International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 3 (2019), pp. 434-439

Receiver Synchronization based Collision Reduction Protocol for Underwater Acoustic Sensor Networks

Sunmyeng Kim

Associate Professor, Department of Computer Software Engineering,
Kumoh National Institute of Technology, Korea.

Abstract

Underwater acoustic sensor networks (UASNs) have long propagation delay of acoustic signals, which pose challenges to the design of medium access control (MAC) protocol. Most of the studies on MAC protocols focus on contention-based techniques. Due to the spatial-temporal uncertainty problem, contention based MAC protocols cause high collision probability as the number of sender nodes increases. In order to reduce the collision probability, we propose a receiver synchronization based collision reduction (RSCR) protocol. The proposed protocol divides the channel time into time slots. A sender node introduces an appropriate delay before transmitting a frame to ensure that the frame arrives exactly at the start time of the time slot of a receiver node. Therefore, the probability of collision decreases. Performance evaluation is conducted using simulation, and confirms that the proposed protocol outperforms the previous protocol.

Keywords: Collision, Contention, MAC, Synchronization, Time Slot, UASN.

I. INTRODUCTION

Underwater acoustic sensor networks (UASNs) are a class of sensor networks deployed in underwater environments [1]. UANs have attracted much attention in recent years due to their potential in various applications. There are significant differences between UANs and wireless networks because of the unique features such as low available bandwidth, long propagation delay, and dynamic channels in acoustic modems. These features pose challenges to medium access control (MAC) protocol design [2,3,4]. And, MAC protocols for wireless networks cannot be directly applied to UASNs because the work is based on high data rates and negligible propagation delays. Especially, carrier sense multiple access / collision avoidance (CSMA/CA) cannot prevent packet collisions well among nodes due to the long propagation delays in UANs. Therefore, it is necessary to design new MAC protocols to take into account the different features.

Significant efforts have been devoted to the underwater MAC protocol design to overcome the negative effects introduced by the harsh underwater environments [3,4,5]. Most of them are based on the handshaking in order to reduce the collision probability in UANs. They use control packets such as Request-to-Send (RTS) and Clear-to-Send (CTS) to contend and reserve channel for data transmissions.

Ng, et al. proposed a bidirectional-concurrent MAC (BiC–MAC) protocol based on concurrent, bidirectional data packet exchange to improve the data transmission efficiency [6]. In the BiC–MAC protocol, a sender-receiver node pair is allowed to transmit data packets to each other for every successful handshake. Noh, et al. proposed a delay-aware opportunistic transmission scheduling (DOTS) protocol [7]. In DOTS, each node learns neighboring nodes' propagation delay information and their expected transmission schedules by passively overhearing packet transmissions. And then, it makes transmission scheduling decisions to increase the chances of concurrent transmissions while reducing the likelihood of collisions. In Reference [8], the authors proposed a multiple access collision avoidance protocol for underwater (MACA-U) in which terrestrial MACA protocol was adapted for use in multi-hop UASNs. In the MACA-U protocol, a source node transmits a RTS packet to a destination node after channel contention. After receiving the RTS packet, the destination node transmits a CTS packet. And then, the source node transmits its own data packet to the destination node. When other nodes receive the RTS or CTS packets, they set their timer and do not participate in the data packet transmission process.

In [9], authors indicated the spatial-temporal uncertainty problem in underwater environment. When sender nodes have frames to transmit, they send RTS packets to the receiver node. The RTS packet arrival time at the receiver node depends on the distance between the sender and the receiver. The contention based MAC protocols cannot control RTS packet collisions well among sender nodes due to the spatial-temporal uncertainty problem. That is, the long propagation delay of acoustic media causes high collision probability as the number of sender nodes increases. It significantly decreases network performance of contention-based protocols.

In order to reduce the collision probability, we propose a receiver synchronization based collision reduction (RSCR) protocol. The proposed protocol divides the channel time into time slots. A sender node introduces an appropriate delay before transmitting a frame to ensure that the frame arrives exactly at the start time of the time slot of a receiver node. Therefore, the probability of collision decreases.

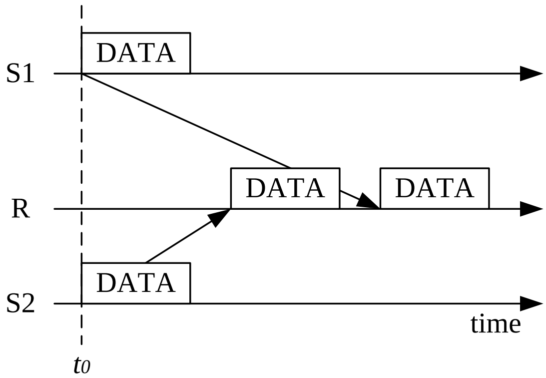
The paper is organized as follows. We discuss related work on the spatial-temporal uncertainty problem of UASNs in section II. In section III, the proposed RSCR MAC protocol is described in detail. In Section IV, performance studies are carried out through simulation results. Finally, we draw conclusions in section V.

II. RELATED WORK

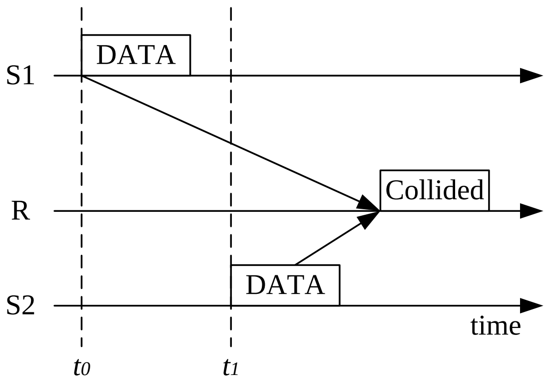
In this section, we first discuss the spatial-temporal uncertainty problem. And then, we explain why the collision probability is high in contention based MAC protocols.

II.I Spatial-temporal Uncertainty Problem

Nodes in terrestrial wireless networks can estimate the channel status easily since the propagation delay is very short and negligible. However, in UASNs, it is essential to consider the location and transmission time of a node due to the long propagation delay of acoustic media [10]. Spatial-temporal uncertainty is defined as two-dimensional uncertainty in determining a collision at a receiver. The packet collision at a receiver depends on both the distance between a sender and a receiver and the sender's transmission time. As the distance between the nodes increases, the current channel status cannot be clearly known due to the propagation delay. Even though nodes do not send packets at the same time, the packets may collide.



(a) Same transmission time, no collision at R



(b) Different transmission time but collision at R

Fig. 1. Example of Spatial-temporal Uncertainty

Fig. 1 shows an example of the spatial-temporal uncertainty problem. In Fig. 1, there are two senders (S1 and S2) and one receiver (R). In Fig. 1(a), two senders transmit their data packets at the same time. However, the receiver receives the packets at different time due to the different propagation delay. In other words, there are no collision at the receiver. On the other hand, two senders transmit their packets at the different time (see Fig. 1(b)). The packets arrive at the receiver at the same time and are collided.

II.II High Collision Probability

Because the propagation delay is very short and can be ignored, nodes in terrestrial wireless networks can easily estimate the channel status. Therefore, with channel detection, all nodes can be synchronized one another.

Fig. 2 shows an example of collisions in wireless networks. In Fig. 2, there are three senders (S1, S2, and S3) and one receiver (R). The three senders simultaneously know the end of the channel busy status at time t_1. And then, they start their backoff procedures, respectively. At time t_2, the backoff procedures are terminated and they send frames (Frame 1, Frame 2, and Frame 3). The frames arrive at the receiver at the same time at time t_2 and they overlap completely. Therefore, a collision occurs.

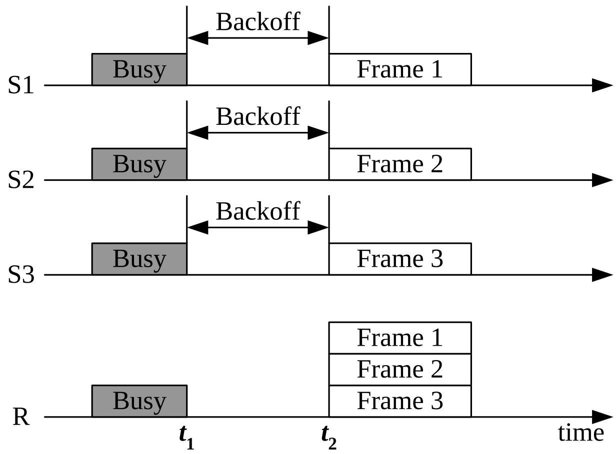


Fig. 2. Example of Collisions in Wireless Networks

In UASNs, it is essential to consider the location and transmission time of a node due to the long propagation delay of acoustic media. Therefore, with channel detection, all nodes cannot be synchronized one another. Two or more transmissions overlap in time, partially or wholly at the receiver. Consequently, UASNs have a much higher collision probability than wireless networks.

In the contention based MAC protocols for UASNs, after completing the backoff procedure, sender nodes send an RTS packet to the receiver node. RTS packets may collide at the receiver node. The probability of collision is related to the vulnerable period. It is defined by the time interval during which RTS packets may collide. If the first bit of a new RTS packet overlaps with just the last bit of an RTS packet almost finished, both RTS packets collide. Fig. 3 shows an example of RTS packet collisions in the contention based MAC protocols. In Fig. 3, the receiver node R receives three RTS packets from the sender nodes S1, S2, and S3. We assume that an RTS packet from the sender node S2 arrives at time t. T_{RTS} is the

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 3 (2019), pp. 434-439

transmission time of an RTS packet. If an RTS packet from the sender node S1 arrives between times t-T_{RTS} and t, the RTS packets from the sender nodes S1 and S2 make a collision. Also, if an RTS packet from the sender node S3 arrives between times t and t+T_{RTS}, a collision occurs. In other words, when a receiver node receives an RTS packet at time t, any RTS reception that begins in interval [t, t+T_{RTS}], or in the prior T_{RTS} seconds leads to collision. Therefore, the vulnerable period is 2 * T_{RTS}.

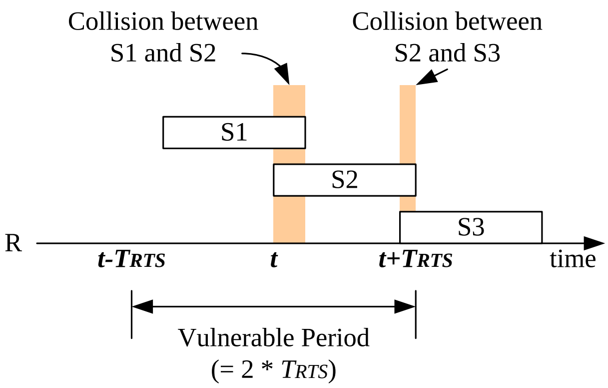


Fig. 3. Example of Collisions in UASNs

III. PROPOSED RSCR MAC PROTOCOL

In this section, we describe the proposed RSCR protocol. All nodes synchronize to the time slot of the receiver node. A sender waits an appropriate delay before transmitting a frame to ensure that the frame arrives exactly at the start time of the time slot of the receiver node. In the proposed RSCR protocol, a collision occurs when frames overlap at the receiver node, but there is no partial overlap of frames. Therefore, the probability of collision decreases.

III.I Basic Idea

In order to reduce the vulnerable period and collision probability, we propose a RSCR protocol in which the time of the shared channel is divided into discrete intervals called time slots. Each time slot corresponds to the length of the RTS packet. In contrast to the previous contention based MAC protocols, the RSCR protocol allows a receiver node to receive RTS packets only at the start time of the time slot. Sender nodes transmit RTS packets to arrive at the start time of the time slot of a receiver node. In the RSCR protocol, there is still a possibility of collision if two or more RTS packets arrive at the start time of the same time slot. Therefore, the vulnerable period is the length of a time slot (= T_{RTS}). Packets collide in the RSCR protocol only when they overlap completely instead of partially. Consequently, the probability of collision is reduced by half.

Fig. 4 shows an example of RTS packet collisions in the RSCR protocol. The sender node S1 delays an RTS packet to arrive at time t. Also, an RTS packets from the sneder node S2 arrive at time t. Therefore, The RTS packets from the sender nodes S1 and S2 a collision. However, an RTS packet from the sender node S3 arrives at t+T_{RTS} and does not make a collision.

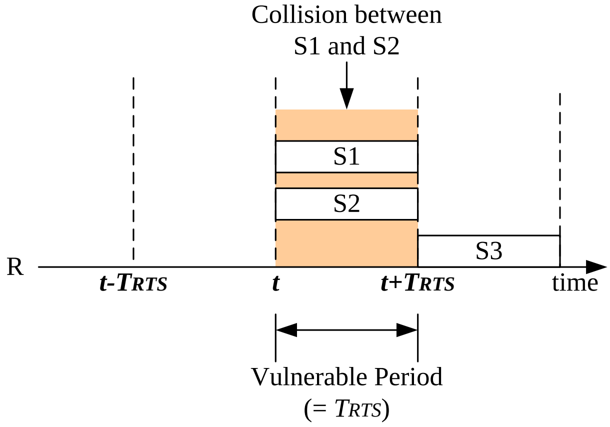


Fig. 4. Example of Collisions in the RSCR Protocol

III.II Receiver Synchronization

The proposed RSCR protocol uses the four-way handshaking (RTS-CTS-DATA-ACK) mechanism to reserve the channel before transmitting data packets.

Before describing the proposed RSCR protocol in detail, we introduce a CTS packet format which is modified from the existing CTS format. As shown in Fig. 5, a 2-byte Offset field is added to the existing frame format. This offset value is used to calculate the delay time that indicates how long a sender node should wait before transmitting frames. We will explain how to calculate the delay time in detail below.

Octets: 2	2	6	2	4
Frame Control	Duration	Receiver Address	Offset	FCS

Fig. 5. CTS Frame Format

MAC Address of Receiver	Delay Time
R_1	T_1
…	…
R_n	T_n

Fig. 6. Format of the DelayTable

As shown in Fig. 6, each sender node maintains a table, referred to as the *DelayTable*. A sender node receives a CTS packet by a receiver node, and then updates its *DelayTable*. The *DelayTable* contains 2 fields. The first field MAC Address of Receiver is MAC addresses of the receiver nodes. The Delay Time field is delay time that the sender waits before transmitting RTS packets. The initial value of the Delay Time field is zero.

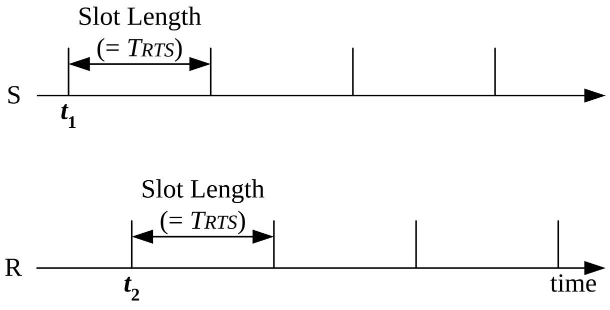


Fig. 7. Time Slot of Nodes

Every node randomly sets the start time of the time slot when it powers on. Fig. 7 shows an example of time slots of a sender node and a receiver node. The start time of the first time slot of the sender node S is time t_1. And the start time of the first time slot of the receiver node R is time t_2. The slots of all nodes are not synchronized with one another since the start time of the first time slot of each node is set randomly.

When a sender node has a data packet to send, it starts its own backoff procedure. After completing the backoff procedure, the sender node transmits an RTS packet at (the start time of the time slot + Delay Time in the DelayTable). If this RTS packet is first transmitted by the sender node, it is transmitted at the start time of the time slot because the initial value of the Delay Time field is zero (see Fig. 8).

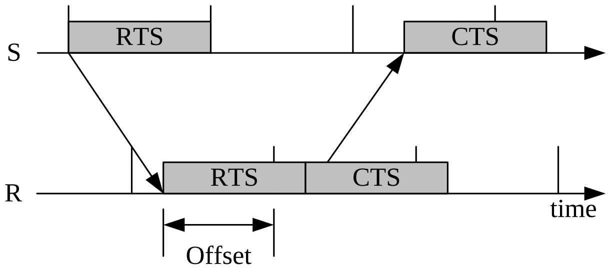


Fig. 8. First RTS Packet Transmission by a Sender Node

After receiving the RTS packet from the sender node, the receiver node computes the offset value. The offset value is the difference between the reception time of the RTS packet and the start time of the next time slot (see Fig. 8). The receiver node saves the calculated value in the offset field of a CTS packet. And then, it transmits the CTS packet to the sender node.

After receiving the CTS packet from the receiver node, the sender node updates the delay time in the DelayTable as following.

$$DT_n = (DT_{n-1} + Offset)\ \%\ T_{RTS} \quad (1)$$

where, DT_n is the updated delay time, DT_{n-1} is the delay time after receiving the (*n*-1)th CTS packet from the receiver node, and offset is the value of the offset field in the *n*th CTS packet. % is the remainder operator.

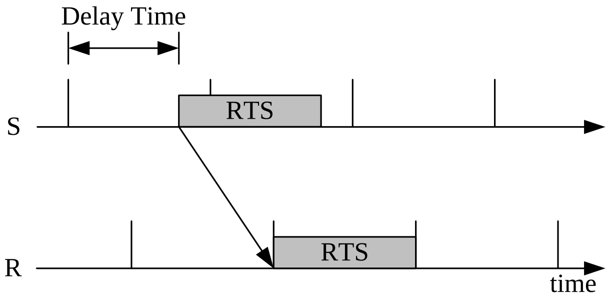


Fig. 9. RTS Packet Transmission

The sender node transmits the second and subsequent RTS packets at time (the start time of the time slot + Delay Time) (see Fig. 9).

Through this operation, sender nodes always synchronize with the time slot of the receiver node. RTS packets arrives exactly at the start time of the time slot of a receiver node.

IV. SIMULATION RESULT

In this section, we analyze simulation results of the proposed RSCR protocol. To study the performance of the RSCR protocol, we actually implemented the protocol in C++. Performance of the RSCR protocol is compared with that of the MACA-U protocol. We simulated a with a maximum data rate of 1,500 bps. The length of control and data packets are constant. Sound speed is 1500m/s. The maximum transmission range is 1,500m.

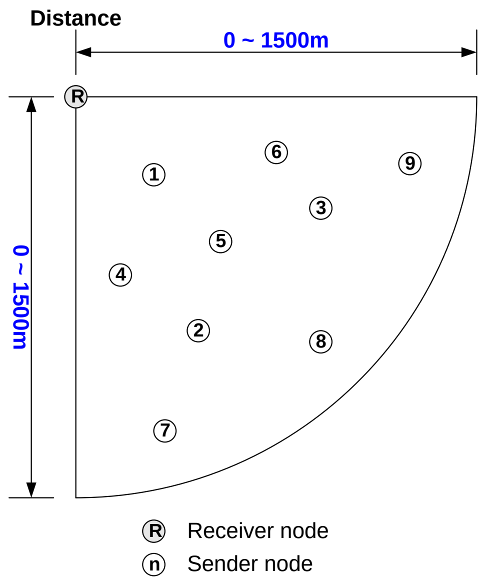


Fig. 10. Simulation Topology

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 3 (2019), pp. 434-439

In the simulation, we consider the topology shown in Fig. 10. In the topology, there are several sender nodes and one receiver node. The sender nodes have data packets to send to the receiver node. The receiver node has no data packets to send and is only a receiver node of them. All sender nodes are deployed in a 2-D area of 1500m * 1500m. All sender nodes are able to hear one another. The receiver node is placed at the point (0, 0). Sender nodes are randomly distributed in the topology.

The main performance metric of interest is the number of collisions that occur until all sender nodes complete transmitting their data. All simulation results were averaged over 10 simulations.

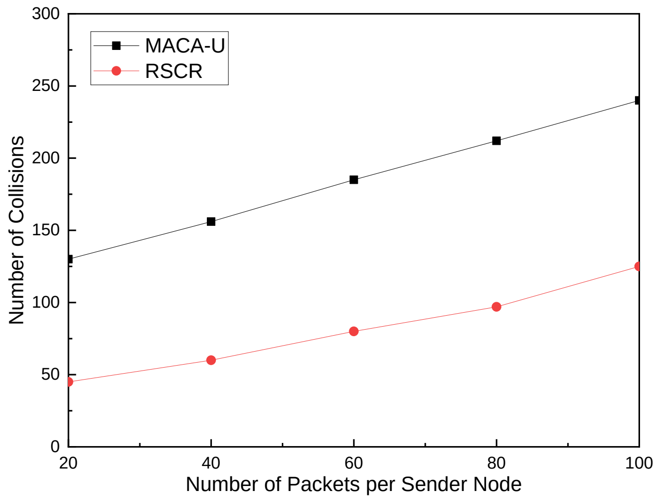


Fig. 11. Number of Collisions according to Number of Packets per Sender Node

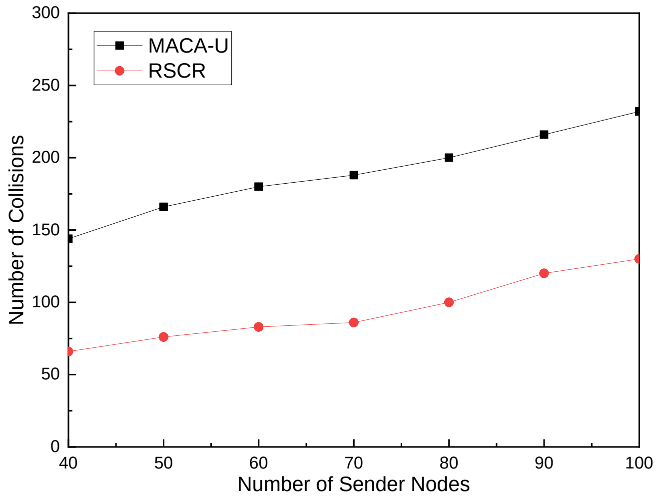


Fig. 12. Number of Collisions according to Number of Sender Nodes

Fig. 11 shows the results for the number of collisions according to the number of packets per sender node. There are 70 sender nodes. Each sender node transmits data packets as many as the number on the X-axis. From the figure, we can see that in the RSCR and MACA-U protocols, the number of collisions linearly increases. The proposed RSCR protocol always shows better performance than the MACA-U protocol. The proposed RSCR protocol has about 50% lower number of collisions than the MACA-U protocol regardless of the number of pacakets per sender nodes. In the proposed RSCR protocol, sender nodes always synchronize with the time slot of the receiver node. Sender nodes transmit RTS packets to arrive at the start time of the time slot of a receiver node. When packets collide in the proposed RSCR protocol, they overlap completely instead of partially, and this significantly decreases the number of collisions.

Fig. 12 shows the results for the number of collisions according to the number of sender nodes. Each sender node has 80 data packets to transmit. Fig. 12 shows the results similar to Fig. 11. From the figure, we can see that the proposed RSCR protocol always has less number of collisions than the MACA-U protocol. In the MACA-U protocol, collisions occur, when two or more transmissions overlap in time partially or wholly at the receiver node. However, in the proposed RSCR protocol, collisions do not occur for any partial overlap of packets at the receiver node.

V. CONCLUSION

In UASNs, most of MAC protocols focus on contention-based techniques. Due to the spatial-temporal uncertainty problem, contention based MAC protocols cause high collision probability as the number of sender nodes increases. Especially, previous MAC protocols have high collision probabilities because two or more transmissions overlap in time partially or wholly at the receiver. High collision probability causes the low channel utilization, which in turn severely affects network performance. In order to reduce the collision probability, we proposed the RSCR protocol which divides the channel time into discrete intervals called time slots. In the proposed RSCR protocol, sender nodes synchronize to the time slot of the receiver node. A sender waits an appropriate delay before transmitting a packet to ensure that the packet arrives exactly at the start time of the time slot of the receiver node. When packets collide in the proposed RSCR protocol, they overlap completely instead of partially, and this significantly decreases the number of collisions. Simulation result shows that the proposed RSCR protocol significantly outperforms the previous protocol.

ACKNOWLEDGEMENT

This research was supported by Kumoh National Institute of Technology.

REFERENCES

[1] P. Casari and M. Zorzi, Protocol Design Issues in Underwater Acoustic Networks, Computer Communications, 34(17), 2011, 2013-2025.

[2] N. Morozs, P. Mitchell, and Y. V. Zakharov, TDA-MAC: TDMA Without Clock Synchronization in Underwater Acoustic Networks, IEEE Access, 6, 2017, 1091 - 1108.

[3] Y. Zhu, Z. Peng, J.-H. Cui, and H. Chen, Toward Practical MAC Design for Underwater Acoustic Networks, IEEE Transactions on Mobile Computing, 14(4), 2015, 872-886.

[4] L. Pu, Y. Luo, H. Mo, S. Le, Z. Peng, J.-H. Cui, and Z. Jiang, Comparing Underwater MAC Protocols in Real Sea Experiments, Computer Communications, 56, 2015, 47-59.

[5] Y. Zhang, Y. Chen, S. Zhou, X. Xu, X. Shen, and H. Wang, Dynamic Node Cooperation in an Underwater Data Collection Network, IEEE Sensors Journal, 16(11), 2015, 4127-4136.

[6] H.-H. Ng, W.-S. Soh, and M. Motani, A Bidirectional-Concurrent MAC Protocol with Packet Bursting for Underwater Acoustic Networks, IEEE Journal of Oceanic Engineering, 38(3), 2013, 547-565.

[7] Y. Noh, U. Lee, S. Han, P. Wang, D. Torres, J. Kim, and M. Gerla, DOTS: A Propagation Delay-Aware Opportunistic MAC Protocol for Mobile Underwater networks, IEEE Transactions on Mobile Computing, 13(4), 2014, 766-782.

[8] H.-H. Ng, W.-S. Soh, and M. Motani, MACA-U: A Media Access Protocol for Underwater Acoustic Networks, IEEE Globecom 2008, 2008, 1-5.

[9] C.-C. Hsu, M.-S. Kuo, C.-F. Chou, and K. C.-J. Lin, The Elimination of Spatial-Temporal Uncertainty in Underwater Sensor Networks, IEEE/ACM Transactions on Networking, 21(4), 2013, 1229-122.

[10] W. H. Liao and C. C. Huang, SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks, IEEE Sensors Journal, 12(6), 2012, 1686-1694.