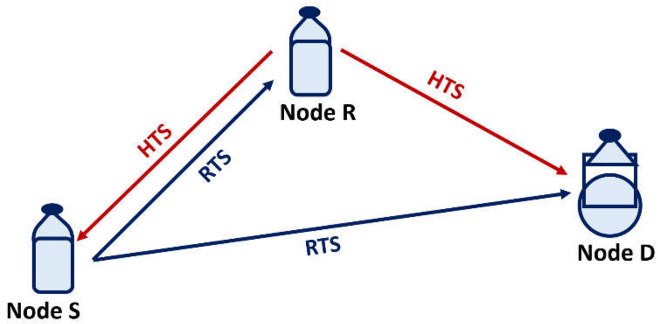

Figure 2. Control Packets Exchange for CoopMAC-U

The Backoff counter will be doubled every time the contending fails until it reaches the B_{max} value and it will be reset to a minimum when contending is successfully done. The Backoff counter is expressed in Equation (2).

$$B_{cnt} = \begin{cases} \min(2 \times B_{cnt}, B_{max}) & if\ contending\ fails \\ B_{min} & if\ contending\ success \end{cases} \quad (2)$$

where B_{max} is the maximum backoff counter allowed and set to 64 whereas B_{min} is the minimum backoff counter, set to 1

RTS packet may contain not only permission requests to send DATA to node D but also request for help to another node that can be a potential relay. It allows the nodes to choose whether the communication is in cooperative mode or simple conventional mode.

Because of the nature of wireless transmission (unguided media), the RTS sent by node S to node D may be overheard by node R which can be a potential relay. Therefore, a single control packet can achieve two different purposes at a time, request permission to transmit and request help from a potential relay. When node R is not in the middle of trying to send data, node R will respond by sending an HTS packet to notify node S and node D at a time that node R is ready to help relay the transmission.

After sending the RTS packet, node S changes its state to WFHTS (wait for HTS) and its duration follows Equation (3). If the HTS packet is received within WFHTS duration, node S will change its state to WFCTS (wait for CTS) and update the timer duration that follows Equation (4). And if the HTS packet is not received within WFHTS duration, node S will check whether it receives CTS from node D or not. Node S can still send data even though it does not receive HTS as long as it still receives CTS from node D. After sending CTS packet, node D changes its state to WFDATA (wait for DATA).

$$W_{hts} = T_{hts} + (3 \times \tau_{max}) \quad (3)$$
$$W_{cts} = T_{cts} + (2 \times \tau_{max}) \quad (4)$$

where W_{hts} and W_{cts} are the waiting time duration for HTS in second and the waiting time duration for CTS respectively, T_{hts} and T_{cts} are HTS packet duration in second and CTS packet duration respectively.

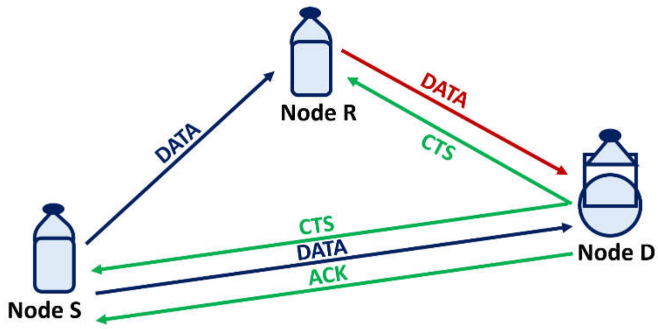

Figure 3. DATA Packet Transmission Process

As illustrated in Figure 3, after node D receives an RTS and followed by an HTS packet, it can reply with a CTS packet to give node S permission to transmit the DATA packet and it will be transmitted by node S as soon as the CTS packet is received and change its state to WFACK (wait for ACK) that its duration expressed if Equation (5). The DATA packet that is overheard by node R is decoded and then will be retransmitted again to node D on the behalf of node S.

$$W_{ack} = T_{ack} + (2 \times \tau_{max}) \quad (5)$$

where W_{ack} is the waiting time duration for ACK in seconds and T_{ack} is the ACK packet duration also in seconds.

Node D will not always receive both DATA packets from node S and relayed DATA from node R. three possibilities could happen to the DATA packets. First is that node D will successfully receive both DATA packets. Second is that node D will successfully receive only the direct transmission from node S. And the last possibility is that node D will only receive the relayed DATA from node R.

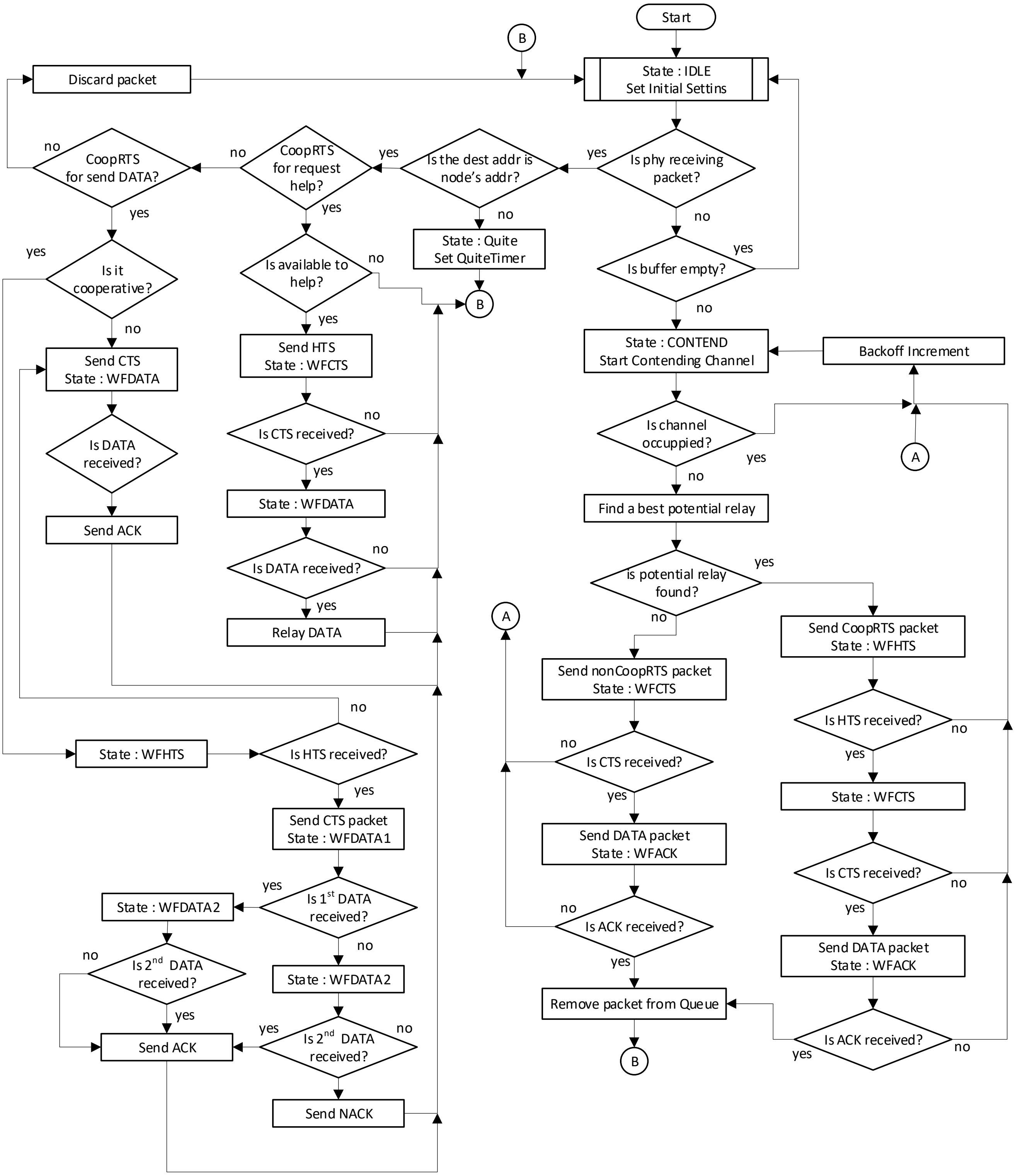

Figure 4. Flowchart of CoopMAC-U Algorithm

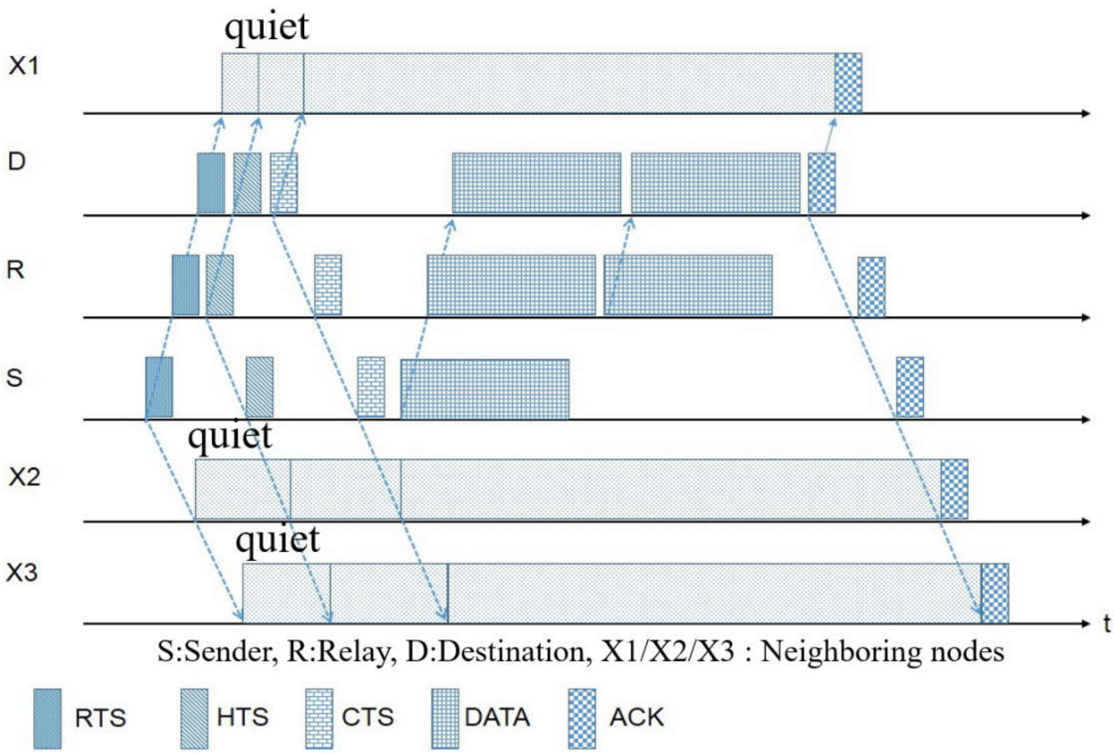

Figure 5. CoopMAC-U Timing Diagram

The three possibilities mentioned above are considered successful transmission for either of them and node D will acknowledge the transmission by sending an ACK packet that ends all transmission process and remove the DATA from the queue buffer. If node D does not receive any of them, it will send NACK packet to inform node S that the transmission fails and need retransmission. The complete CoopMAC-U algorithm is shown in Figure 4 and Figure 5.

As illustrated in Figure 5, the neighboring nodes that are not involved but overhear the transmission will defer their transmission and change their state to QUIET within the duration that conforms to the type of received packet. It is possible that the neighboring nodes to receive another packet in this QUIET state so it will update the QUIET duration to the longest duration between the remaining duration and the latest QUIET duration. The QUIET duration is based on the packet's type expressed in Equations (6), (7), and (8).

$$Q_{rts} = T_{rts} + (2 \times \tau_{max}) \quad (6)$$
$$Q_{hts} = T_{hts} + (2 \times \tau_{max}) \quad (7)$$
$$Q_{cts} = T_{cts} + (2 \times \tau_{max}) \quad (8)$$

where Q_{rts}, Q_{hts}, Q_{cts} are the quiet duration for receiving RTS, HTS, and CTS respectively. All variables are in second

B. Topology Used in Simulations

The simulation involves 16 sensor nodes placed as illustrated in Figure 6. Sensor nodes are represented by red dots and the sink is represented by a blue dot. It is assumed that the network is a fully connected network, in other words, transmission from one node can reach all nodes in the network.

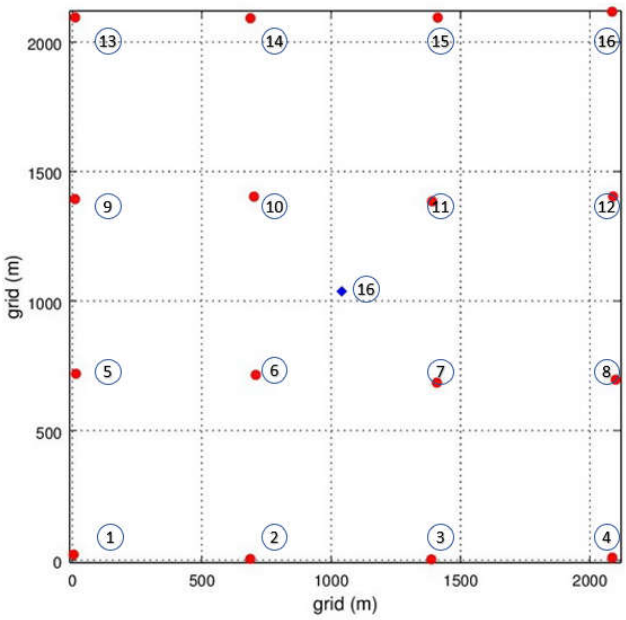

Figure 6. Underwater Acoustic Sensor Network Topology Seen from Above

C. Underwater Acoustic Propagation Model

In this research, the simulation will be conducted in NS3 discrete event simulator which is equipped with a built-in underwater acoustic network (UAN) propagation model. Acoustic wave in the NS3 simulator is assumed to experience attenuation caused by geometric loss and absorption loss as expressed in Equation (9). The first term of the equation is geometric loss which its value is dependent on distance (d) and spreading factor (k). The second term of the equation is absorption loss which the value depends on distance (d) and the absorption coefficient as the function of frequency $(\alpha(f))$. [10], [11], [12].

$$A(d,f) = k \cdot 10\log(d) + d \cdot \alpha(f) \quad (9)$$

where $A(d,f)$ is acoustic wave transmission loss in dB, k is the spreading factor, d is the distance between transmitter and receiver in meters, and $\alpha(f)$ is absorption coefficient in dB/km. In this simulation, k's value equals 1.5 (practical spreading) and the absorption coefficient will adopt the Thorp model which is expressed in Equation (10).

$$\alpha(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f} + 2.75 \cdot 10^{-4}f^2 + 0.003 \quad (10)$$

where $\alpha(f)$ is the absorption coefficient in dB/km, and f is the frequency in kHz. Equation (10) is only valid for frequencies over 0.4kHz.

Noise in this simulation is assumed only the ambient noise affects the transmission which is a combination of turbulence, shipping, wind, and thermal noise. All of those kinds of noise are expressed in these Equations (11)-(15) [10], [11], [12].

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \quad (11)$$
$$10\log N_t(f) = 17 - 30\log(f) \quad (12)$$
$$10\log N_s(f) = 40 + 20(s-0.5) + 26\log(f) - 60\log(f+0.03) \quad (13)$$
$$10\log N_w(f) = 50 + 7.5w^{1/2} + 20\log(f) - 40\log(f+0.4) \quad (14)$$
$$10\log N_{th}(f) = -15 + 20\log(f) \quad (15)$$

where $N(f)$ is ambient noise in $dB\,re\,1\mu Pa/Hz$, f is frequency in kHz, $N_t(f)$ is turbulence noise, $N_s(f)$ is shipping noise, s is the shipping noise coefficient (between 0 and 1), $N_w(f)$ is wind noise, w is the wind speed in m/s, and $N_{th}(f)$is thermal noise. The shipping noise coefficient is assumed 0 and the wind speed is 1m/s for the entire simulation in this research.

From attenuation and noise equations before, the signal-to-noise ratio (SNR) can be calculated using Equation (16).

$$SNR = \frac{P_t}{A(d,f) \cdot N(f) \cdot \Delta f} \quad (16)$$

where P_t is transmitted power in $dB\ re\ \mu Pa$, $A(d,f)$ is acoustic wave transmission loss in dB, $N(f)$ is ambient noise in $dB\ re\ 1\mu Pa/Hz$ and $\Delta f$is signal bandwidth $in\ Hz$.

From Equation (10) it is known that the absorption coefficient value increases with increasing frequency. And from Equation (9) it is known that transmission losses also increase with increasing transmission distance. Thus, if it is assumed that the signal power spectral density is 1, then the relationship between SNR, frequency and distance can be illustrated in Figure 7.

Figure 7 proves the statement in the Introduction section that the farther the transmission distance, the narrower the available bandwidth and the lower the optimal frequency. The available bandwidth for every distance indicated by the yellow line under every curve. [6].

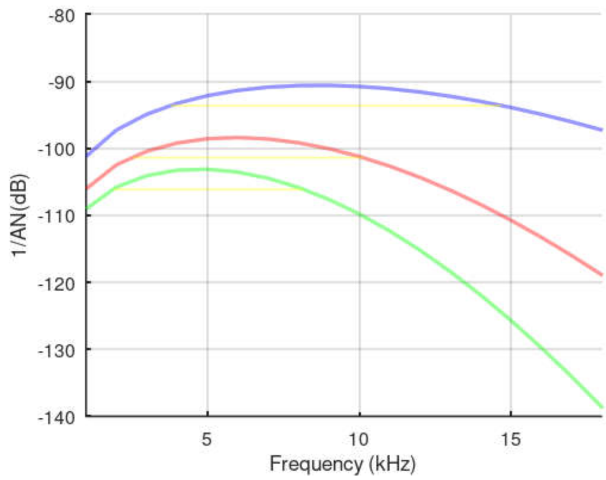

Figure 7. Signal to Noise Ratio in An Acoustic Channel Depends on the Frequency and Distance Through The Factor $1/A(d,f)N(f)$
Blue : 5km, Red : 10 Km, Green : 15 Km.

Modems used in the simulation are assumed to implement BPSK modulation on the Rayleigh fading model and Eb/No is assumed to equal SNR therefore the probability of bit error can be calculated using Equation (17) [14].

$$P_b = \frac{1}{2}\left(1 - \sqrt{\frac{\bar{\gamma}_b}{1+\bar{\gamma}_b}}\right) \quad (17)$$

where P_b is the probability of bit error and $\bar{\gamma}_b$ is Eb/No as known as SNR per bit.

Using Equation (18), the probability of packet error can be calculated after the probability of bit error is obtained.

$$P_p = 1 - (1 - P_b)^N \quad (18)$$

where P_p is the probability of packet error, P_b is the probability of bit error, and N is packet length (bit).

As mentioned before, acoustic wave propagates slower when compared to radio frequency or optical wave in the air. There are several empirical formulas to calculate sound speed underwater such as formulas proposed by Wilson, Leroy, Medwin, Mackenzie, and still many others. Most of the formulas involve temperature, salinity, pressure, or depth in the calculation. Every proposed formula has its range and limitations. Using a formula to calculate a parameter out of its range tends to give a false result. For example, here it is the Mackenzie formula that is expressed in Equation (19). [5]

$$c = 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T^3 + 1{,}340(S-35) + 1{,}630 \times 10^{-2}D + 1{,}675 \times 10^{-7}D^2 - 1{,}205 \times 10^{-2}T(S-35) - 7{,}139 \times 10^{-13}TD^3 \quad (19)$$

Where c is sound speed in meters/second, T is the temperature in Celcius, S is salinity and D is the depth of seawater in meters. This Mackenzie's equation is only valid when $-2°C < T < 35°C$, $25\% < S < 40\%$, and $0 < D < 8000m$. However, to simplify the simulation, the sound speed used in the simulation is assumed to be fixed at 1500 m/s.

D. Modem and Control Packet Technical Details

Modems are configured and simulated using the following settings described in Table I.

TABLE I
MODEM CONFIGURATION DETAILS

Parameter	Value
Data rate	4096 bits/second
Packet length	4096 bits/packet
TxPower	137 $dB\ re\ \mu Pa$
Frequency (center)	25 KHz

And here are the control packet formats used in the simulation described in Table II to Table VI.

TABLE III
CTS PACKET FORMAT

Header	Length (bit)
Frame's number	8
Retry number	8
RxSINR	24
Packet's type	8
Source address	8
Destination address	8
Helper address	8

TABLE II
RTS PACKET FORMAT

Header	Length (bit)
Frame's number	8
Number of frames	8
Frame's length	16
Retry number	8
Packet's type	8
Source address	8
Destination address	8
Helper address	8

TABLE IV
HTS PACKET FORMAT

Header	Length (bit)
Frame's number	8
Retry number	8
Packet's type	8
Source address	8
Destination address	8
Helper address	8

TABLE V
DATA PACKET FORMAT

Header	Length (bit)
Frame's number	8
Packet's type	8
Phase state	8
Data	4048
Source address	8
Destination address	8
Helper address	8

TABLE VI
ACK PACKET FORMAT

Header	Length (bit)
Frame's number	8
Acknowledged Frame	8
Packet's type	8
Source address	8
Destination address	8
Helper address	8

E. Traffic Generation and Normalized Throughput

Traffic is assumed to have a constant length for every DATA packet of 512 bytes (consisting of data dan its header). To have desired offered traffic load with a constant length of data, the interarrival time needs to be configured to follow an exponential distribution random variable. The offered load is expressed in Equation (20).

$$\rho_{node} = (\lambda_{node} \cdot L_{DATA})/R \quad (20)$$

where ρ_{node} is normalized offered load (unitless), λ_{node} is average packet arrival per second, L_{DATA} is DATA packet length in bit and R is data rate in bps.

To obtain the packet arrival rate from desired normalized offered load, Equation 19 can be rewritten as Equation 21.

$$\lambda_{node} = (\rho_{node} \cdot R)/L_{DATA} \quad (21)$$

Because interarrival time is the inverse of packet arrival rate and also equals the total of on and off packet duration as ilustrated in Figure 8, therefore :

$$1 + OffDuration = \frac{1}{\lambda_{node}} = \frac{L_{DATA}}{\rho_{node} \cdot R} \quad (22)$$

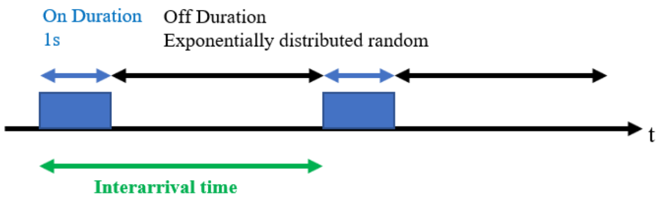

Figure 8. Relationship between On and Off Duration to Offered Load

As mentioned before, the packet length is fixed to 4096 bit/packet or 512 bytes/packet and the data rate is 4096 bit/second. Therefore, a node will need 1 second to complete one transmission and it's called On Duration whereas the off duration will be an exponentially distributed random variable with a mean value following Equation (23) [15].

$$MeanOffDuration = \frac{L_{DATA}}{\rho_{node} \cdot R} - 1 \quad (23)$$

The protocol performance will be measured on its normalized throughput which is expressed in Equation (24) [15].

$$NT = \frac{D_{total}/(n \cdot \tau_{sim})}{R/L_{DATA}} \quad (24)$$

where NT is normalized throughput (unitless), D_{total} is the total data packets received by the sink, n is the number of nodes, τ_{sim} is simulation duration in second, L_{DATA} is DATA packet length in bits and R is the data rate in bytes/seconds.

F. Relay Selection Algorithm

The improved CoopMAC-U implements a preselected relay to establish contention and handshaking between the source, potential relay, and destination. That means the source decides which node that will be requested to be a relay node. As mentioned before, an RTS packet can achieve two different purposes at a time, request permission to transmit and request help from a potential relay. That is how the preselected relay works. The problem is how a source node can decide which node is the best to be a relay node.

To make the preselected relay method works, every node must have a CoopTable that contains data about transmission quality among them by learning every transmission from the neighboring nodes. Every node has to record a pair of received signal quality which is represented by SINR (Signal to Interference plus Noise Ratio) with its sender. It's assumed that the SINR value from receiving a signal from a node will result in the same SINR when transmitting a signal to that node. In this proposed algorithm, this information is called Hop1SINR. But this information alone can't be used to decide the best relay for cooperative mode.

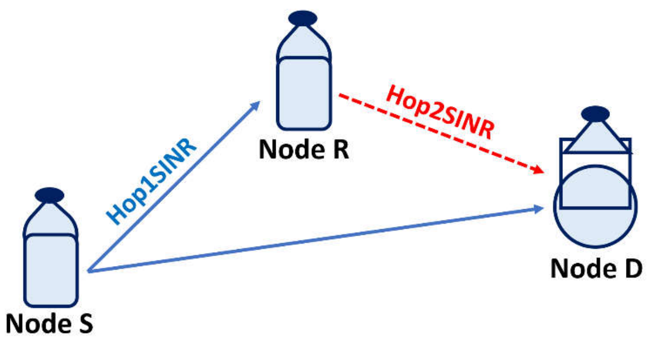

Figure 9. Informations Needed to Build CoopTable

As illustrated in Figure 9, if node S considers node R as its relay, node S needs to compare the SINR for two different transmissions. They are Hop1SINR which represents transmission quality from node S to node R and Hop2SINR which represents transmission quality from node R to node D. As mentioned before, Hop1SINR can be learned easily when node R transmits any signal to other nodes. But the Hop2SINR is difficult to learn by node S if there is no one to tell node S about the Hop2SINR value. Therefore, in the CTS packet header is embedded a piece of information that is called RxSINR to tell the RTS sender and any other neighboring nodes about the signal quality of the RTS packet when it is received by node D. Thus, node S can learn an estimated total SINR via the relay route from a potential relay.

After both Hop1SINR and Hop2SINR are obtained, the next step is to compare both of them to get the minimum value between them, it's called the estimated minimum SINR. After some time, the estimated minimum SINRs are collected and used to decide the best potential relay by selecting the highest estimated minimum SINR.

III. RESULTS AND DISCUSSION

The simulation duration for every offered load is 604800 seconds or seven days and it is repeated for 25 different offered loads. Figure 10 shows that the improved CoopMAC-U produces higher normalized throughput than the initial version. Especially when compared to ALOHA which produces the best-normalized throughput only in the lowest offered load.

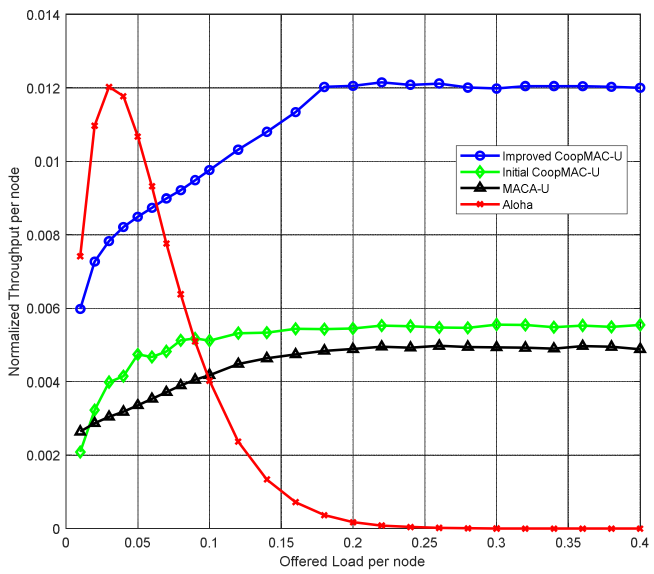

Figure 10. Normalized Throughput of Improved CoopMAC-U

Aloha has no channel contention mechanism or scheduling system to prevent packet collision. Every time node has data queuing in its buffer, at the moment node will transmit the data. Therefore, ALOHA only produces good normalized throughput when the offered load is lower than 0.1. Greater than that, the normalized throughput will decrease significantly due to collision.

Both the initial and improved CoopMAC-U protocols produce normalized throughputs that increased as the offered load increases for the offered loads lower than 0.2 and then produce a stagnant normalized throughput even though the offered loads are increased. Those steady normalized throughputs represent the maximum throughput that protocols can achieve and indicate that collision prevention mechanisms work well in both the initial CooopMAC-U and the improved CoopMAC-U version.

Improved CoopMAC-U produces higher normalized throughput than the initial version because of an additional header in the DATA packet namely "Phase state" that is not available in the initial version. Phase state is used to distinguish between the DATA packet sent by source or relay. It simplifies the sink to acknowledge the successful transmission. In addition, the new formula expressed in Equations (3) and (4) extends the waiting duration to ensure HTS/CTS packet is received before the waiting time is out. It prevents the source node to assume fail transmission when in fact may result in a successful transmission.

Table VII shows how the CoopMAC-U protocol performs to produce successful transmission. The 1st Phase column shows how many packets which is successfully received by the sink just in the first phase but never receive in the second phase. And the first opposite, the 2nd Phase column shows how many packets which is successfully received by the sink only in the second phase and fail for the first one. The Coop Complete column shows how many packets successfully receive both the first and second phase packet. No Coop column shows the number of the packet that is received in conventional mode. The successfully received packet for every node depends on the node's distance to the sink.

TABLE VII
DETAILED RECEIVED PACKETS FROM EACH NODE
FOR OFFERED LOAD EQUALS 0.01 IN IMPROVED COOPMAC-U

Node	1st Phase	2nd Phase	Coop Complete	No Coop	Total Received Packets
1	501	888	1601	4	2994
2	591	796	2246	8	3641
3	649	785	2393	26	3853
4	506	1198	1168	8	2880
5	622	765	2386	18	3791
6	0	0	1	4460	4461
7	0	0	1	4478	4479
8	586	812	2296	13	3707
9	601	777	2209	18	3605
10	0	0	1	4424	4425
11	0	0	0	4403	4403
12	621	820	2317	17	3775
13	487	1177	1064	6	2734
14	614	810	2420	14	3858
15	625	876	2387	19	3907
16	476	0	861	0	1337

Table VII also shows that nodes number 6, 7, 10, and 11 produce a high number of received packets only in "No Coop" column. It shows that they worked in a conventional mode almost all the time instead of in a cooperative mode. Except for those nodes, all of the nodes work mostly in cooperative mode. This result shows that the relay selection algorithm works so every node can decide whether work in cooperative mode when they find a potential relay or conventional mode when they don't find a potential relay. This is consistent with the network topology illustrated in Figure 6 that all of the nodes but them have a potential helper for relaying data to the sink. Because they are the nearest nodes to the sink, they have the highest opportunity to win a contention and result in a successful transmission, that is the reason why they result in a higher number of total received packets.

TABLE VIII
DETAILED RECEIVED PACKETS FROM EACH NODE
FOR OFFERED LOAD EQUALS 0.2 IN IMPROVED COOPMAC-U

Node	1st Phase	2nd Phase	Coop Complete	No Coop	Total Received Packets
1	182	189	670	5	1046
2	283	176	991	23	1473
3	252	158	918	22	1350
4	142	251	478	7	878
5	296	208	1057	27	1588
6	0	3	4	25991	25998
7	0	0	0	23646	23646
8	264	204	958	25	1451
9	256	210	1017	17	1500
10	1	0	3	24720	24724
11	1	0	1	27237	27239
12	280	229	1027	22	1558
13	171	303	406	2	882
14	214	165	873	10	1262
15	238	206	1012	20	1476
16	189	0	376	0	565

To have a better understanding of the Improved CoopMAC-U protocol's behavior, other detailed data are presented in Table VIII. It shows that Improved CoopMAC-U turns out doesn't work properly in terms of fairness when the offered load is increased. The channel is occupied most of the time by nodes 6, 7, 10, and 11. But the normalized throughput is almost level when the offered load is greater than 0.2 indicating that the collision prevention mechanism still works.

IV. CONCLUSION

The improved CoopMAC-U outperforms the initial version of CoopMAC-U by two times thanks to a revision in the DATA header and the new formulas for the waiting timer after sending an RTS packet. The improved CoopMAC-U is also proven to have the ability to select between conventional mode or cooperative mode depending on the availability of a potential relay. The normalized throughput keeps flat when offered load equals or greater than 0.2 indicating that the collision prevention mechanism is functioning. Nonetheless, the fairness of channel contention still to be a concerning problem when the offered load is increased.

REFERENCES

[1] X. Guo, M. R. Frater dan M. J. Ryan, "Design of a Propagation-delay-tolerant MAC Protocol for Underwater Acoustic Sensor Networks," *Oceanic Engineering, IEEE Journal of,* vol. 34, no. 2, pp. 170-180, 2009.

[2] I. F. Akyildiz, D. Pompilo dan T. Melodia, “Underwater Acoustic Sensor Networks: Research Challenges.,” *Ad hoc networks,* vol. 3, no. 3, pp. 257-279, 2005.

[3] P. Wang, F. Wei, L. Zhang and V. O. Li, "Asynchronous Cooperative Transmission in Underwater Acoustic Networks," in *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*, 2011.

[4] F. T. Ulaby, Fundamentals of Applied Electromagnetics., Prentice Hall, 1995.

[5] P. C. Etter, Underwater Acoustic Modeling and Simulation, CRC Press, 2013.

[6] M. Stojanovic and J. Preisig, "Underwater Acoustic Communication Channels: Propagation Models and Statistical Characterization," *IEEE Communications Magazine,* vol. 47, no. 1, pp. 84-89, 2009.

[7] M. S. Suja'i, Endroyono and Wirawan, "Protokol CoopMAC-U Untuk Jaringan Sensor Akustik Bawah Air," in *SENIATI*, Malang, 2016.

[8] H.-H. Ng, W.-S. Soh and M. Motani, "MACA-U : A Media Access Protocol for Underwater Acoustic Networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008.

[9] Y. W. P. Hong, C. C. J. Kuo and W.-J. Huang, Cooperative Communications and Networking, Springer, 2010.

[10] T. Cinar and M. B. Orencik, "An Underwater Acoustic Channel Model Using Ray Tracing in ns-2," in *Wireless Days (WD), 2009 2nd IFIP*, Paris, 2009.

[11] A. F. Harris III and M. Zorzi, "Modelling the Underwater Acoustic Channel in ns2," in *{Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, 2007.

[12] A. Sehgal, I. Tumar and J. Schonwalder, "Aquatools: An Underwater Acoustic Networking Simulation Toolkit," in *IEEE, Oceans, Sydney, Australia May*, 2010.

[13] R. J. Urick, Principles of Underwater Sound, McGraw-Hill, 1983.

[14] J. G. Proakis and M. Salehi, Digital Communication, McGraw-Hill Education, 2007.

[15] W. Lin, E. Cheng and F. Yuan, "A MACA-based MAC Protocol for Underwater Acoustic Sensor Networks," *Jurnal of Communications,* vol. 6, no. 2, p. 179, 2011.